Inteligencia Artificial, 22(64) (2019), 1-13
doi: 10.4114/intartif.vol22iss64pp1-13

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Building Dynamic Lexicons for Sentiment Analysis

Nicolás Mechulam, Damián Salvia, Aiala Rosá, Mathias Etcheverry
Instituto de Computación, Facultad de Ingeniería, Universidad de la República. Uruguay
nicomebu@gmail.com
damian.salvia@gmail.com
aialar@fing.edu.uy
mathiase@fing.edu.uy

Abstract: Nowadays, many approaches for Sentiment Analysis (SA) rely on affective lexicons to identify emotions transmitted in opinions. However, most of these lexicons do not consider that a word can express different sentiments in different predication domains, introducing errors in the sentiment inference. Due to this problem, we present a model based on a context-graph which can be used for building domain specific sentiment lexicons (DL: Dynamic Lexicons) by propagating the valence of a few seed words. For different corpora, we compare the results of a simple rule-based sentiment classifier using the corresponding DL, with the results obtained using a general affective lexicon. For most corpora containing specific domain opinions, the DL reaches better results than the general lexicon.

Keywords: Lexicon Induction, Sentiment Analysis, Natural Language Processing.

1 Introduction

Sentiment Analysis identifies and extracts opinions in order to classify their sentiments. According to Taboada et al. (2011), sentiment can be induced by lexicon based approaches, or supervised text classification, eventually including lexicon based features. Generally, the sentiment is inferred by applying heuristics or machine learning methods on texts, eventually using sentiment information of words, such as *sentiment lexicon*.

The basic structure of sentiment lexicons is a reduced and selected set of words with their *valence* (sentiment or affection, e.g. ‘bad’ as negative or ‘good’ as positive). Many classification algorithms take this dictionary as input for identifying the sentiment of a whole text or for improving the results obtained by machine-learning techniques. Even though its use is necessary, usually it is not enough.

As Levin (1991) says, the lexicon usage “has often proved to be a bottleneck in the design of large-scale natural language systems”. This fact, despite technological advances from the last two decades, is still being a problem for creating and applying this kind of resources because of heterogeneous and ambiguous word’s meaning. For instance, Rosá et al. (2017) remarks that the use of sentiment lexicons does not produce significant improvements in sentiment prediction.

According to Pang and Lee (2008), when we face a lexicon construction problem, we must define as a first step which will be the lexicon’s units. There are many options to achieve this goal depending on text-preprocessing cost and sentiment dissipation from same-meaning lexical variations (i.e. by token, by

ISSN: 1137-3601 (print), 1988-3064 (on-line)

word, by lemma, by stem). For example, '*Educated*' and '*educaaated*' tokens refers to '*educated*' word, this word and '*educating*' word refers to '*educate*' lemma, and this lemma with '*educative*' lemma refers to '*educ*' stem. Note how sentiment is grouped when lexical unit is more general, but it requires more preprocessing on each lexical treatment.

Once lexical units are defined, we should define how sentiment is expressed and how lexical items will be retrieved. For the first problem, according to Jurafsky and Martin (2017), sentiments can be expressed in three ways: in a discrete domain (or categorical), in a continuous domain (a real number), or by vectors (where each component represents a sentiment feature). The same authors, for the second problem, say that there are two main approaches for retrieving relevant words and their sentiment: 1) by manual inference (human annotator/s), and 2) by machine learning strategies. The first family presents high accuracy with a considerable human effort, while the second one leads to a wider lexical coverage.

The process of building sentiment lexicons is a complex task because there is not a 'universal receipt' that indicates how sentiment is expressed. In consequence, these kinds of resources have been built by linguistics or experts, and it is hard to achieve high coverage in any context. For example, the word 'long' expresses opposite sentiments in the sentences "it is a long road to get there" and "a long weekend is coming". In the first case, 'long' expresses a negative opinion over 'the travel time', and in the second example it expresses a positive opinion over 'the weekend duration'. Also, on the same predication domain, there can be conflicting phrases (e.g. 'low' in "low price" and "low salary" for financial domain), or there can be phrases influenced by valence shifters (e.g. negation in 'it's not clean' really means 'it's dirty').

In this paper we present a method for the automatic construction of affective lexicons. For this, we estimate the valence of words accor ding to their contexts and to the domain on which opinions are expressed. We call these lexicons Dynamic Lexicons. Despite the method was tested only for Spanish, its nature makes it language-independent.

The remaining document is structured in four main sections. Section 2 corresponds to related works that present some novelty about lexicon inference, Section 3 describes the bases of the generation model, Section 4 shows empirical results and Section 5 concludes the work made.

2 Related work

Liu (2012) exposes four problems when using lexicons for predicting sentiments: 1) Opposite affection depending on the opinion object (e.g. 'big' is positive for capacity but negative for electronic devices), 2) Opinions without affection words (e.g. "last night my phone got wet" is negative but it does not have any key word), 3) Neutral sentences with affection word (e.g. questions like "will my phone have a good signal there?"), and 4) Sarcasm/irony sentences (e.g. "what a good phone I bought! Two days it works..."). The problem becomes more complex when combining other linguistic phenomena, such as: intonation in written text (e.g. emojis or words as 'EXCELLENT' or 'eeeexcelent') (Muhammad et al., 2013), intensification/negation (e.g. 'tiniest') (Polanyi and Zaenen, 2006), or when treating global vs. local sentiment (e.g. "it's a good phone with a bad design") (Pang and Lee, 2008).

Hatzivassiloglou and McKeown (1997) propose one of the first methods for inducing a sentiment lexicon of adjectives. Their system starts from 1336 seed words with known valence assuming there are some conjunctions that preserve the valence (e.g. 'and') and others that invert it (e.g. 'but'). They report a precision of 82.05%.

Wu and Wen (2010) try to infer the polarity of ambiguous adjectives. They propose the concept of *sentiment expectation* that relates a noun (with known valence) with an adjective. For instance, if 'salary' has positive expectation, then 'low' should be negative-like in the phrase "the salary is low" due this phrase is negative. The authors report 80.06% of macro-F measure and 79.72% of accuracy.

Turney and Littman (2002) use PMI (Pointwise Mutual Information) and LSA (Latent Semantic Analysis) as a semi-supervised mechanism, reporting 74% of precision. Probably, the most important fact of their work is that they use only two seeds, one of each class ('excelent' and 'poor'). Also, the nature of the applied method (PMI+LSA) encapsulates domain-specific information from the corpora used to build the lexicon.

Many works for sentiment lexicon construction rely on a graph based approach, in the following we comment the most relevant ones for our work.

Kamps et al. (2004) use WordNet to find word synonyms from a set of seeds in order to propagate their valence, or invert them when they are antonyms. The valence assigned to new words is calculated by a function called EVA (short for 'evaluation') which measures the distance between 'good' and 'bad' adjectives. The authors report 68.19% of precision.

Esuli and Sebastiani (2007) also use WordNet to build SentiWordNet, but they build a graph based on word's glosses and then apply a modified version of PageRank algorithm over it. For evaluation they use *Kendall Correlation* measure, reporting 0.324 for positive words and 0.284 for negative words.

Another work inspired in WordNet is Hamilton et al. (2016), who introduce SentProp. They build a graph using word embeddings where each word is a node connected with their k-nearest neighbors in the semantic word space model. Then, they apply a label-propagation mechanism over the valence from a few seed words following a probabilistic model defined on the weight of each edge. They report 86.7% of accuracy, 68.8 % of f-measure and 0.4 of Kendall correlation.

Chen and Skiena (2014) build a graph with semantic inter-language edges using a variety of linguistic resources, starting from an English sentiment lexicon as seeds. They cover 136 different languages and evaluate their results using existing non-English sentiment lexicons.

Our method is inspired by these graph based approaches. We take the idea of building a word's graph in order to propagate the valence of a few seed words (with known valence) through it. However, our method differs from them in the fact that edges in our graph represent syntagmatic co-occurrence of words in texts from a specific domain, instead of representing semantic relations between words, provided by resources such as WordNet or word embeddings.

3 Our Approach

Our work is based on the following hypothesis: "*the sentiment of a word depends on its context*". Such context should be understood by 1) the neighbor words (or textual context), 2) the negation influence, and 3) the predication domain. In these terms, we create a mechanism for building a lexicon of <word,valence> pairs based on the architecture described in Figure 1.

We started from a set of diverse corpus (C_1,...,C_n) with sentiment-tagged opinions (e.g. reviews from movies, sports, restaurants, etc.), described in Tables 1 and 2 – refer to the author's paper for further information (annotation method, valence meaning, etc). The complete corpus has 79.643 opinions. Each corpus was preprocessed applying word correction and lemmatization, identifying negated words and standardizing their heterogeneous valences.

To detect words within a negation scope we trained a bidirectional long-short term memory network using the SFU-NEG corpus (Martí Antonín et al., 2016). The inputs of the neural model are word embeddings built using skip-gram on a Spanish corpus of almost six billion words Azzinnari and Martínez (2016). More details on these experiments can be found in Salvia and Mechulam (2018).

After these preprocessing steps, we used the 10% of each corpus to build an *Independent Lexicon* (IL) where words sentiment does not depend on the context to which the words belong, they are always positive or negative.

From the remaining corpora, we used 72% of each corpus C_i to build each *Dynamic Lexicon* (DL_i), using the words of the IL as seeds. The dynamic lexicon is built by generating a context-graph and applying a semi-supervised valence propagation mechanism over it.

We used the remaining 18% of each C_i as test corpus. To evaluate we compared the results obtained using the domain specific generated lexicon against a general purpose lexicon in sentiment analysis detection.

In the following, we describe the mechanisms applied to build the IL and the DL.

3.1 Independent Lexicon

We built an *Independent Lexicon* (IL) taking the ideas of Ghag and Shah (2014). These authors propose Senti-TFIDF, a mechanism which treats sentiments as a Bag-of-Words (BoW). The result will be a lexicon where word's sentiment is independent of the context, since it includes words that are frequently used

[1] The complete list is: cars, hotels, washing machines, books, cellphones, music, computes, movies.

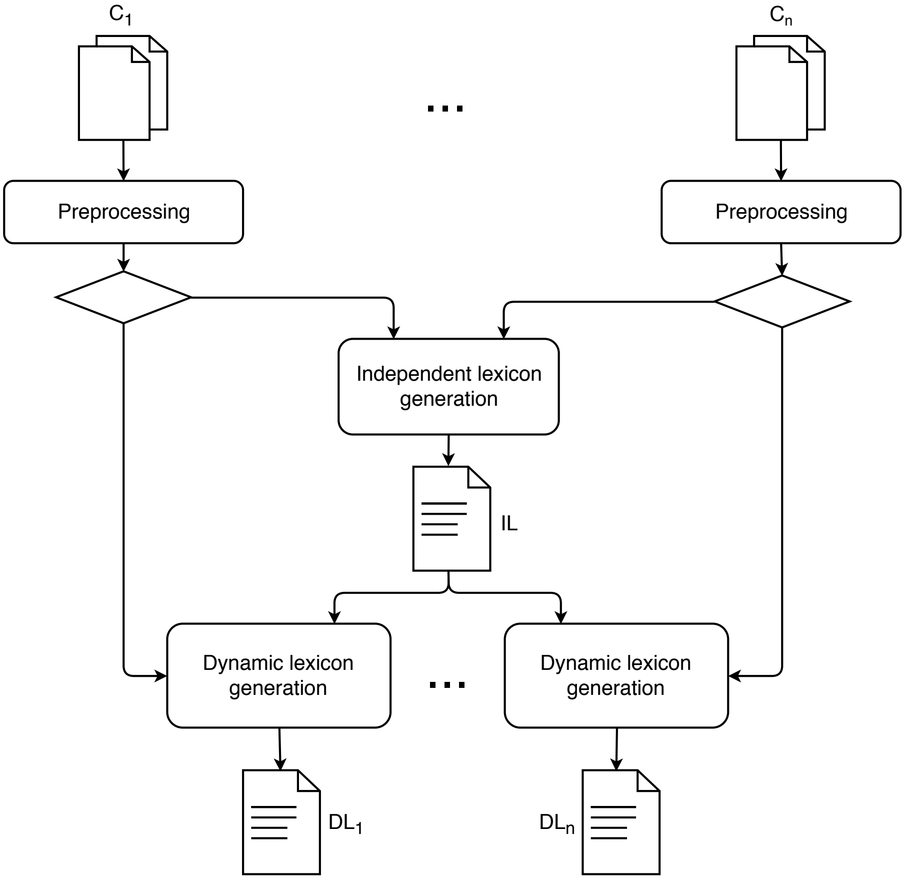

C1
Cn
Preprocessing
Preprocessing
Independent lexicon generation
IL
Dynamic lexicon generation
Dynamic lexicon generation
DL1
DLn

Figure 1: Architecture

with the same polarity in the different domains we studied. These words are used as seeds for sentiment propagation, in the construction of the dynamic lexicon. In this work, despite we know when a word is negated or not, we modify the original mechanism to include the negation in the calculus, resulting in the following equations:

$$\text{X-TF}_{t,d} = \frac{\begin{cases} f_{t^+,d} & \text{if } d \in D_X \\ f_{t^-,d} & \text{if } d \in D_{\overline{X}} \end{cases}}{\sum_{x \in D_X} f_{t^+,x} + \sum_{x \in D_{\overline{X}}} f_{t^-,x}}$$

$$\text{X-IDF}_t = \log \frac{|D_X \cup D_{\overline{X}}|}{|\{d \in D_X : t^+ \in d\} \cup \{d \in D_{\overline{X}} : t^- \in d\}|}$$

where X represents the POS or NEG class and $\overline{X}$ the complementary class of X; besides, t^-/t^+ represent if term t is negated or not, respectively. The words in the IL will be the top 150 with higher absolute value of TF-IDF (product of both formulas).

Taking a portion of the resources described in Table 1 (as it was detailed in the introduction of this chapter), we built an IL limited to 150 words, including 75 positive words and 75 negative words. For instance, the most representative words on each category were *desastre* ('disaster'), as negative, and *excelente* ('excellent'), as positive.

3.2 Dynamic Lexicon

In this section we describe the construction of the context graph and the mechanism for valence propagation, which uses the words from the IL as seeds.

Reference	**Valences**	**Domain**
Apps by Dubian (2013b)	pos, neg	Mobile apps
Mov by Cruz Mata et al. (2008)	1,2,3,4,5	Movies
Hotel by Molina-González et al. (2014)	1,2,3,4,5	Hotels
Med1 by Plaza-Del-Arco et al. (2016)	0,1,2,3,4,5	Medicine
Med2 by Jiménez-Zafra et al. (2017)	positive, negative, neutral	Medicine
Misc1 by Vilares et al. (2015)	pos 1-5, neg 1-5	General
Misc2 by Martí Antonín et al. (2016)	pos, neg	Cars, hotels,... [1]
Misc3 by Mori et al. (2016)	P, N, NEU, NONE	Politics
Jour1 by Mordecki et al. (2015)	Pos, Neu, Neg	Journalism
Jour2 by Boldrini et al. (2009)	positive, negative, ε	Journalism
Rest1 by Jiménez-Zafra et al. (2009)	1,2,3,4,5	Restaurants
Rest2 by Dubian (2013a)	pos, neg	Restaurants

Table 1: Collected corpus with sentiment annotation by domain

Corpus	**Total**	**IL** (10%)	**DL** (72%)	**Eval** (18%)
Apps	8.281	828	7.453	1.491
Mov	3.878	388	3.490	698
Hotel	1.816	182	1.634	327
Med1	743	74	669	134
Med2	453	45	408	82
Misc1	3.600	360	3.240	648
Misc2	400	40	360	72
Misc3	2.466	247	2.219	444
Jour1	2.260	226	2.034	407
Jour2	974	97	877	175
Rest1	2.202	220	1.982	396
Rest2	52.570	5.257	47.313	9.463
Total	79.643	7.964	71.679	14.337

Table 2: Corpus sizes after splitting for IL, DL and Evaluation.

3.2.1 Context Graph

The idea behind the construction of the *Context Graph*, inspired by the semantic graphs used in (Kamps et al., 2004; Esuli and Sebastiani, 2007; Hamilton et al., 2016; Santos et al., 2012), is that nodes represent the vocabulary of the domain, and edges represent the textual contexts (contiguous words in the text). For instance, if the words 'A' and 'B' appear together in the text, then there is an edge $<$ A, B $>$ in the graph, representing an *adjacency relation*. Each specific corpus will be associated to a graph from which a *DL* will be obtained. Thus, a context graph will be derived from each particular domain.

Normally, in sentences like "The sky is turning red", the words 'The' and 'is' lack lexical content, and their occurrence avoids adjacency relations such as 'sky' - 'beautiful'. To solve this issue we decided to discard stop words and auxiliary verbs.

Note that the resulting graph contains many connected components, this issue hinders the propagation of the valences outside each component, reducing the graph coverage. As a way to reduce the generation of connected components, we decided to compact the different inflections of a word in a single node using

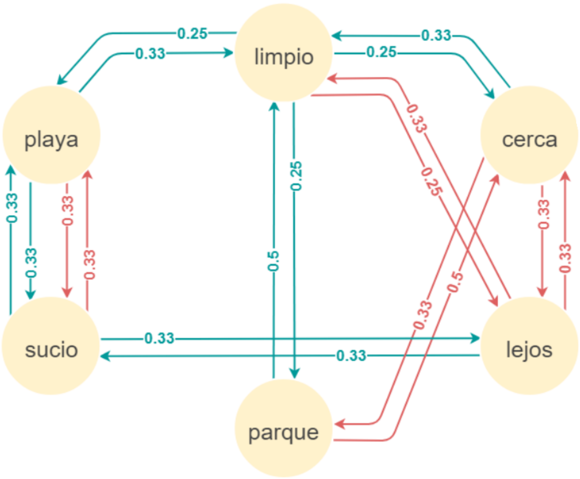


Figure 2: Context graph of a simple corpus.

its lemma and, in this way, decrease the sparsity without losing affective content.

The presence of negations in the text affects the valence of the words within its scope. Since the context graph seeks to model the contexts of words, negations are an additional aspect to be represented in the structure.

To model the negation, we decided to maintain two types of relationships: ((direct)) and ((inverse)). However, the affection transmitted by the relationships <no-X, Y > and < X, no-Y > is the same. That is, considering "The dog that does not bite is good" and "The dog that bites is not good", the relations between ((bites)) and ((good)) is assumed to express the same sentiment, although the sentences are semantically different [2].

Formally, direct relations ($< x$, $y >_{dir}$) are understood as links between two neighboring words with the same affective value (both are denied or none are). On the other hand, an inverse relation ($< x$, $y >_{inv}$) is one in which the neighboring words have a different affective value (one has the opposite value than the other one). In these conditions, for the first case, the valence is maintained, while the second is inverted.

Although for a set of texts the graph reflects the words contexts, it does not take into account the frequency with which they occur. That is, assuming that the words <dog, bites> co-occur more frequently than <dog, plays>, the former should take more importance in the representation, since 'dog' is closely related to 'bites' and probably has a similar affective value. For this reason, we decided to apply a weighting mechanism based on the number of co-occurrences in the texts, the weight (w) of an edge is the probability that two words are adjacent (directly or indirectly).

It is important to note that the probability $P(y|x)$ depends on the relation of the node "x". Therefore, since the weight of $< x, y >$ and $< y, x >$ could take different values, there is a need to express the origin and destination of the relation. Consequently, the context graph must be directed in order to adequately represent the weights. Given that there exist direct and inverse relations, the context graph is a directed graph with multiple weighted edges.

In figure 2 we show the context graph resulting from the following corpus, designed to explain the concept:

- La playa es sucia y está lejos. / The beach is dirty and far.
- El parque es limpio y no está lejos. / The park is clean and it is not far.
- La playa está limpia y es cerca. / The beach is clean and near.
- El parque no está cerca, está lejos. / The park is not near, it is far.

[2] Logically bites good does not express the same thing that bites good.

- La playa no está sucia. / The beach is not dirty.

As we can see, *sucia/sucio* (*dirty*) and *limpia/limpio* (*clean*) are represented by a unique node, labeled with the corresponding lemma (*sucio* and *limpio*). Stop words (*el, la*) and auxiliary verbs (*es, está*) are not included in the graph. Green edges indicate direct relations while red ones indicate inverse relations. Each edge weight is computed dividing the number of co-occurrences of its endpoints (distinguishing negated and non-negated contexts) by the number of occurrences of its origin. In order to illustrate this, we analyze the three occurrences of the lemma *playa*:

- the first one is in the non-negated context of *sucio*, generating a direct edge (green) with a weight of 0.33 (1/3) between the nodes *playa* and *sucio*,
- the second one is in the negated context of *sucio*, hence an inverse edge (red) with a weight of 0.33 (1/3) is added,
- the last one is in the non-negated context of *limpio*, resulting in a direct edge (green) with a weight of 0.33 (1/3).

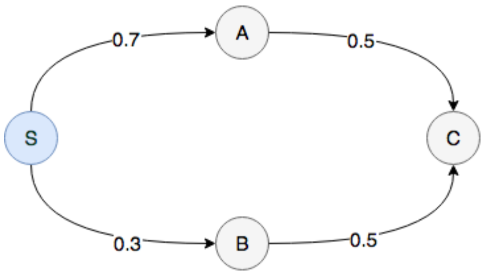
3.2.2 Valence Propagation

In this section, we describe the algorithm used for valence propagation over the *context graph*. This algorithm allows us to obtain the valence of each node of the *context graph*, starting from the seeds (IL), whose affective value is known. The result of this algorithm will determine the DL.

The algorithm is based on the concept of influence (inf), a positive real value that represents how similar is the valence of a node respect to another. This value is given by the path that maximizes the product of the edges. It is calculated recursively according to Equation 1, where in returns the incident nodes to the target. Notice how the function, when looking for the incidents of the destination node ($in(y)$), implicitly calculates the maximum path from x to y. To clarify this idea, the Example 3.1 is proposed (for simplicity the bidirectionality of the edges is omitted).

$$inf(x,y) = \begin{cases} 1 & : \text{if } x = y \\ \max\limits_{z \in in(y)} \{inf(x,z) * \omega(<z,y>)\} & : \text{else} \end{cases} \tag{1}$$

Example 3.1. *Given the following simplified context graph, it is shown the calculation (top-down) of the influence of node S (seed) towards node C.*



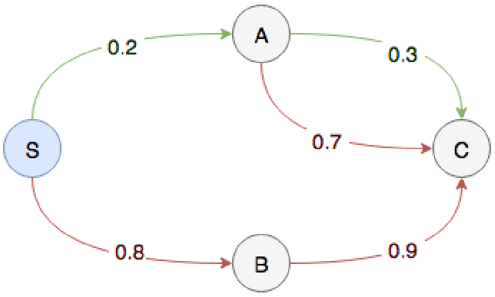
$$\begin{aligned} inf(S,C) &= \max \begin{Bmatrix} inf(S,A) * \omega(<A,C>) \\ inf(S,B) * \omega(<B,C>) \end{Bmatrix} \\ &= \max \begin{Bmatrix} inf(S,S) * \omega(<S,A>) * \omega(<A,C>) \\ inf(S,S) * \omega(<S,B>) * \omega(<B,C>) \end{Bmatrix} \\ &= \max \begin{Bmatrix} 1 * 0.7 * 0.5 & = 0.35 \\ 1 * 0.3 * 0.5 & = 0.15 \end{Bmatrix} \\ &= 0.35 \end{aligned}$$

Note that Example 3.1 does not include inverse edges. This simplification is intentional since the Equation1 only describes the influence of direct relations. Remember that a direct relation represents two neighboring words with the same sentiment, while an inverse relation represents words with opposite sentiments. In this sense, it is necessary to define **direct influence** (inf_{dir}) and **inverse influence** (inf_{inv}) in a mutually recursive way according to the Equation 2

$$
\begin{aligned}
inf_{dir}(x,y) &= \begin{cases} 1 & : \text{if } x = y \\ \max\limits_{z \in in(y)} \begin{Bmatrix} inf_{dir}(x,z) * \omega(< z,y >_{dir}) \\ inf_{inv}(x,z) * \omega(< z,y >_{inv}) \end{Bmatrix} & : \text{else} \end{cases} \\
inf_{inv}(x,y) &= \begin{cases} 0 & : \text{if } x = y \\ \max\limits_{z \in in(y)} \begin{Bmatrix} inf_{dir}(x,z) * \omega(< z,y >_{inv}) \\ inf_{inv}(x,z) * \omega(< z,y >_{dir}) \end{Bmatrix} & : \text{else} \end{cases}
\end{aligned}
\tag{2}
$$

It is important to note that the *direct influence* of a node on itself is 1 since a word always has the same valence as itself. Analogously, the *inverse influence* of a node on itself is 0 since a word never has a valence opposite to itself. As a way to analyze the influence calculations with this new formula, Example 3.2 is proposed. A real context graph is much more complex, for this reason an algorithm inspired by Dijkstra was designed to optimize the calculations.

Example 3.2. *Given the following context graph, the calculation (bottom-up) of the direct and inverse influences from the node S (seed) to node C is shown.*



Node S (base step)

$inf_{dir}(S,S) = 1$

$inf_{inv}(S,S) = 0$

Node A

$$inf_{dir}(S,A) = \max \begin{Bmatrix} inf_{dir}(S,S) * \omega(< S,A >_{dir}) \\ inf_{inv}(S,S) * \omega(< S,A >_{inv}) \end{Bmatrix} = \max \begin{Bmatrix} 1 * 0.2 & = 0.20 \\ 0 * 0.0 & = 0.00 \end{Bmatrix} = 0.20$$

$$inf_{inv}(S,A) = \max \begin{Bmatrix} inf_{dir}(S,S) * \omega(< S,A >_{inv}) \\ inf_{inv}(S,S) * \omega(< S,A >_{dir}) \end{Bmatrix} = \max \begin{Bmatrix} 1 * 0.0 & = 0.00 \\ 0 * 0.2 & = 0.00 \end{Bmatrix} = 0.00$$

Node B

$$inf_{dir}(S,B) = \max \begin{Bmatrix} inf_{dir}(S,S) * \omega(< S,B >_{dir}) \\ inf_{inv}(S,S) * \omega(< S,B >_{inv}) \end{Bmatrix} = \max \begin{Bmatrix} 1 * 0.0 & = 0.00 \\ 0 * 0.8 & = 0.00 \end{Bmatrix} = 0.00$$

$$inf_{inv}(S,B) = \max \begin{Bmatrix} inf_{dir}(S,S) * \omega(< S,B >_{inv}) \\ inf_{inv}(S,S) * \omega(< S,B >_{dir}) \end{Bmatrix} = \max \begin{Bmatrix} 1 * 0.8 & = 0.80 \\ 0 * 0.0 & = 0.00 \end{Bmatrix} = 0.80$$

Node C

$$inf_{dir}(S,C) = \max \begin{Bmatrix} inf_{dir}(S,A) * \omega(< A,C >_{dir}) \\ inf_{inv}(S,A) * \omega(< A,C >_{inv}) \\ inf_{dir}(S,S) * \omega(< B,C >_{dir}) \\ inf_{inv}(S,S) * \omega(< B,C >_{inv}) \end{Bmatrix} = \max \begin{Bmatrix} 0.2 * 0.3 & = 0.06 \\ 0 * 0.7 & = 0.00 \\ 0 * 0.0 & = 0.00 \\ 0.8 * 0.9 & = 0.72 \end{Bmatrix} = 0.72$$

$$inf_{inv}(S,C) = \max \begin{Bmatrix} inf_{dir}(S,A) * \omega(< A,C >_{inv}) \\ inf_{inv}(S,A) * \omega(< A,C >_{dir}) \\ inf_{dir}(S,B) * \omega(< B,C >_{inv}) \\ inf_{inv}(S,B) * \omega(< B,C >_{dir}) \end{Bmatrix} = \max \begin{Bmatrix} 0.2 * 0.7 & = 0.14 \\ 0 * 0.3 & = 0.00 \\ 0 * 0.9 & = 0.00 \\ 0.8 * 0.0 & = 0.00 \end{Bmatrix} = 014$$

Once defined the influence between nodes (direct and inverse), the valence of a word is calculated according to the influence it receives from the seeds (Equation 3).

$$val(x) = \frac{\sum_{s \in S} \; (inf_{dir}(s,x) * val(s) \; - \; inf_{inv}(s,x) * val(s))}{\sum_{s \in S} \; (inf_{dir}(s,x) \; + \; inf_{inv}(s,x))} \tag{3}$$

In order to summarize the strategy hereby described, Algorithm 1 is presented considering all the aspects discussed.

```
Input: context graph (G), seeds (S)
Output: lexicon (L)
influences_nodes ← ∅
foreach (seed, valence) in seeds do
    influences ← get_influences(graph, seed)
    foreach (node, inf_dir, inf_inv) in influences do
        influences_nodes[node] ← (seed, inf_ndir, inf_inv)
    end
end
L ← ∅
foreach inf_node in influences_nodes do
    sum_val ← 0
    sum_inf ← 0
    foreach (s, inf_dir, inf_inv) in inf_node do
        sum_val ← sum_val + seeds[s] * ( inf_dir − inf_inv )
        sum_inf ← sum_inf + inf_dir + inf_inv
    end
    L[node] ← sum_val / sum_inf
end
return L
```

Algorithm 1: Propagation algorithm by seeds influence

Another propagation mechanism has been tested in Salvia and Mechulam (2018) original work, inspired in BFS algorithm (*Breath First Search*) with a slightly poorer different result.

4 Results

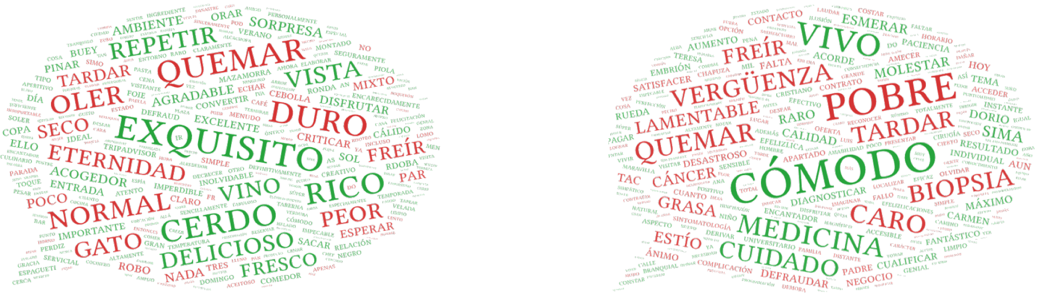
In order to evaluate the Dynamic Lexicons obtained for each domain, we separated 18% of each corpus (14.337 opinions, see Table 2) before building the respective DL. We apply a simple lexicon based heuristic for sentiment detection: we compute the sentiment of an opinion through the valence average of the affective words it contains (inverting the valence of negated words). For each domain, we compare the results obtained using the corresponding DL with the results obtained using a *General Lexicon* (GL) with word polarities for Spanish ((Saralegi and San Vicente, 2013), version *ElhPolar_esV1.lex* has 5.200 words). In Table 3 we show the Macro-F measure for each corpus and for each lexicon (DL vs GL). This evaluation was carried out on the portion of the corpus reserved for testing.

As can be seen, for most specific domain corpora (movies, hotels, restaurants, medicine) the classifier based on the Dynamic Lexicon reaches slightly better results than the one based on the General Lexicon, except for apps domain. On the other hand, for corpora with opinions covering different domains (miscellaneous and journalistic texts) the General Lexicon has a better performance, as expected. The best performance was obtained using the domain specific lexicon on the largest corpus (REST2), reaching a Macro-F of 87.5

Notice that in most cases the improvement obtained is not statistically significant, excepting the REST2 corpus, which is remarkably larger than the others. These results show the relevance of the corpus size in the Dynamic Lexicon generation. Moreover, as a positive aspect, it is important to note that similar (even better) results are obtained using the Dynamic Lexicons, which are much smaller than the general one (20 times smaller approximately).

Corpus	Test Size	Dynamic Lex. Size	Dynamic Lexicon	General Lexicon
APPS	1.491	177	71.1 ±2.3	**73.7** ±2.2
MOV	698	214	**68.9** ±3.4	63.5 ±3.5
HOTEL	327	246	**85.1** ±3.8	84.6 ±3.9
MED1	134	155	**69.4** ±7.8	60.1 ±8.3
MED2	82	79	**49.0** ±10.8	40.7 ±10.6
MISC1	648	211	48.8 ±3.8	**63.5** ±3.7
MISC2	72	137	59.0 ±11.4	**68.8** ±10.7
MISC3	444	123	61.2 ±4.5	**63.1** ±4.5
JOUR1	407	49	59.9 ±4.8	**75.6** ±4.2
JOUR2	175	113	51.1 ±7.4	**54.7** ±7.4
REST1	396	266	**84.8** ±3.5	84.0 ±4.5
REST2	9.463	210	**87.5** ±0.6	84.9 ±0.7

Table 3: Macro-F_1 for each corpus using the Dynamic Lexicons and the 5.200 words General Lexicon.



(a) REST2

(b) MED1

Figure 3: Word clouds results.

In Figure 3 we show some word clouds for restaurant and medicine domains. Bigger words represent higher valences, while green/red colors represent positive/negative words respectively. This graphical representation of the result helps to intuitively understand how well the Dynamic Lexicon behaves in the domain where it had been inferred. Note that words like ‘cerdo’ (‘pig’), that have a positive valence in the Restaurants domain, could have a negative valence in other cases.

5 Conclusions

We have presented a semi-supervised mechanism for building a sentiment lexicon from unlabeled corpora and a set of seed words (named IL). The resources we have generated prove that it is possible to build a lexicon considering the words contexts as long as the corpus belongs to a specific domain. The generated lexicons achieve a competitive score in comparison to external general-purpose lexicons, even having much fewer words. Also, as it was expected, our sentiment lexicons induced from specific domains produced better results than those induced from general domain corpora, in which case a general-purpose lexicon works slightly better.

The context-graph proposal has value in itself. We have mentioned only one valence propagation mechanism but other algorithms over it could be applied. Also, the context-graph can be used for other problems in which the words context must be taken into account.

As future work, concerning the generation of the specific domain lexicons, we propose to include aspect information using word’s local context, in order to better solve the problem of valence ambiguity.

References

Azzinnari, A. and Martínez, A. (2016). Representación de palabras en espacios de vectores. Proyecto de grado, UdelaR (Universidad de la República).

Boldrini, E., Balahur, A., Martinez-Barco, P., and Montoyo, A. (2009). Emotiblog: an annotation scheme for emotion detection and analysis in non-traditional textual genres. *Proceedings of the 1st Workshop on Opinion Mining and Sentiment Analysis, WOMSA09*, pages 491–497.

Chen, Y. and Skiena, S. (2014). Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 383–389.

Cruz Mata, F., Troyano Jiménez, J. A., Enríquez de Salamanca Ros, F., and Ortega Rodríguez, F. J. (2008). Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Procesamiento del lenguaje natural. N. 41.*

Dubian, L. (2013a). Analisis de sentimientos sobre un corpus en español: Experimentación con un caso de estudio. *Jornadas Argentinas de Informática.*

Dubian, L. (2013b). Procesamiento de lenguaje natural en sistemas de análisis de sentimientos. Tesis de grado, UBA (Universida de Buenos Aires).

Esuli, A. and Sebastiani, F. (2007). Pageranking wordnet synsets: An application to opinion mining. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 424–431. Association for Computational Linguistics.

Ghag, K. and Shah, K. (2014). SentiTFIDF sentiment classification using relative term frequency inverse document frequency. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 5(2).

Hamilton, W. L., Clark, K., Leskovec, J., and Jurafsky, D. (2016). Inducing domain-specific sentiment lexicons from unlabeled corpora. *CoRR*, abs/1606.02820.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jiménez-Zafra, S. M., Martín-Valdivia, M. T., Molina-González, M. D., and na López, L. A. U. (2009). Corpus COAR, with opinions about restaurants. Available in: `http://sinai.ujaen.es/coar/` (last access 02/11/17).

Jiménez-Zafra, S. M., Martín-Valdivia, M. T., Molina-González, M. D., and Ureña López, L. A. (2017). Corpus annotation for aspect based sentiment analysis in medical domain. *International Artificial Intelligence in Medicine Conference (AIME).*

Jurafsky, D. and Martin, J. H. (2017). *Speech and Language Processing.* (Book Draft), 3 edition.

Kamps, J., Marx, M., Mokken, R., and De Rijke, M. (2004). Using wordnet to measure semantic orientations of adjectives. *Institute for Logic, Language and Computation (ILLC), FNWI.*

Levin, B. (1991). Building a lexicon: The contribution of linguistics. *International Journal of Lexicography*, 4(3):205.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* Morgan-Claypool.

Martí Antonín, M. A., Taulé Delor, M., Nofre, M., Marsó, L., Martín Valdivia, M. T., and Jiménez Zafra, S. M. (2016). La negación en español: análisis y tipología de patrones de negación. *Procesamiento del Lenguaje Natural, Revista n 57.*

Molina-González, M. D., Martínez-Cámara, E., Martín-Valdivia, M. T., and Urena-López, L. A. (2014). Cross-domain sentiment analysis using spanish opinionated words. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 214–219. Springer.

Mordecki, G., Kremer, F., and Dufort y Alvarez, G. (2015). Estudio de reputación a partir de comentarios extraídos de redes sociales. Proyecto de grado, UdelaR (Universidad de la República).

Mori, M., Tambucho, M., and Cardozo, D. (2016). Estudio de reputación a partir de comentarios extraídos de redes sociales. Proyecto de grado, UdelaR (Universidad de la República).

Muhammad, A., Wiratunga, N., Lothian, R., and Glassey, R. (2013). Contextual sentiment analysis in social media using high-coverage lexicon. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 79–93. Springer.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–158.

Plaza-Del-Arco, F., Martín-Valdivia, M., Zafra, S. M., González, M. D., and Martínez-Cámara, E. (2016). Copos: Corpus of patient opinions in spanish. application of sentiment analysis techniques. *SINAI Research Group, Universidad de Jaen*, 57:83–90.

Polanyi, L. and Zaenen, A. (2006). Contextual valence shifters, chapter 1. computing attitude and affect in text: Theory and applications.

Rosá, A., Chiruzzo, L., Etcheverry, M., and Castro, S. (2017). RETUYT in TASS 2017: Sentiment analysis for spanish tweets using SVM and CNN. In *Proceedings of TASS 2017: Workshop on Sentiment Analysis at SEPLN co-located with 33nd SEPLN Conference*, volume 1896. CEUR Workshop Proceedings.

Salvia, D. and Mechulam, N. (2018). Construcción de léxicos dinámicos para el análisis de sentimientos. Proyecto de grado, UdelaR (Universidad de la República).

Santos, A. P., Oliveira, H. G., Ramos, C., and Marques, N. C. (2012). A bootstrapping algorithm for learning the polarity of words. In *International Conference on Computational Processing of the Portuguese Language*, pages 229–234. Springer.

Saralegi, X. and San Vicente, I. n. (2013). Elhuyar at tass 2013. In *Proceedings of XXIX Congreso de la Sociedad Española de Procesamiento de Lenguaje Natural, Workshop on Sentiment Analysis at SEPLN (TASS2013)*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Turney, P. D. and Littman, M. L. (2002). Unsupervised learning of semantic orientation from a hundred-billion-word corpus. *CoRR*, cs.LG/0212012.

Vilares, D., Thelwall, M., and Alonso, M. A. (2015). The megaphone of the people? Spanish SentiStrength for real-time analysis of political tweets. *J. Inf. Sci.*, 41(6):799–813.

Wu, Y. and Wen, M. (2010). Disambiguating dynamic sentiment ambiguous adjectives. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1191–1199, Stroudsburg, PA, USA. Association for Computational Linguistics.

6 Appendix

In this section we show some examples extracted from the corpora described in Table 1.

- **Corpus APPS (Dubian, 2013b)** - Rank: NEGATIVO - Compre un galaxy s3 ... Una estafa. Nunca responde el servicio de atencion al cliente. Compre un galaxy s3 que prometia 50gb durante 2 años y ha pasado ya un año y pese a que reclame mil veces la respuesta nunca llego. Sansung y dropbox estafadores.
- **Corpus MOV (Cruz Mata et al., 2008)** - Rank: 1 - La alianza del mal. Resumen: Silicona, esteroides, pactos demoníacos y otras basuras habituales son la base que sustentan esta aberración. De verguenza. Comentario: Una fiesta llena de excesos, rubias despampanantes, musculitos por doquier, algún que otro muerto. nada nuevo. La alianza del mal es el nombre de este Thriller sobrenatural que narra ... Porque la única realidad es que La alianza del mal es un desecho cinematográfico, una insufrible sucesión de imágenes cuya nica virtud es la de no durar demasiado. De vergenza.
- **Corpus HOTEL (Molina-González et al., 2014)** - Rank: 5 - Un hotel digno de mención! Como bien les comenté a los propietarios a la hora de abandonar el hotel, no dudaré un momento en recomendar una y otra vez el Hotel Albero de Granada. Su situación respecto del centro de Granada no es la mejor, pero para nuestros propósitos era perfecto (escapada de fin de semana con visita a la Alhambra). Se encuentra en la carretera de paso a Sierra Nevada y muy cercano a la Alhambra. Por la zona se puede encontrar aparcamiento y este se encuentra en una zona segura y tranquila. Los parkings del centro de Granada que nos recomendaron en el hotel fueron lo que nos dijeron (nada caros) y pudimos movernos por el centro perfectamente desde allí. Las habitaciones muy limpias y las camas confortables. El desayuno fue espectacular. Ya teníamos buenas referencias de este maravilloso hotel de una estrella (que para mí que viajo constantemente son más) pero ha superado con creces nuestras expectativas. Si vuelvo a Granada no dudaré en hospedarme en el mismo hotel. Muchas gracias por todo!
- **Corpus MED1 (Plaza-Del-Arco et al., 2016)** - Rank: 5 - Todo un acierto. Encantado de ser su paciente. Siempre se toma su tiempo para saber tu caso y resolver tus dudas.
- **Corpus MED2 (Jiménez-Zafra et al., 2017)** - Rank: NEGATIVE - En lugar de calmar el dolor por una infección dental, me da efectos secundarios desagradables y una horrible migraña al día siguiente. El Tramadol 5 es mucho mejor.
- **Corpus MISC1 (Vilares et al., 2015)** - Rank: 4 - @Chunjiram @LjoeRam Son tan adorables, ajkjgshdkjhegftws.
- **Corpus MISC2 (Martí Antonín et al., 2016)** - Rank: NEGATIVO - No lo he utilizado mucho.Actualmente (agosto 2002) tiene 22.800 km.Con solo 22.000, averia del embrague 70.000 ptas. Que dice Seat at.cliente? ha pasado la garantia no se hacen cargo,a pesar, que el mecanico me dijo que NO se debia a mal uso por mi parte. Lo arreglo. Resulta que el pedal, una vez arreglado, no sube del todo. Me lo vuelven a arreglar y actualmente sigue mal. Tiene ujnh ruido en los bajos que parece un concierto. No m atrevo a viajar.
- **Corpus MISC3 (Mori et al., 2016)** - Rank: POSITIVO - Uruguay se floreó en casa y está un poco más cerca de Rusia 2018 https://t.co/mu5UJJL4P SOYCELESTE @SoyCelestee https://t.co/tt8bpbEP2y.
- **Corpus JOUR1 (Mordecki et al., 2015)** - Rank: NEGATIVO - anuncio de Batlle sobre juicio a ex socios del Comercial. Batlle dijo que cuando recuerda la conversación con Mulford, del Credit Suisse, siente "ganas de matar".
- **Corpus JOUR2 (Boldrini et al., 2009)** - Rank: POSITIVE- Ahora vamos a lo nuestro, el tema del calentamiento global es uno de los más interesantes y conocer sobre el protocolo de Kyoto se hace sumamente necesario. Es por eso que en esta oportunidad les traigo un informe que publicaron los 147 amigos de la BBC Mundo que espero te sirva para conocer un poco más sobre esta situación que a todos nos involucra.
- **Corpus REST1 (Jiménez-Zafra et al., 2009)** - Rank: 1,3 - Caro para el conjunto ofrecido. Aunque los alimentos estaban bien cocinados no me pareció coherente el precio que se paga con lo ofrecido en general. Desde mi punto de vista, cuando pagas 45-50 persona hay muchas cosas que valorar y que deben ser coherentes con el precio: la forma de servir los platos, la cubertería, la mantelería, etc. Todo debe ser coherente con esos precios y en este restaurante no lo es: manteles personales de plástico, vasos de tubo para refrescos, presentación del producto anticuado, etc. Creo que pagar esos precio para el conjunto que ofrecen no es apropiado sino mas bien caro.
- **Corpus REST2 (Dubian, 2013a)** - Rank: NEGATIVO - Cada dia esta peor, precios para turistas, la entrada casi incomible. El mozo trajo primero la ensalada, despues las empanadas y la carne cuando se le ocurrio. Ya no hay tanta gente por algo sera. Dificilmente regresemos fuimos 5 personas y todos coincidimos.

Inteligencia Artificial 22(64) (2019), 14-35
doi: 10.4114/intartif.vol22iss64pp14-35

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Feature Learning with Multi-objective Evolutionary Computation in the generation of Acoustic Features

José Menezes[1,A], Giordano Cabral[1,2,B], Bruno Gomes[2,C], Paulo Pereira[2,D]

[1]UFRPE - Federal Rural University of Pernambuco, Recife 52171-900, Brazil
[2]UFPE - Federal University of Pernambuco, Recife 50670-901, Brazil

[A]joseantonio.menezes@ufrpe.br, [B]grec@cin.ufpe.br, [C]btmg@cin.ufpe.br, [D]prps@cin.ufpe.br

Abstract To choice audio features has been a very interesting theme for audio classification experts. This process is probably the most important to solve the classification problem. In this sense, techniques of *Feature Learning* generate attributes more appropriate for classification model. Generally these techniques do not depend on knowledge domain and can apply in various types of data. Yet, less agnostic approaches learn a knowledge restricted to the area studded and audio data requires a specific knowledge. Many techniques aim to improve the performance in generation of new acoustic features, among there is the technique based in evolutionary algorithms to explore analytical space of function. Despite the efforts made, there are still opportunities for improvement. This work proposes and evaluates a multi-objective alternative to the exploitation of analytical audio features. Experiments were arranged to validate the method, with the help a computational prototype implementing the proposed solution. Then it was verified the model effectiveness and was shown there is still opportunity for improvement in the chosen segment.

Resumen Elegir características de audio ha sido un tema muy interesante para los expertos en clasificación de audio. Este proceso es probablemente el más importante para resolver el problema de clasificación. En este sentido, las técnicas de Feature Learning generan atributos más apropiados para el modelo de clasificación. En general, estas técnicas no dependen del dominio del conocimiento y pueden aplicarse a diversos tipos de datos. Sin embargo, los enfoques menos agnósticos aprenden un conocimiento restringido al área tachada y los datos de audio requieren un conocimiento específico. Muchas técnicas tienen como objetivo mejorar el rendimiento en la generación de nuevas características acústicas, entre ellas, la técnica basada en algoritmos evolutivos para explorar el espacio analítico de la función. A pesar de los esfuerzos realizados, todavía hay oportunidades de mejora. Este trabajo propone y evalúa una alternativa multi-objetivo a la explotación de las características de audio analíticas. Se organizaron experimentos para validar el método, con la ayuda de un prototipo computacional que implementó la solución propuesta. Luego se verificó la efectividad del modelo y se mostró que todavía hay oportunidades de mejora en el segmento elegido.

Keywords: Automatic audio classification, feature learning, analytical space, evolutionary algorithms, multi-objective optimization.
Palabras clave: Clasificación automática de audio, aprendizaje de características, espacio analítico, algoritmos evolutivos, optimización multiobjetivo.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

1 Introduction

Automatic Audio Classification (AAC) is a subject of great interest to Music Information Retrieval (MIR) specialists. The process involves many concepts of computational intelligence, among them Feature Learning [1].

It is known the choice of good features is determinant for the efficiency of classification tasks, because they reasonably delimit each class of the problem [2], [3]. However, it is not always easy to determine the best features to solve a problem. Therefore, strategies for designing new features are important. There is an interest in Feature Learning techniques, such as: PCA (*Principal Components Analysis*) [4], ICA (*Independent Components Analysis*) [5] and *Deep Learning* [6], among others. Feature Learning is also important in reducing of extraction cost and overfitting, since it reduces the data dimension and allows the classification mechanism to generalize observations.

Due to the complexity of the real AAC problems, the process of composing acoustic attributes often becomes handcrafted, demanding specialized knowledge and design time. Consequently, analytical approaches are promising since they dispense specialist knowledge. They are scalable, less costly and are adaptable for reuse.

Beside providing good results, it is our interest that the search for good features be intelligible. Although the aforementioned Feature Learning alternatives are effective, they are also generic in the field of knowledge which they can be applied, being useful in a wide range of fields: image processing, video, sensors, etc. But the audio has particularities that suggest other alternatives, which can obtain significant gains in terms of classification performance, usability, project or execution time.

It is in this context that techniques of audio feature learning arise, such as the automatic exploration of the analytical space of acoustic attributes (Section 3). In this sense, EDS (Extractor Discovery System) [7] stands out. It uses genetic programming to explore an analytic space of functions and find new audio features. The technique evolves individually features and return the featureset explored by the search. This approach has stagnated, despite the evolution over the following years [8], [9]. Possibly due to advances of Deep Learning, which has been increasingly applied in MIR problems [10]. Deep Learning is feature learning capable and is understood in the context of this work as a Feature Learning alternative. However, it is presented as a black-box option, and it is not possible to extract meaning from the features learned. Then it not cooperate with specialists to understand what characterizes the classes of his problem.

In order to design intelligible acoustic features, we identify improvement opportunities in the EDS. They involve the use of multi-objective evolutionary algorithms, which can restore the innovative character of the solution and to remove from the inertia the analytical development of acoustic attributes.

Some AAC problems require the exclusive minimization of the false positives or negatives. Simple genetic algorithm methods, such as EDS, do not meet this need. By these methods, the goal can only be achieved by a side effect: the solution obtains good accuracy, consequently minimizing errors. We believe this particularity can be treated more narrowly with multi-objective algoritms.

In addition, EDS process involves two steps in choosing attributes: optimization of audio features and selection of the most relevant them. The use of multi-objective algorithms also possibility to simplify the process of choosing features, simultaneously performing tasks that EDS does in two stages.

For the purpose of to achieve less agnostic approaches of feature learning, we have the questions: "Is it possible to improve the performance of an audio featureset used in AAC tasks through the multi-objective optimization of this set?", "Can it help in the exclusive minimization of false positives or negatives?". This work proposes to answer these questions. It analyzes the potential of multi-objective algorithms in the generation of audio features, composed according to the EDS model.

It was necessary to develop a computational tool that implemented such requirements, because there is no such proposal in the literature. It was compared with a mono-objective technique (EDS-like) and another domain-independent feature learning technique (PCA). We were able to confirm the initial hypothesis that it is possible to obtain better results with the approach proposed. It was also possible to note the potential to improve objectives related to the matrix of confusion.

Experiment was carried out with a database of a real problem: shot recognition. It was verified that the use of multi-objective optimization of audio features can improve the accuracy of the classification model. Besides it being more adapted to the specific needs of the problems.

2 Problem Details

In applications that demand intelligence in digital audio manipulation, AAC has been widely used. Figure 1 presents an ontology of subproblems and their respective applications.

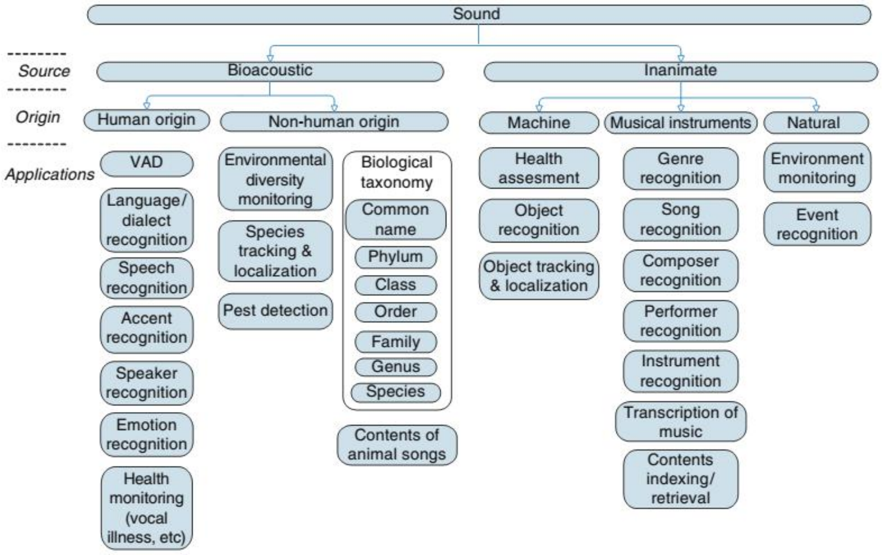


Figure 1. Taxonomy of sounds in the perspective of human applications [11]

The applications are in areas such as telecommunications, security, health, biology, music and so on. These applications include the detection of vocal activity [12], speech recognition [13], languages detection [14], speaker recognition [15], emotion recognition [16], health monitoring (vocal chords, respiration, etc.), localization of sound sources [17], pest detection [18], biological taxonomy [19], animal communication, audio monitoring, sound event recognition such as baby crying [20] and a range of musical applications. The latter are promising due to the increasing amount of musical content produced and shared. These applications include the recognition of musical styles [21] music recognition, recognition and modeling of composers styles, detection of musical instruments [22], separation of sources, musical transcription (chords, notes, rhythms, attacks, rhythm, tonality) [23] and so on [11].

With regard to the implementation of a new classifier model, three steps are involved: selection / extraction of audio features, training and validation. In the selection of features, the sound must be represented by a set of attributes (wave frequency, frequency power, spectral, etc.) [24]. In traditional methods this step works with pre-defined audio attributes. After that, those which best represent problem classes are chosen to train a new classifier using some audio base. Finally the obtained model is validated using another base.

Choice of good features is decisive for the mapping between audio samples and classes of the problem. However, some Feature Learning strategies are particular since some audio attribute do not apply to images, texts, or any other form of data.

Classifier, in turn, is chosen according to its efficiency. But the determination of which is the best one is not exact and depends on the nature of the problem, as well as sound attributes involved. To search for the best classifier one can optimize the hyper-parameters of classification algorithms, a solution investigated by [25], which employs SMAC (Sequencial Model-based Algorithm Selection) [26] to find the best classifier and fit its parameters.

The efforts required in feature selection and classifier selection result in two distinct approaches:

- *Bag-of-Frame* [27], [28]: Here the sound samples are divided into frames. For each frame a feature vector is computed. These vectors are aggregated (hence the bag) and used in the rest of the process: selection of feature subset, training and validation of the classifier. Currently this approach serves a wide range of problems such as musical genre classification, instrument recognition, nasality detection, voice or mood identification; etc.

- *Ad-hoc* [8]: Although Bag-of-Frame to be efficient in many cases, problems of lesser abstraction are more difficult to solve with it. For example, it is easy to distinguish between Rock and Jazz, but it is difficult to distinguish between Be-bop and Hard-bop (subgenres of the Jazz), because there is a lot of similarity and subtle differences between styles. It is hard even for humans. An Ad-hoc approach aims to devise a new features set that make this distinction possible. This can be done by applying two or more functions on the signal (e.g. applying a filter on the signal and then an FFT to get the maximum power). Disadvantages are that this process requires specialist knowledge, is costly and is by trial and error. Moreover, reuse is rare. Therefore the analytical features obtained by this process are unlikely to serve another problem.

It is in this context that techniques of automatic generation of features gains importance, especially when they use evolutionary algorithms [29], [30], [7] and achieve optimal solutions with less computational effort.

2.1 Analytical space of audio features

In an audio classification approach that uses generic features (preexisting high-level attributes, e.g.: Zero Crossing, Root Main Square – RMS, Mel Frequency Cepstral Coefficient – MFCC, etc.) [24] there is no concern with evolution these features. Designer only selects those most relevant using dimensionality reduction techniques. We will call the preexisting feature set of generic space.

Mean difference in the use of analytical features, in contrast to the previous approach, is that they consist of attributes generated from the combination of two or more generic features. Interest is to improve their performances through analysis. So another approach would be the *heuristic search in the analytic space*, which aims to find new audio features starting from a generic set.

As the analytical space has high complexity, not every method is feasible to search for the optimal attributes. Evolutionary algorithms are interesting option. In audio feature learning, the use of Evolutionary Computing is still very little explored. However the theme is comprehensive and its potential in AAC has been neglected. Results from theme exploration are particularly interesting because they are specific to the knowledge domain of MIR such as these used by EDS (Extractor Discovery System) [31], [7], [8].

There are three challenges to AAC that are relevant to this paper. They are: the need to find meaningful features; the difficulty in designing them and the possibility of improving in Evolutionary Computing as a method of solving these problems.

The *need* to find new audio features is due to preexisting ones often do not satisfactorily solve restricted classification problems. It is necessary, but costly, to design manually acoustic features more appropriate to the nature of the problem. This artisanal design do not guarantee of satisfactory solutions. And even if one arrives at good attributes, they can hardly be reused in another classification problem. Hence the need for an automatic and fast method of search these features.

Main *difficulty* involves the infinity search space [8] since it is possible to combine generic features and modify its parameters countless times. This makes the optimal solution unlimited and its viability is its viability is method-dependent. In the end, this task can be summarized in a process of trial and error, even if computational methods are used. In this sense, evolutionary algorithms are good options to approach the optimal solution, due to its heuristic nature.

In addition to the previous problems, it is assumed that it is also possible to improve the efficiency in solving any classification problems through the search for new attributes. Because in an audio classification problem, the search space is infinite, so hypothetically, it is possible to find better features than the current ones [8]. Indeed this hypothesis was validated in the works related to EDS [31], [7], [8].

Implementations such as EDS are *possible to improve*. They use a mono-objective algorithm and lack an objective way of treating peculiarities. Often AAC tasks are not only interested in improving the accuracy of the results, but solving the problem in such a way that certain constraints are met. Failure to meet these constraints may even have serious consequences, depending on the application [25]. Take a health monitoring system: the system may even fail to emit a false alarm about a patient's situation, but, depending on what is being monitored, can not fail to send the true alarm in the occurrence of some abnormal event . Thus, the system accuracy is not the main aspect to be improved. In this case the optimization should be directed to reduce errors related to true alarms. And the features that must emerge from the search process must be able to define precisely the critical events, satisfying the constraints.

Moreover, EDS approach needs a complementary process. At the end of the search there is a considerable number of attributes. But it is still necessary to select the most relevant ones. Selection methods are still required. In a strategy that seeks to evolve the feature set and not only one, at the end of the process already it has the best feature set. No post-optimization selection methods are required. Multi-objective evolutionary algorithms are presented as a good alternative to these questions.

2.2 Expected Solution

We expect that a solution will find a set of intelligible attributes for AAC tasks in a viable time. The solution must use evolutionary multi-objective computational techniques.

Among other satisfaction criteria to be achieved are the following:

- *Correctness*. Strategy of optimizing must to improvement accuracy of the classifier model;
- *Adequacy*. Expected solution must to satisfy the constraints of the problems .
- *Reusability and scalability*. It is unusual to apply the audio features of one problem to another, but technique to find them must be adaptable to problems of any kind and scale;
- *Economy of knowledge*. Solutin must be independent of specialized knowledge about signal processing. It would be more accessible to developers of intelligent audio technologies.

We did not find in the state of the art a solution that contemplates all these criteria. Thus we propose alternative one. The following sections present the state of the art, the proposed solution and its validation.

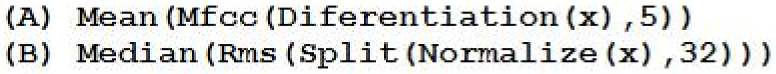
3 State of Art

This section presents Extractor Discovery System. The only solution within the scope of the research: evolutionary computation methods for analytical space exploration of acoustic functions.

3.1 Extractor Discovery System (EDS)

The EDS was developed in the laboratory of Sony CSL in Paris and presents a good proposal in audio feature learning using genetic programming techniques [32].

Starting from a finite set of elementary operators, e.g.: Mathematicians (addition, multiplication by scalar, mean, etc.); signal processing (Fourier transform, filters, spectral centroid, etc.); specific for music (Pitch or ltas), it is possible to combine these operators in a valid way. It obtain expressions as in figure 2 in which (A) can be understood as *the mean of the first five ceptral coefficients (MFCC) of the derivative of the signal 'x'.* And (B) is *Median energy value (RMS) of 32 splits of the normalized signal 'x'.* Combination of these operators defines a new audio feature, also called a function.



```
(A) Mean(Mfcc(Diferentiation(x),5))
(B) Median(Rms(Split(Normalize(x),32)))
```

Figure 2. Feature example [8].

3.1.1 Typing Rules and Heuristic

Design of functions is controlled by two mechanisms: typing and heuristics. Typing rules control combination of operators guaranteeing that their inputs and outputs have the same data types. For example, an FFT will receive an input an acoustic signal and will transform into a spectral output, or vice versa. A Mean receives any sequence of information and transforms it into a scalar. Thus, EDS can generate: *fft(HpFilter(x))*, but not *fft(mean(x)*.

Heuristics represent the specialized knowledge of signal processing professionals. It allows to bet on some functions without having to calculate their performance. It is also characteristic of the mechanism to prevent formations of unnecessary functions, such as *fft(fft(fft(fft(x))))* [7], [8].

3.1.2 Generic Operators and Pattern

Typing system allows the creation of *generic operators* (it is not generic feature), which are regular expressions that support one or more operators and form functions whose type of output is forced [7]. For example: the operator “*_a (x)” suggests a combination of several operators whose output type is a scalar “a”. In the case *Square (Mean (x))* is a valid function to satisfy the operator.

EDS implements three generic operators:

- "?_T" points to 1 operator whose output type is "T".
- "*_T" points to several operators whose output types are all "T".
- "!_T" points to several operators whose only final output is of the type "T"

It's possible to define patterns of functions as: “?_a (!_Va (Split (*_t:a (SIGNAL))))” which supports the following functions and so on [7]:

- *Sum_a (Square_Va (Mean_Va(Split_Vt:a (HpFilter_t:a(SIGNAL_t:a, 1000Hz), 100))))* or
- *Log10_a (Variance_a (NPeaks_Va(Split_Vt:a (Autocorrelation_t:a(SIGNAL_t:a), 100), 10)))*.

3.1.3 Mechanisms of Genetic Algorithm

Genetic algorithm seeks to "evolve" a population of individuals in order to find those most apt. For this it uses operations of recombination, mutation and selection. In the context of this work, given a random population of audio features (individuals), it is possible to improve the quality of these attributes by applying successive genetic operations.

Operations are shown through the examples that follow.

Starting from the expression *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))* the system executes the following operations:

- Cloning – It changes parameters of some functions.
  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  After: *Sum (Square (Mean (Split (HpFilter (SIGNAL, **430Hz**), **65ms**))))*
- Mutation (head or tail) – It changes part of head or tail of the expression.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  After: ***Max (Max** (Split (HpFilter (SIGNAL, 430Hz), 65ms))))*

- Deletion – It removes any function from the expression.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  After: *Sum (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms)))*.

- Addition – It adds any function to the expression head.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  After: **Log** (Sum (Square (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms)))))

- Replacement – It swaps one function for any other with equivalent type.

  E.g. Before: *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  After: *Sum (Square (Mean (Split (**LpFilter** (SIGNAL, 500Hz), 50ms))))*

- Cross-overs – It recombinates two individuals to generate a new one.

  E.g. A possible result of the cross between *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))* and the expression *Mean (Autocorrelation (SIGNAL))*, can be the two individuals shown below: 1º - *Sum (Square (Mean (Split (Autocorrelation (SIGNAL), 50ms))))* and 2º -*Mean (HpFilter(SIGNAL, 500Hz))*

It is not scope of this work to explain the operators of the examples showed. Many of them can be found in [24].

In addition to the operators shown, EDS defines algorithm meta-parameters (number of generations, population size and etc.). Finally, to determine when a feature in question is better than another, the system defines the fitness metric. It uses Fisher Discriminant Ratio [33] or some classifier model that, according to its accuracy, an fitness is attributed to input feature.

3.1.4 Global Algorithm

Implementation of EDS is organized in two parts: Learning of new attributes by genetic algorithm and the selection of relevant features resulting from this process. Figure 3 illustrates this, while figure 4 presents pseudocode of the algorithm implemented by EDS.

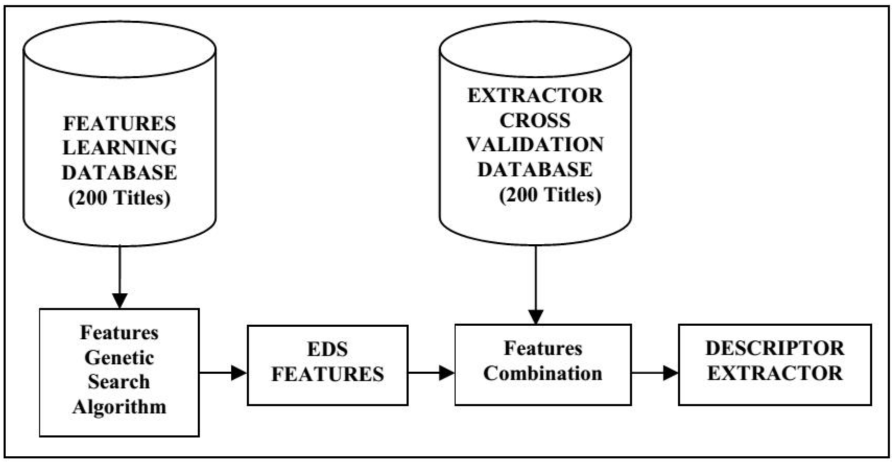
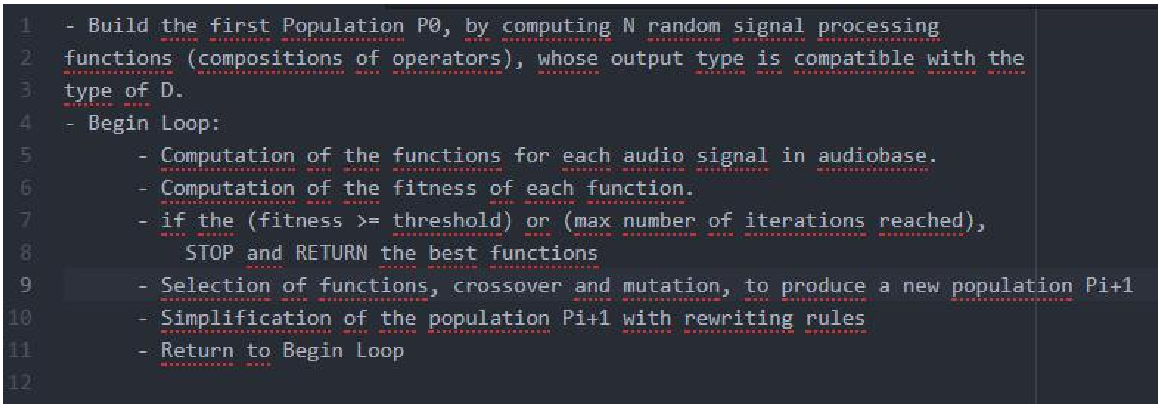


Figure 3. Global architecture of the Extractor Discovery System [7].



```
- Build the first Population P0, by computing N random signal processing
functions (compositions of operators), whose output type is compatible with the
type of D.
- Begin Loop:
      - Computation of the functions for each audio signal in audiobase.
      - Computation of the fitness of each function.
      - if the (fitness >= threshold) or (max number of iterations reached),
          STOP and RETURN the best functions
      - Selection of functions, crossover and mutation, to produce a new population Pi+1
      - Simplification of the population Pi+1 with rewriting rules
      - Return to Begin Loop
```

Figure 4. Global algorithm of the EDS [7].

3.1.5 Solution requirements met by EDS

According to the satisfaction criteria for the expected solution (Section 2.2) EDS improves the classification results (*correctness*); it serves for different classification problems (*reusability and scalability*); it can dispense specialized knowledge in execution (*economy*). However, it fails in the *adequacy* criteria, since it does not allow the development of solutions with satisfaction of constraints.

For AAC problems that need to satisfy constraints a solution is to use multi-objective genetic algorithms.It evolves the accuracy of the classification at the same time that it fits the constraint to be satisfied. For example, in a voice identification problem to access control of some system. It is tolerable that rarely access control fails to identify the voice of a registered person and denies permission (false negative), but it is intolerable that the system allows access to any unregistered person (false positive) due to a failure in their voice identification process. Thus, beside to improve the classification accuracy it is also necessary to reduce the occurrence of false positives (or false negatives, depending on the nature of the problem). A simple genetic algorithm does not make it possible to improve these two aspects of the same problem. But multi-objective strategy allows to seek both aspects during the process.

In addition to the satisfaction criteria, another gap left by EDS is that features are evaluated in isolation. Those with low fitness do not survive throughout the iterations, but there is a possibility that features lead to better classification results when are combined with others. In order to avoid wasting features with low fitness, EDS can save the generated attributes list by the genetic algorithm and apply an attribute selection technique. However, this approach does not allow low fitness features to interfere in evolution of another feature during algorithm iterations. In this way it is necessary to guarantee their survival, which suggests the individual fitness is not the only aspect to optimize by search.

Furthermore, we understand that it is possible to simplify the procedure by eliminating the Features Combination step (Figure 3).

In short Extractor Discovery System can be improved because of the following motivators:

- It do not use heuristics to satisfy false positive or negative constraints;
- It do not preserve, during genetic programming, audio attributes with low fitness;
- It is possible to simplify the process by eliminating one step.

We present a solution for automatic audio classification, using analytical attributes resulting from a multi-objective search, in order to satisfy the points listed.

4 Proposed solution

4.1 Elements

Although EDS have good results, we consider possible to improve it, either in the classification accuracy or in the constraints satisfaction. Fundamental difference between mono-objective and multi-objective optimizations in this work is that the former considers only the isolated audio feature and the latter considers a feature set. This is important because it determines how a population of individuals evolves over the generations of the algorithm. In mono-objective approach, the features with low fitness do not survive throughout the iterations, but it could be possible that these features achieve to better results when combined with others. A multi-objective approach does not allow features to be discarded because of their low individual fitness. By surviving, they can contribute to better collective outcomes.

EDS sets each individual as an audio feature. In our proposal, we represent the individual as a feature set. Figure 5 shows the isolated fitness attributed to each expression in a similar manner to EDS and collective fitness assigned to the function set. The *n-th* expression has a fitness of 0.32 (32% is its accuracy) which can easily be surpassed by others. But when combined with the others it will contribute to a collective accuracy of 0.88 (88%) and can be better than anyone of all other isolated functions. The additional fitness mesure can be relevant for the evolution of the features, since the bad feature would not persist in isolation, but because it is part of a set it survives and contributes to improve of set.

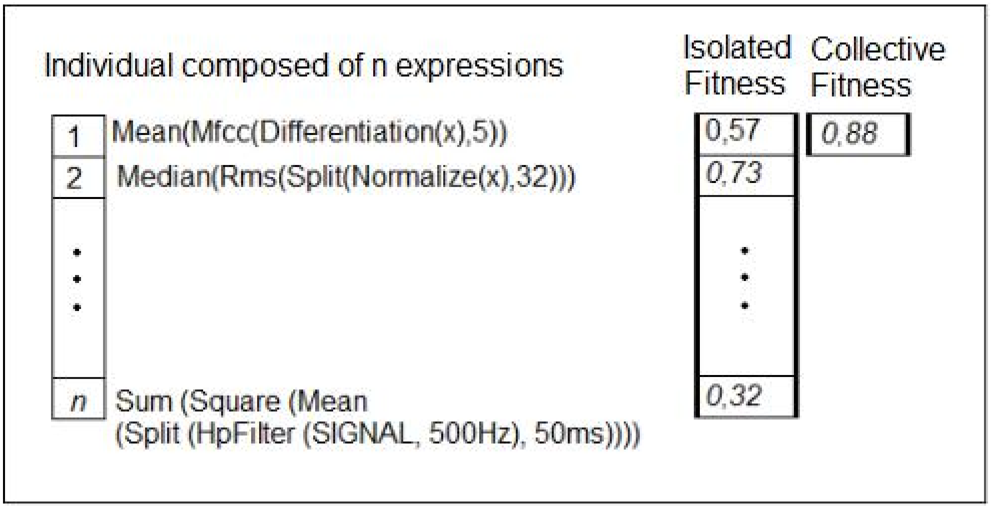

Individual composed of n expressions
Isolated Fitness
Collective Fitness
1 Mean(Mfcc(Differentiation(x),5))
2 Median(Rms(Split(Normalize(x),32)))
n Sum (Square (Mean (Split (HpFilter (SIGNAL, 500Hz), 50ms))))
0,57
0,73
0,32
0,88

Figure 5. Ilustrative example of individual multi-objective solution and its fitness measures (between 0-1).

New evolutionary operations were devised to match with the new form of individuals representation. For example, crossing functions between an individual and another or, remove or add functions. Thus, we adapted the mono-objective operations as follows:

From an individual with three genes (functions): *1 - Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms)))); 2 - (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms)))) and 3 - Mean (Autocorrelation (SIGNAL))*, the system performs the following operations:

- Mutation (head or tail) – It changes entirely part of the head or tail of an individual, swapping by any other expression. E.g.:
  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  - ***Median(FFT (SIGNAL, 500Hz), 50ms))***
  - ***RMS (Normalize (SIGNAL))***
- Deletion – Removes any expression from the set. E.g.:
  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  - *Mean (Autocorrelation (SIGNAL))*
- Addition – Adds any expression to the set. E.g.:
  - *Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))*
  - *Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms)))*
  - *Mean (Autocorrelation (SIGNAL))*
  - ***RMS (Normalize(SIGNAL))***
- Replacement – Swaps an expression for any other. E.g.:
  - ***RMS (Normalize(SIGNAL))***
  - *Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))*
  - *Mean (Autocorrelation (SIGNAL))*
- *Cross-overs* – Recombinates two individuals to generate a new one. E.g.: Crossing with the individual of two genes: *A - RMS (Normalize(SIGNAL))* e *B- Median(FFT (HpFilter(SIGNAL, 500Hz), 50ms))))*, can result in two new children:

  Child 1:

  A - ***RMS (Normalize(SIGNAL))***

  2 - Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))

  3 - Mean (Autocorrelation (SIGNAL))

Child 2:

1 - Sum (Square (Mean (Split (HpFilter(SIGNAL, 500Hz), 50ms))))

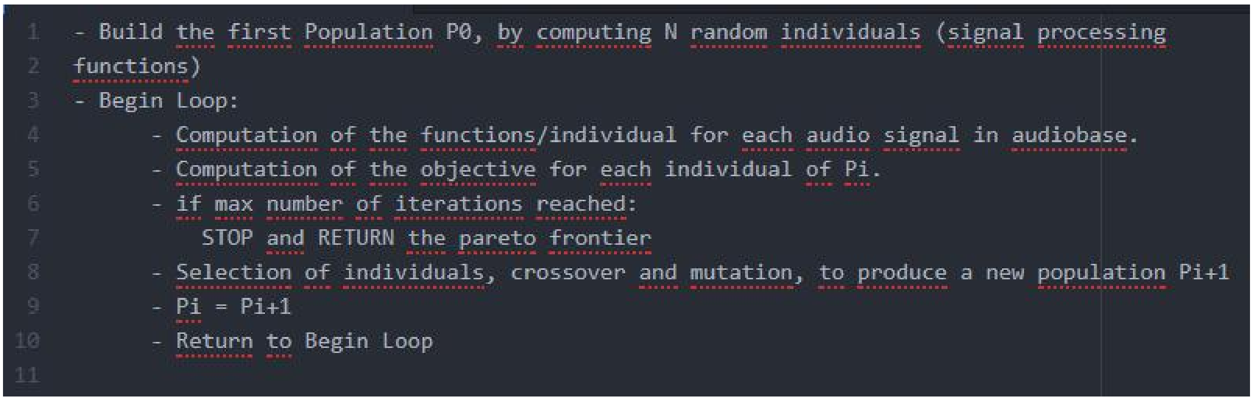
B - ***Median(FFT (HpFilter(SIGNAL, 500Hz), 50ms))))***

In addition, the mutation operations of the EDS approach could still be used as a particular type of mutation in the new algorithm since they act on a single expression.

We also conceive a restriction. It was found that throughout the generations the individuals grew tedious to be bigger, by the action of the recombinações, making the solution slow and costly. We try to circumvent this tendency by allowing a limit on the size of individuals, which are penalized when they exceed this limit. In this way, the method tends to evolve by keeping the size of individuals up to a certain threshold chosen by the user.

In what concerns the measurement of aptitude, we use the same mechanism of evaluation of individual of the previous implementation, the instance of a classifier to verify the results of fitness. Two or more goals can be defined for the problem. One of them being necessarily "increase hit rate". And the others being any of: "increase the fitness of the best feature of a individual", "increase the fitness of the 'worse' feature of the individual", "increase the distance between the fitness of the best and worst feature of the individual", "Decrease the distance between the fitness of the best and the worst feature of the individual", or restrictions such as "decrease in the number of false positives or negatives of the individual", among other objectives that can be designed specifically for each problem instance of AAC.

As these are objectives that may be in conflict, the result will be a Pareto frontier [34] (figure 6). What matters is the individual (set of audio features) that best fits the nature of the problem, that is, it will be up to the professional to decide, starting of the results achieved, which solution of the border is most appropriate for their problem.



```
- Build the first Population P0, by computing N random individuals (signal processing
functions)
- Begin Loop:
      - Computation of the functions/individual for each audio signal in audiobase.
      - Computation of the objective for each individual of Pi.
      - if max number of iterations reached:
          STOP and RETURN the pareto frontier
      - Selection of individuals, crossover and mutation, to produce a new population Pi+1
      - Pi = Pi+1
      - Return to Begin Loop
```

Figure 6. Global algorithm multi-objeticve.

In a problem of satisfying constraints, in the reduction of false positives or negatives, the result must be the Pareto frontier, being at the discretion of the developer to define which of the non-dominated individuals should be used as a solution to their problem. But in case the problem has no restrictions, what interests us is only the accuracy. Therefore, if the best value is reached by the first objective (collective fitness) of an individual then it should be used in the classifier model, but if the best value is reached by the 2nd goal (isolated fitness) of some gene (audio feature), then only this feature should be used.

4.2 Prototype

With the design of the aspects presented previously, the proposed solution covers what was not covered by the EDS, however it remains to know how effective an implementation can be that uses our approach. For this, a computational prototype was developed which we call ExpertMIR.

4.2.1 Architecture and flow

Initially ExpertMIR was developed with an EDS-like system, seeking to implement the features of the Extractor Discovery System in order to better study the behavior of such an approach, this was necessary because Sony's solution was closed, making it impossible to try. In a second moment, it was proposed an evolution of this system, using multi-objective algorithms in order to achieve some improvement.

Thus, the solution was constructed to operate in two perspectives: in the mono-objective and multi-objective optimization of audio features. Both operating on the same analytic space, having in common the operators and patterns used to generate individuals in genetic algorithms.

ExpertMIR can explore features that follow the following standards:

- " !_a (SIGNAL)" – any function whose final output is a scalar;
- " Mean(!_t:a (SIGNAL))" – the average of any function whose output is in the time domain.
- " Mean(!_f:a (SIGNAL))" – the average of any function whose output is in the frequency domain.

Also the same technological framework was used for both: the extractor system, the classifier algorithm among others. Figure 7 shows how the system modules are designed to interact.

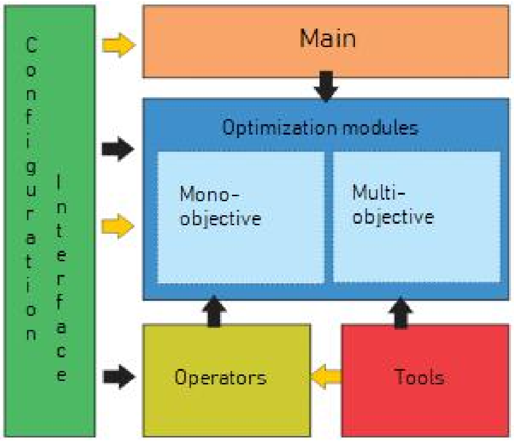


Figure 7. Architecture of the prototype's modules.

Configuration interface: Set of classes that allow the non-expert user to configure a new problem (classes, audio base, constraints or goals and so on), define the perspective in which the system will operate, have files and reports to record the result of the process.

Main: It gathers all possibilities of tool execution, controlling what can be done and how it should be done. Here the problems are instantiated and the calls of the evolutionary algorithms are made.

Mono and multi-objective optimization modules: They are the main modules of the system, responsible for executing feature learning, they are never executed at the same time. Because of the complexity and processing cost of genetic algorithms, it is important that each approach can be performed separately. In addition, not overloading the computational resource also allows better evaluation of the performance of each algorithm.

Operator Package: Here, each mathematical, signal processing and music-specific operator is implemented, which is the set of operators that determine the analytical space of expressions that the genetic searches will follow. This module of the system is scalable, it can, whenever it is necessary to add as many operators as you like, as well as activate and deactivate an operator.

Tools Package: In order to execute the genetic algorithms well according to our interest and necessity, it was necessary to develop a package of support to the algorithms, in this package it is possible to resort to several inherent tools of the process, like: algorithm classifier (used in the calculation of the fitness of individuals) (extracts the values from the converted audio samples), features validator (responsible for checking if the feature found in the search is being correctly constructed) and so on.

To understand the various transformations of data throughout a system run, the execution flow diagram is shown in figure 8.

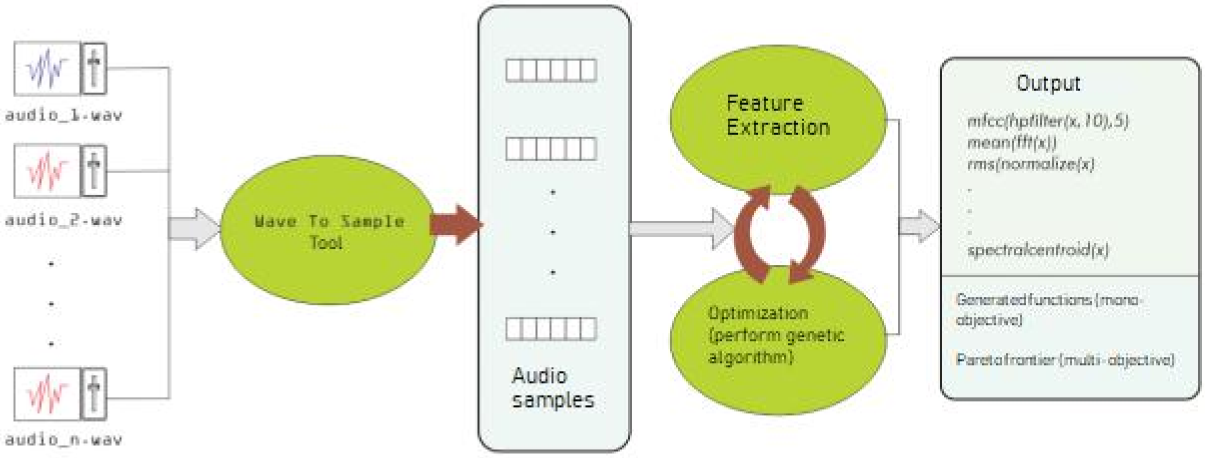


Figure 8. Flow of data manipulated in the execution of the prototype.

When looking for new audio features the solution requires an annotated audio base, since the learning is supervisioned. Files in the WAVE format representing instances of the classes of a given problem are preprocessed and represented as system objects. This is necessary to make the process faster, as, throughout the genetic interactions of learning, the information contained in the samples will be frequently requested.

The second moment is the most important step of the process and occurs with two modules constantly exchanging information. The optimization module, each generation of a population, will require the feature extractor to calculate the value of the features generated for each sample of the base. This information should be used in the individual's fitness calculation, which uses a classifier instance to perform cross-validation in the annotated database. This allows to find out how much the features found are adequate for the solution of the problem.

Finally, the result of the process is the set of features learned by the mono-objective algorithm, or the set of non-dominated solutions (Pareto frontier) of the multi-objective algorithm.

4.2.2 Technologies used

ExpertMIR was developed in Java 8 EE through the Eclipse IDE. The choice was due to the good options of complementary tools in this language and that could be used by ExpertMIR. These tools are shown below:

JMIR is a suite of Java open source applications for MIR searches. It was proposed by Cory McKay [35] in order to offer wide support the most varied applications of MIR. The jMIR allows the manipulation of the acoustic signal in digital signal format (Wave, MP3 and etc.) as well as in symbolic format (MIDI) and serves the most diverse applications as: audio data mining in the Web, audio classification, and so on. It is a complete tool in terms of MIR and reference in the.

Precisely it is the automatic audio sorting feature makes us interested in this tool. An AAC problem is solved by using two of its modules: jAudio and ACE. Each can be understood for the following purposes:

- *jAudio Feature Extractor* [36]: Module responsible for extracting audio features. It contains a set of audio and signal features (FFT, MFCC, normalization, compactness, histogram, etc.) and the possibility of saving its values for each sample of the problem in jMIR, ACE XML standard format or until even in Weka's own ARFF format [37]. Among the other features of the tool are, for example, the possibility of audio recording, execution, adjustment of sampling rate, fragmentation and normalization of the signal. It was used in this work to assist in the extraction of features because it has several of them already implemented.
- ACE (*Autonomous Classification Engine*) [2]: It is the jMIR Machine Learning module. Responsible for applying classification algorithms to the values extracted by jAudio and associating them with categories. In addition, it implements seven types of classifiers: k-NN, Naive Bayesian, Decision Tree C4.5, Multilayer Perceptron, Support Vector Machine, Adaboost and Bagging with C4.5. And also three

dimensionality reduction techniques: PCA, exhaustive search and genetic search. It was necessary in this work to instantiate the classifier used in the calculation of fitness.

In ExpertMIR, jMIR 2.4 was used, which had many audio features coded in its extractor module and also classification algorithms.

JMetal [38] is a Java framework for the development of multi-objective applications with metaheuristics. It also supports mono-objective heuristics and was useful in the specific activities from evolutionary algorithms. It was up to us to design the individual representation, the definition of the operators and the method of evaluation of individuals. JMetal still has versatility in the alternation of its algorithms (NSGA II, SPEA, etc.), which can easily be replaced and tested for when one wants to infer the impact of each one on the quality of the features generation. The version used in our implementation was 4.5.

5 Evaluation

This chapter aims to present and discuss the results obtained with the experimentation of the proposed solution. It will be describes how the tests were performed and the validation of the hypothesis that "it is possible to improve the performance of a set of audio features used in AAC activities through the multi-objective optimization of the feature set" (Section 1).

5.1 Methods

The problem of AAC to classify very similar audios needs a set of features in such a way that these audios can be differentiated in the best way. As pointed out in our state of the art, mono-objective optimization techniques have been promising in generating analytical features. However multi-objective techniques, which have not been proposed in the literature, suggest an improvement in the efficiency of feature learning, besides allowing better adaptation to problems with constraints.

This raises our research question: Is it possible to improve the performance of an audio featureset used in AAC tasks through the multi-objective optimization of this set?

The performance of the features was determined based on two aspects: 1 - the accuracy obtained through the set these; 2 - the measure of sensitivity or specificity when minimizing false negative or positive, respectively.

To answer this question, we organized an experiment. The criterion for choosing that problem was for two fundamental reasons: to be related to a real situation difficult to solve due to the great similarity of the classes of the problem and the availability of the audio bases.

It is hoped to show that the use of multi-objective genetic algorithms has a strong indication of effectiveness among other learning techniques, which does not mean that the solution of this work must always overcome others, but it arouse the interest of the scientific community for exploration possibilities would already be an excellent contribution.

The techniques in comparison were three: the search in the analytical space with an evolutionary algorithm mono-objective (algorithm EDS-Like), the search in the analytical space with multi-objective evolutionary algorithm (ExpertMIR) and the Feature Learning with PCA (jMIR). Although we are particularly interested in evolutionary strategies, we find it important to compare with some alternative outside this group in order to situate the solution among the options of the area. The choice of PCA was mainly due to its implementation being present in the technological support framework of this work, jMIR.

To ensure consistency in the comparison of approaches, a few points had to be equated. First, the quantity and type of operators used by both. A total of 9 operators were used among the mathematical and signal processing groups listed below:

- MFCC
- FFT of binary frequencies
- Normalization
- *Power Spectrum*
- *Magnitude Spectrum*

- RMS
- *Zero Crossing*
- *Spectral Centroid*
- *Spectral Roll-off*

These operators are the generic features from which will result in the analytical features of each learning process. The description of these agents can be found throughout many others in [24].

Second, it was necessary to set the maximum size of a feature for the processes that involve genetic algorithms. We consider it appropriate to adopt the size employed by [8], ten operations.

Finally, for the evaluation of the techniques it was necessary to use the same type of classification algorithm, K-NN with k = 1 and the same validation technique, cross-validation, which according [39] it is the best method of data sampling to be used in the evaluation of a classification model.

Having defined these aspects, we can assume a fair comparison between the three solution models for the problems.

Table 1. Methods used incorporating the approaches analyzed. GA: Genetic Algorithm.

Tag	Method
PCA	PCA with 1-NN classification algorithm
MO	GA mono-objective with 1-NN classification algorithm
MT1	GA multi-objective with 1-NN classification alg. (Obj2 = reduces false negative)
MT2	GA multi-objective with 1-NN classification alg. (Obj 2 = increase the accuracy of the best feature)

Table 1 shows the algorithms used to compare the techniques. We can observe that MT1 is the evolutionary algorithm whose second objective is to reduce the incidence of false negatives, which means to improve the sensitivity of the solution. MT2 is the algorithm whose second aim is to increase the accuracy of the most fit feature of a multi-objective individual. Both methods implement the solution proposed in this work, however, these different choices regarding the second objective were to see if there is difference in choosing as a secondary objective the decrease of false negatives (or positives) when the nature of the problem so requires. For this, it is necessary to compare with another method that operates with any other secondary objective (in this case MT2 seeking to optimize the accuracy of the most apt feature).

According to the Central Limit Theorem, where the distribution of sample means tends to a normal distribution as sample size n increases. Because of the size of the databases involved, we performed 10 times the first experiment (n = 10) and 30 times the second (n = 30), obtaining four samples with data about the accuracy and sensitivity of the methods and performed a series of tests through the R Studio tool [40].

For the methods with evolutionary algorithms (MO, MT1 and MT2), p = 15 and e = 1500, where p and e are, respectively, the size of the population and the number of evaluations to be made. The recombination and mutation rates were respectively 90% and 5%. MO implements a simple genetic algorithm, while MT1 and MT2 perform the NSGAII.

5.2 Experiment I: Monitoring of Environments and Safety (MES)

A security system for monitoring environments is able to recognize alert situations such as firearm shots, vehicle collision, shattered glass, shouting and so on. The audiobase used in this experiment is a closed real base, used for commercial purposes and has up to 18,000 examples labeled of the most varied types of sound for alert and safety.

Such a solution requires a wealth of knowledge covering geolocation, sound and image capture, coding, data transmission, image processing, compression, storage, and so on. Among them the classification of sounds is a fundamental activity.

For this work, the problem addressed had a balanced subset of 1,900 instances of this already fragmented base. The situation chosen is to distinguish sounds between two classes: 1 - Firearm shooting; 2 - Burst of fireworks. The strong similarity between these classes makes the problem rather difficult to solve, which suggests the use of feature optimization to improve the results of the techniques used to solve the problem. A similar question was addressed by [41] and [42] with the difference that they seek to distinguish between gun sound and any other sound. In our case, we approach a more specific problem by choosing fireworks as the second class of the problem, by the similarity between them, which is so large that classification is not easy even for the human ear. [41] and [42] go through a traditional path, selecting the most relevant generic features of a set.

In this experiment, 950 firing instances of various types of firearms (revolvers, pistols, machine guns, rifles, rifles, etc.) of various calibers were collected as opposed to 950 instances of fireworks bursts. Each with a sampling rate of 48 kHz and an average duration of 0.085 seconds.

5.2.1 Results

As regards the accuracy of the solutions, the T-test was performed for each of the samples and it was found that MT1 presented the best mean (90.67%) and it was also responsible for the highest accuracy found (92, 68%). The results obtained are illustrated in the graph of figure 9, although MT1 has the best results, MT2 closely resembles it (average accuracy = 90.57%), better accuracy = 91.74%). And although the maximum obtained in MO (91.05%) approaches the multi-objective algorithms, it is perceived that its average is just below (85.83%).

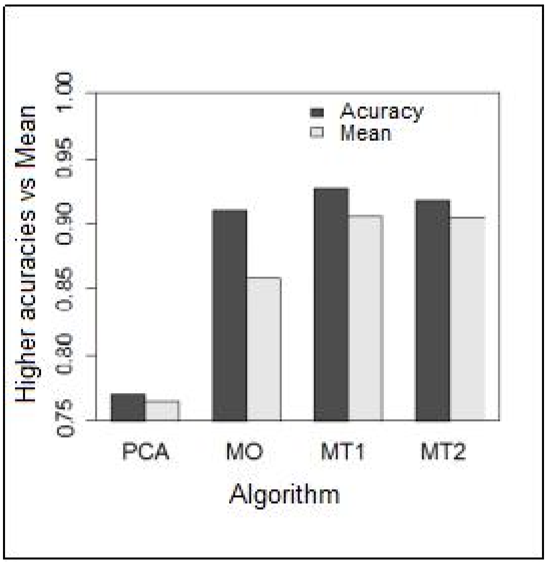


Figure 9. MES. – Maximum recorded accuracy x accuracy mean of methods.

Table 2. MES. – T-tests for each method + maximum accuracy recorded.

	Higher accuracy (%)	Mean (%)	Confidence interval (%)	Significance (%)	p-value
PCA	77	76,46	76,34– 76,6	5	2.2e-16
MO	91,05	85,83	83,67 – 87,99	5	1.34e-16
MT1	92,68	90,67	89,51 – 91,83	5	2.2e-16
MT2	91,74	90,57	89,99 – 91,15	5	2.2e-16

The boxplot (figure 10) allows us to more clearly visualize the empirical distribution of data. We can observe where the most relevant part of the sample data is concentrated and how similar the methods are studied. We observed that MT1 has slightly greater variability than MT2, and also slightly different interval, mean and median, and these solutions are similar. In the parameters chosen, the variability shown in the graph closely approximates the confidence intervals of the tests, and it is possible to be guided by it to understand the test. It is also possible, observing the interval of the methods box, to understand behaviors such as the possibility of a given being outside the mean and how far it can distance itself from it. For example, the MO test indicates that in

95% of cases its accuracy will be between 83.67 and 87.99 (fourth and fifth column of table 1), but the range of MO in the boxplot is even larger and this explains the appearance of 91.05% (its best recorded accuracy) as a predictable event. Values outside this range would be discrepant and in this case it would not be guaranteed to obtain them by repeating the experiment.

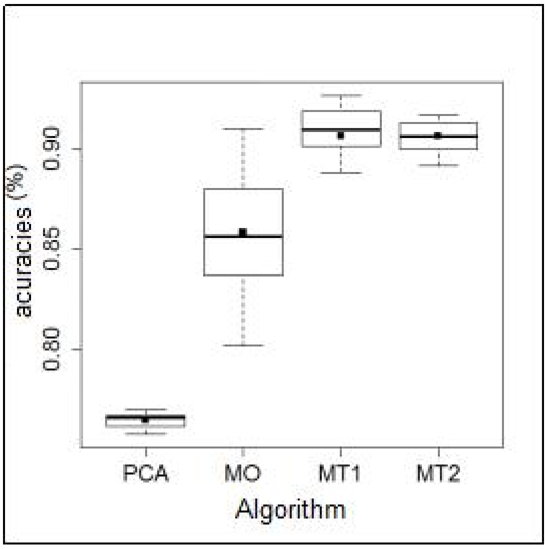


Figure 10. MES. – Boxplot of the accuracy of the methods.

The unilateral binomial test at right [43] was applied to know if, in fact, MT1 is better than MT2. We obtained the following values:

- Threshold of 50%
- Significance α = 0,05 (5%)
- p-value = 0.05469

The test reveals that possibly in more than half of the cases MT1 exceeds MT2 and still gives the probability of 80% chance of MT1 being better. Figure 11 provides these ordered data so that you can visualize the behavior indicated in the binomial test.

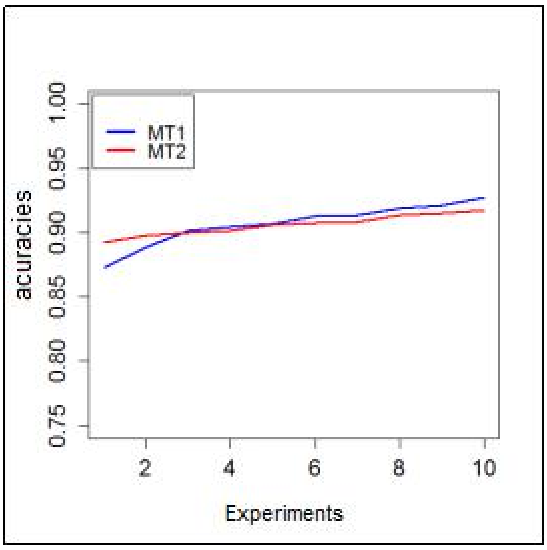


Figure 11. MES. – Accuracy of MT1 and MT2 methods.

For the database, sensitivity describes the probability that a legitimate firearm shot will be classified as a firearm shot.

The T-test was performed for each sample and it was found that the difference between MT1 and MT2 was more pronounced, showing the best means of sensitivity between the methods, 89.46% and 87.83, respectively. The other methods do not have as interesting an outcome as those achieved by multi-objective optimization (figure 12). Table 3 provides the data of the illustration.

Table 3. MES. – T-tests for each method + maximum sensitivity recorded.

	Higher sensitivity (%)	Mean (%)	Confidence interval (%)	Significance (%)	p-value
PCA	76,63	72,55	75,58 – 76	5	2.2e-16
MO	89,79	83,57	81,15 – 85,99	5	4.684e-14
MT1	92	89,46	87,83 – 91,1	5	7.575e-16
MT2	91,37	87,83	86,42 – 89,24	5	2.316e-16

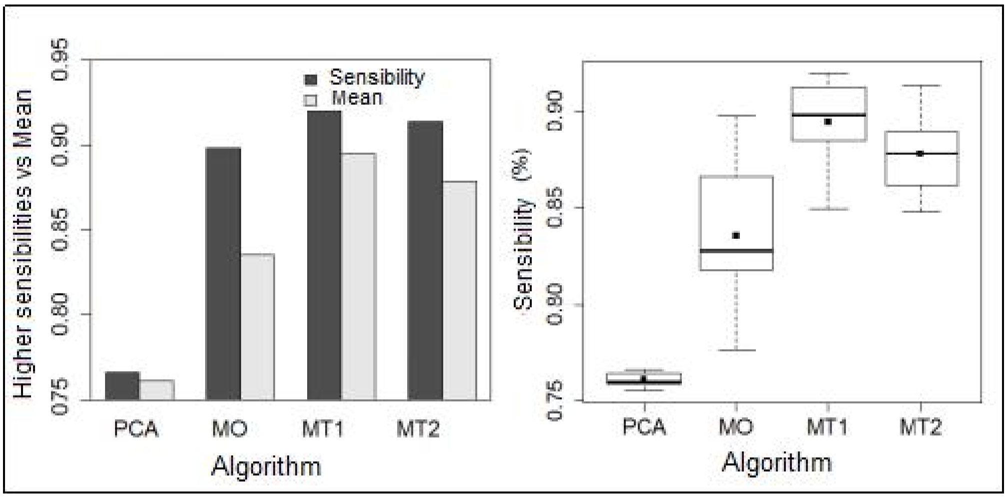


Figure 12. MES. – Sensitivity tests obtained.

Performing the binomial test for the times MT1 exceeds MT2, the following result is obtained:

- Threshold of 50%
- Significance α = 0,05 (5%)
- p-value = 0,0009766

The test is conclusive as to the hypothesis of MT1 to overcome MT2, the same test presents a probability of success of 100%. In figure 13 you can see their behavior in reducing false negatives. The lowest values of MT1 and MT2 are 4% and 4.3% respectively and represent the percentage of false negatives in the confusion matrix of the best instance of each sample.

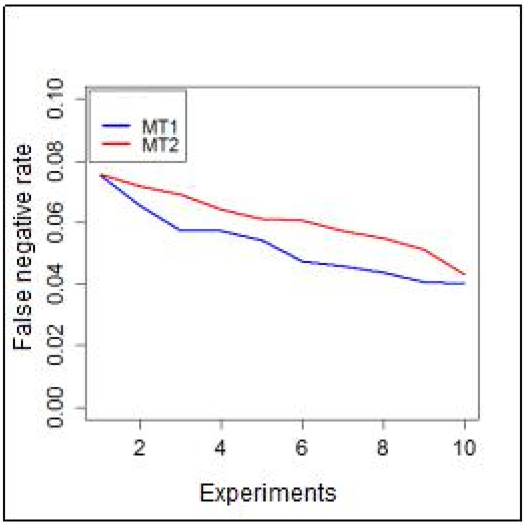


Figure 13. MES. – False Negative Rate of methods MT1 and MT2. The smaller the better.

5.2.2 Analysis

We return to our research question: "Is it possible to improve the performance of an audio featureset used in AAC tasks through the multi-objective optimization of this set?".

We have the following hypotheses that relate to the general objective explained in the introduction of this work (analyze the power of evolutionary multi-objective algorithms in the design of audio analytical features for problems of automatic audio classification) and that it concerns the measures of performance (accuracy and sensitivity for the said problem) of the research question:

Hypothesis 1: Solution accuracy.

- H0: The use of the multi-objective optimization model does not improve the accuracy of the classification.
- H1: The use of the multi-objective optimization model improves the accuracy of the classification.

Hypothesis 2: Solution sensitivity.

- H0: The use of the multi-objective optimization model does not improve the classification sensitivity.
- H1: The use of the multi-objective optimization model improves the classification sensitivity.

We used the MT1 method that obtained the best performance among the multi-objective methods and compared it with MO, since this was clearly the best method not to use the multi-objective technique. table 4 presents the results of each hypothesis test..

Table 4. Results of binomial tests (MT1 and MO) for each hypothesis.

	Threshold	Significance	Confidence interval	p-value
Hipótese 1	50 %	5 %	60,58 % - 100 %	0,01074
Hipótese 2	50 %	5 %	60,58 % - 100 %	0,01074

Since p-value is less than the level of significance of both problems, null hypotheses are discarded. This means that statically the multi-objective optimization method with false negative reduction (MT1) is superior to mono-target optimization in more than half of the cases. According to the same test, this effectiveness is statistically superior to 60.58% of the cases for both hypotheses and is probably 90%.

5.3 Consolidation of Results

When analyzing the methods in terms of accuracy and sensitivity, it was obtained important indications that the multi-objective solution has a significant weight in the effectiveness of the solution of AAC problems. MT1 (a multi-objective genetic algorithm in which the second objective is to reduce false negatives combined with 1-NN) was the method that obtained the best performance in each criterion of the experiment performed.

The problem addressed in the experiment is real, which shows the applicability of the method. The particularly interesting indication of how promising it can be is the approximation with the original commercial solution, discussed in [25]. In a problem of two classes: 1 - firing of firearm; 2 - another noise, the solution already marketed obtained a reduction of error to 3.57% [25] already the proposal in this work even reduced the error to 7.32% (equivalent to the accuracy of 92.68% of the method MT1 in table 2), but, there is a difference in the database as to the quantity and type of sounds, since the original problem is distinguishing firearm shots from any other sound, while the problem addressed here is to distinguish between gun shots of fire and fireworks, which are more similar classes and therefore the distinction is more difficult. It is possible that the proposed solution offers more interesting results on the same basis as the original problem. An opportunity for future study.

In addition to the statistical conclusions, some aspects can be highlighted regarding each method. Feature learning with PCA and the simple genetic algorithm are the fastest, reaching the results in minutes, whereas the multi-objective approach usually gives results in hours. However, taking into account that it is an automated process the result is more interesting than manual solutions like ad-hoc. In the end, eliminating the selection step at the end of the EDS implementation (Section 3.1.4), while simplifying the process does not leave the solution more agile.

The solution proposed in this work was the one implemented by the two MT1 and MT2 methods and is statistically superior to the others. Of course, this is only said about the ability to optimize audio features. There was no concern in optimizing the classifier algorithm, which was fixed.

Referring to the expected solution of session 2 it can be seen that the proposed solution corresponds to the expectation of the research, so to speak:

- Correctness: The solution influenced the improvement of the classification.
- Adequacy: The solution took into account the particularities of the problems, seeking the satisfaction of constraints.
- Reusability and scalability: The technique can be employed in several types of problem.
- Economy of knowledge: It is not necessary to know the nature of the audio features, it is enough only the handling of the technologies of classification.

6 Conclusions

The audio classification area is huge and diverse, where feature learning is a very important aspect. Despite the significant advances achieved with Deep Learning, this work showed that there is still room for improvement in other ways.

The main purpose of the solution is to improve the performance of the audio featureset in the classification tasks. Inspired by the Extractor Discovery System, it was dared to extrapolate the space of generic features, which are usually those employed in the extraction and selection phases, seeking to build more dynamic solutions that are not easily thought of by a specialist.

These solutions are intrinsically linked to the classification algorithm used, since no search or attempt was made to improve the classifier by adjusting its parameters. Even so, the classification produced interesting results. It is concluded that the analyzed feature optimize the classifier for that used data base. Although one type of classifier is not suitable for a given Automatic Audio Classification problem, it is possible to improve its performance through the development of analytical features. It does not mean that another type of classifier configured with other parameters can not overcome it, but that within its specifications it will be optimized. Therefore, ExpertMIR may be able to optimize all sorts of audio classifiers.

The need for tools that assist the community in the development of audio features can be met by implementations of the approach proposed by this work.

Taking into account the difficulty of process and cost in the design of analytical features and, aiming at better adapting the nature of some problems of AAC, we tried to develop an alternative that suited the needs of the area. With the consolidated results of the experiment it was possible to state the usefulness of the proposal. ExpertMIR is an open and useful solution for the optimization of acoustic features, especially when you can not count on specialists, you do not have the resources and time, or even when a solution is difficult to be designed analytically.

It is important to expand the comparison with other approaches and to do tests with other bases in order to identify any limitations of the model or to evolve it. Thus, our materials and methods are available so that others people can carry out further studies and complementary research.

Acknowledgements

This research was supported by FACEPE (Foundation of Support to Science and Technology of Pernambuco) with financial.

References

[1] BENGIO, Yoshua; COURVILLE, Aaron; VINCENT, Pascal. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, v. 35, n. 8, p. 1798-1828, 2013.

[2] MCKAY, Cory et al. ACE: A Framework for Optimizing Music Classification. In: **ISMIR**. 2005. p. 42-49.

[3] YASLAN, Yusuf; CATALTEPE, Zehra. Audio music genre classification using different classifiers and feature selection methods. In: **18th International Conference on Pattern Recognition (ICPR'06)**. IEEE, 2006. p. 573-576.

[4] BURKA, Zak. Perceptual audio classification using principal component analysis. 2010.

[5] ERONEN, Antti. Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs. In: **Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on**. IEEE, 2003. p. 133-136.

[6] HUMPHREY, Eric J.; BELLO, Juan Pablo; LECUN, Yann. Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics. In: **ISMIR**. 2012. p. 403-408.

[7] PACHET, François; ZILS, Aymeric. Evolving automatically high-level music descriptors from acoustic signals. In: **International Symposium on Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2003. p. 42-53.

[8] PACHET, François; ROY, Pierre. Exploring billions of audio features. In: **2007 International Workshop on Content-Based Multimedia Indexing**. IEEE, 2007. p. 227-235.

[9] PACHET, François; ROY, Pierre. Analytical features: a knowledge-based approach to audio feature generation. **EURASIP Journal on Audio, Speech, and Music Processing**, v. 2009, n. 1, p. 1, 2009.

[10] ZHOU, Xinquan; LERCH, Alexander. Chord Detection Using Deep Learning. In: **Proceedings of the 16th ISMIR Conference**. 2015.

[11] POTAMITIS, Ilyas; GANCHEV, Todor. Generalized recognition of sound events: Approaches and applications. In: **Multimedia Services in Intelligent Environments**. Springer Berlin Heidelberg, 2008. p. 41-79.

[12] SOHN, Jongseo; KIM, Nam Soo; SUNG, Wonyong. A statistical model-based voice activity detection. **IEEE signal processing letters**, v. 6, n. 1, p. 1-3, 1999.

[13] VARILE, Giovanni Battista; ZAMPOLLI, Antonio. **Survey of the state of the art in human language technology**. Cambridge University Press, 1997.

[14] MUTHUSAMY, Yeshwant K.; BARNARD, Etienne; COLE, Ronald A. Reviewing automatic language identification. **IEEE Signal Processing Magazine**, v. 11, n. 4, p. 33-41, 1994.

[15] LIPPMANN, R. P. Review of neural networks for speech recognition. **Neural Computation**, MIT Press, v. 1, n. 1, p. 1–38, 2016/07/27 1989. Disponível em: http://dx.doi.org/10.1162/neco.1989.1.1.1.

[16] KWON, Oh-Wook et al. Emotion recognition by speech signals. In:**INTERSPEECH**. 2003.

[17] GUO, Y. B.; AMMULA, S. C. Real-time acoustic emission monitoring for surface damage in hard machining. **International Journal of Machine Tools and Manufacture**, v. 45, n. 14, p. 1622-1627, 2005.

[18] POTAMITIS, Ilyas; GANCHEV, Todor; FAKOTAKIS, Nikos. Automatic acoustic identification of insects inspired by the speaker recognition paradigm. In: **INTERSPEECH**. 2006.

[19] LEE, C.-H.; HAN, C.-C.; CHUANG, C.-C. Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients. **Audio, Speech, and Language Processing, IEEE Transactions on**, v. 16, n. 8, p. 1541–1550, 2008. ISSN 1558-7916.

[20] SAHA, B.; PURKAIT, P.; MUKHERJEE, J.; MAJUMDAR, A.; MAJUMDAR, B.; SINGH, A. An embedded system for automatic classification of neonatal cry. In: **Point-of-Care Healthcare Technologies (PHT), 2013 IEEE**. [S.l.: s.n.], 2013. p. 248–251.

[21] GOLUB, S. Classifying recorded music**. MSc in Artificial Intelligence. Division of Informatics. University of Edinburgh**, 2000.

[22] EGGINK, Jana; BROWN, Guy J. A missing feature approach to instrument identification in polyphonic music. In: **Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on**. IEEE, 2003. p. V-553-6 vol. 5.

[23] KLAPURI, Anssi; DAVY, Manuel (Ed.). **Signal processing methods for music transcription**. Springer Science & Business Media, 2007.

[24] PEETERS, Geoffroy. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. Rep., IRCAM, 2004.

[25] ARAÚJO, D. F. D. E. Busca como sistema de apoio à melhoria de classificadores automáticos de áudio. 2014.

[26] HUTTER, Frank; HOOS, Holger H.; LEYTON-BROWN, Kevin. Sequential model-based optimization for general algorithm configuration. In:**International Conference on Learning and Intelligent Optimization**. Springer Berlin Heidelberg, 2011. p. 507-523.

[27] WEST, Kristopher; COX, Stephen. Features and classifiers for the automatic classification of musical audio signals. In: **ISMIR**. 2004.

[28] AUCOUTURIER, Jean-Julien; DEFREVILLE, Boris; PACHET, François. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. **The Journal of the Acoustical Society of America**, v. 122, n. 2, p. 881-891, 2007.

[29] RITTHOF, O. et al. A hybrid approach to feature selection and generation using an evolutionary algorithm. In: **2002 UK workshop on computational intelligence**. 2002. p. 147-154.

[30] MIERSWA, Ingo; MORIK, Katharina. Automatic feature extraction for classifying audio data. **Machine learning**, v. 58, n. 2-3, p. 127-149, 2005.

[31] CABRAL, Giordano; PACHET, François; BRIOT, Jean-Pierre. Recognizing chords with EDS: Part one. In: **International Symposium on Computer Music Modeling and Retrieval**. Springer Berlin Heidelberg, 2005. p. 185-195.

[32] KINNEAR, Kenneth E. **Advances in genetic programming**. MIT press, 1994.

[33] FISHER, Ronald A. The use of multiple measurements in taxonomic problems. Annals of eugenics, v. 7, n. 2, p. 179-188, 1936.

[34] EIBEN, Agoston E.; SMITH, James E.Introduction to evolutionary computing. Heidelberg: springer, 2003.

[35] MCKAY, Cory. **Automatic music classification with jMIR**. 2010. Tese de Doutorado. McGill University.

[36] MCENNIS, Daniel; MCKAY, Cory; FUJINAGA, Ichiro; DEPALLE, Philippe. jAudio: A feature extraction library. In: **Proceedings of the International Conference on Music Information Retrieval**. 2005. p. 600-3.

[37] HOLMES, Geoffrey; DONKIN, Andrew; WITTEN, Ian H. Weka: A machine learning workbench. In: **Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on**. IEEE, 1994. p. 357-361.

[38] DURILLO, Juan J.; NEBRO, Antonio J. jMetal: A Java framework for multi-objective optimization. **Advances in Engineering Software**, v. 42, n. 10, p. 760-771, 2011.

[39] KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Ijcai**. 1995. p. 1137-1145.

[40] R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[41] VALENZISE, Giuseppe et al. Scream and gunshot detection and localization for audio-surveillance systems. In: **Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on**. IEEE, 2007. p. 21-26.

[42] GEROSA, Luigi et al. Scream and gunshot detection in noisy environments. In: **Signal Processing Conference, 2007 15th European**. IEEE, 2007. p. 1216-1220.

[43] GAMERMAN, Dani; DOS SANTOS MIGON, Helio.**Inferência estatística: uma abordagem integrada**. Instituto de Matemática, Universidade Federal do Rio de Janeiro, 1993.

Inteligencia Artificial 22(64), 36-46
doi: 10.4114/intartif.vol22iss64pp36-46

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

On the Enhancement of Classification Algorithms Using Biased Samples

Safaa O. Al-mamory
College of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
salmamory@uoitc.edu.iq

Abstract Classification algorithms' performance could be enhanced by selecting many representative points to be included in the training sample. In this paper, a new border and rare biased sampling (BRBS) scheme is proposed by assigning each point in the dataset an importance factor. The importance factor of border points and rare points (i.e. points belong to rare classes) is higher than other points. Then the points are selected to be in the training sample depending on these factors. Including these points in the training sample enhances classifiers experience. The results of experiments on 10 UCI machine learning repository datasets prove that the BRBS algorithm outperforms many sampling algorithms and enhanced the performance of several classification algorithms by about 8%. BRBS is proposed to be easy to configure, covering all points space, and generate a unique samples every time it is executed.

Keywords: Classification, LOF, Decision Boundary, Biased Sampling, imbalanced dataset.

1 Introduction

The diverse applications of classification algorithms encouraged researchers to enhance the performance of these algorithms; these applications include customer target marketing [1], medical disease diagnosis [2], supervised event detection [3], multimedia data analysis [4], biological data analysis [5], document categorization and filtering [6], and social network analysis [7]. However, enhancing classifiers performance is a challenging mission. There are several ways to improve classifier's accuracy such as preprocess dataset [8], enhancing algorithms performance, and post-process the classifiers' results [9]. Data sampling is one of data preprocessing techniques. This paper boosts the classification algorithms by enhancing selection of training sample as a preprocess step.

Uniform distributions and dataset with one level of density are rare in real applications. Some records may be of more value in the sample than others; knowing points' importance could help in sampling by assigning an importance value for each point. Having points' importance helps in obtaining representative samples which in turn enhances classifier's performance. Most of the existing sampling algorithms neglect representing small clusters in the sample. In this paper, Border and Rare Biased Sampling Algorithm (BRBS) is proposed in which the border points and rare class points are more important than others. We will use point and instance terms interchangeably.

In more details, BRBS algorithm deals with the dataset from the point of view of the classifier. Classifying border points (between different classes) and rare points is the most challenging task for the classifier. In addition, the classifiers vary in shaping the decision boundary; the classification algorithm strategy in separating the various dichotomies is different from classifier to another. BRBS depends on local outlier factor (LOF) algorithm to specify points' importance score; then it uses these scores in selecting the border and rare points to be included in the training set. Figure 1 presents synthesized dataset containing 400 points. It is clear to note how BRBS (first figure from the left) ensures selecting border points (i.e. red points). In contrast, sampling with replacement and

ISSN: 1137-3601 (print), 1988-3064 (on-line)

sampling without replacement algorithms are not biased as can be seen in the figure. The main contribution of this paper is to suggest a sampling mechanism, depending on LOF, to ensure rare classes and border points' coverage. The experiments on 10 different datasets with three classifiers prove that BRBS has outperformed holdout sampling and cross validation and enhanced different classifiers performance by about 8% (on average).

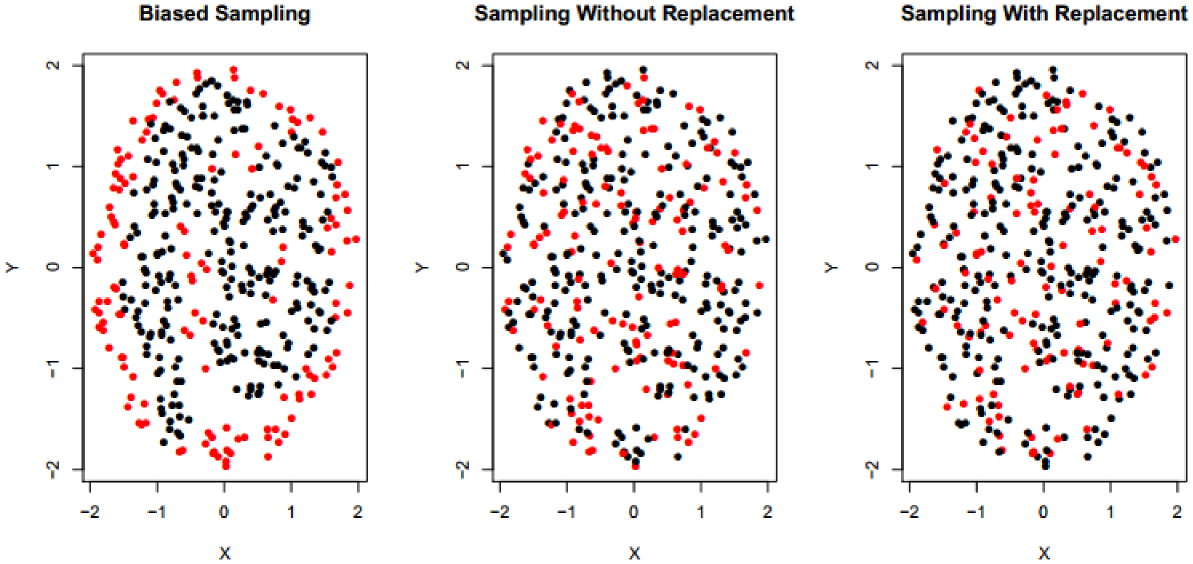


Figure. 1: visual comparison between BRBS, sampling without replacement, and sampling with replacement on synthesized dataset

This paper is organized as follows. The related work is presented Section 2. Section 3 states the proposed algorithm. The experimental results are presented in Section 4. Finally, Section 5 concludes this paper.

2 Related Work

In the last few decades, the research community has studied precisely the sampling in order to improve classification and clustering performance. The first group of researchers worked on biased sampling to force the sample to have points from dense regions of data more than the points from sparse regions. In other words, they generate samples that allow clustering processes to find clusters more accurately. George Kollios et al. [10] proposed a density-biased sampling by calculating the density of the local space around each point and then place the point in the sample with probability that is a function of local density. They chose kernel function for density estimation and applied their method to cluster and outlier detection problems. Ana Paula Appel et al. [11] presented a biased box sampling algorithm using local density. The technique is based on a multi-dimensional and multi-resolution grid structure where its depth depends on points' local density of the related region. Christopher R. Palmer [12] used a weighted sample in order to preserve the data density. They introduce a sampling technique to improve on uniform sampling when skewed clusters are processed. A hashing function is used when doing a biased sampling to map bins in space to a linear ordering. An incremental algorithm is introduced by Frédéric Ros et al. [13] to combine distance and density concepts. They manage distance concepts in order to make sure space coverage and fit cluster shapes by selecting representative points in every cluster.

The second group of researchers did their best to enhance classification accuracy when the dataset had rare classes which are hard to classify. An over-sampling approach is proposed by Nitesh V. Chawla et al. [14] in which the minority classes are over-sampled by generating synthetic points instead of over-sampling with replacement. Piyasak Jeatrakul et al. [15] did a combination of over-sampling using synthetic minority over-sampling technique (SMOTE) and under-sampling using complementary neural network. Gencheng Liu et al. [16] creates a weighted fuzzy rules from the training data, then it produces new minority points under the fuzzy rules guidance. The rule weight of any fuzzy rule determines the number of minority points to be generated. Georgios Douzas et al. [17] suggested to focus data generation on important areas of the input space by clustering the data using k-means. Henceforth, a cluster is a safe region when there is a high ratio of minority points. To avoid noise creation, oversampling safe clusters only to enable k-means SMOTE. Hui Han et al. [18] noticed that the borderline points of the minority class are easier to be misclassified than those points away from the borderline. Thus, they oversampled points of the minority class at the borderline.

The third group of researchers focused on random sampling to enhance classifiers' performance. The holdout algorithm samples the data randomly into two samples which are 66% for training and 34% for testing (these percent are approximated) [19]. Random subsampling approach apply hold-out method several times to boost the estimation of a classifier performance [19]. Cross-validation (CV) is an alternative to random subsampling, in which each record is used the same number of time for training and once only for testing [19]. The idea of the bootstrap is to sample the dataset with replacement to form a training set [20].

The proposed method is different from all previous work in two folds. The first fold is that it is uses local outlier factor as a guide for the sampling. Unlike the previous work in which the work was biased to select from denser regions more than sparse regions, the second fold is that it is biased towards selecting border points and rare class points to be in the sample. To the best of our knowledge, this is the first paper uses importance scoring of points in the sampling.

3 The Proposed Algorithm

The performance of any classifier could be improved if the training examples are representative. The sample containing instances from rare classes and border points is more representative than simple random sampling. Subsection 3.1 describes the concept of local outlier factor while Sunsection 3.2 states how to use that concept of local outlier factor in enhancing sampling.

3.1 Local Outlier Factor

An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism [21]. Three different techniques to detect outliers are available which are supervised, semi-supervised, and unsupervised [22]. LOF [23] is a density based and unsupervised algorithm which gives a numeric value for each point representing the outlier factor. The normal value is 1; the higher the value is the outlier the point is. In this paper, LOF algorithm is used as a base algorithm to rank points. LOF has one control parameter, MinPts. The remaining of this subsection will present the main concepts of LOF algorithm.

To detect density-based outliers, it is necessary to compare the densities of different sets of objects, which means that we have to determine the density of sets of objects dynamically. Therefore, we keep MinPts as the only parameter and use the values reach-distMinPts(p, o), for o ∈ N MinPts(p), as a measure of the volume to determine the density in the neighborhood of an object p. Intuitively, the local reachability density of an object p is the inverse of the average reachability distance based on the MinPts nearest neighbors of p. Equ. 1 presents the local reachability distance [23].

$$lrd_{MinPts}(p) = 1/\left(\frac{\sum_{o\in N_{MinPts}(p)} reach\text{-}dist_{MinPts}(p,o)}{|N_{MinPts}(p)|}\right) \tag{1}$$

The outlier factor of object p captures the degree to which we call p an outlier. It is the average of the ratio of the local reachability density of p and those of p’s MinPts-nearest neighbors. It is easy to see that the lower p's local reachability density is, and the higher the local reachability densities of p's MinPts-nearest neighbors are, the higher is the LOF value of p. The (local) outlier factor of p is defined as in Equ. 2 [23].

$$LOF_{MinPts}(p) = \frac{\sum_{o\in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \tag{2}$$

where lrd() is the local reachability density of a given point with respect to MinPts, and $N_{MinPts}(p)$ is the list of nearest MinPts to the point p given in Equ. Z. In this paper, we will use LOF(*p*) or (LOF) instead $LOF_{MinPts}(p)$ for abbreviation.

In LOF algorithm, the local density of any point is compared to the neighbors' local densities. A point is considered an outlier if its density is lower than its neighbors density. The score of LOF is approximately 1 means that the density around the point is similar to its neighbors while LOF value much larger than 1 is indicator of an outlier.

3.2 The Biased Sampling Algorithm

The border between two neighboring regions is known as the decision boundary [19]. The borderline points and rare points are simply misclassified compared with those ones far from the borderline. Therefore, BRBS is suggested to ensure that the training sample contain these classifiers' challenging points. To achieve this goal, an importance score is given to each point in dataset. LOF is used, in this paper, to give this score for all points. The borderline points and rare points always have high LOF score since they have less set of neighbors' points. Figure 2 depicts the LOF values for instances of Yeast dataset where the red line represents the highest 10% LOF values. Most of these 10% points are from rare and border points from different classes.

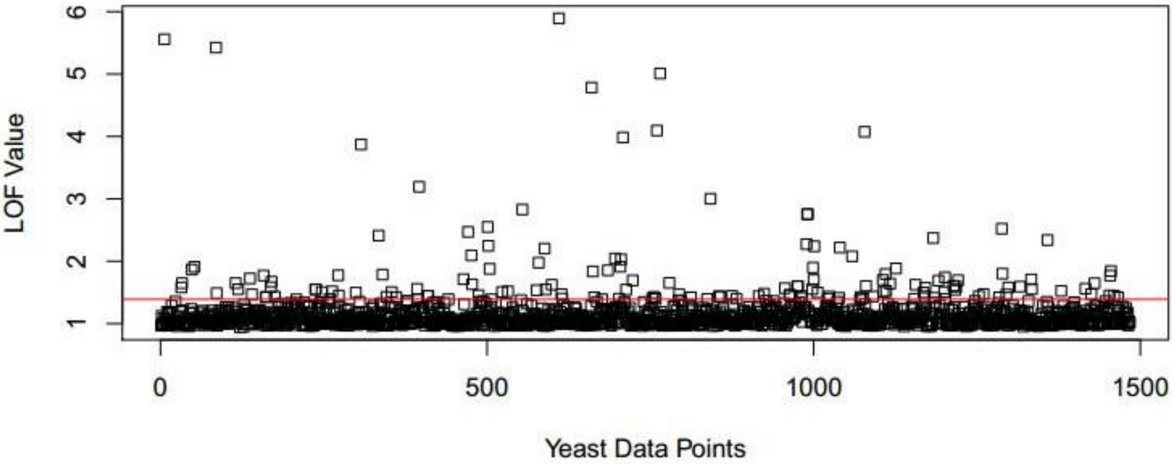


Figure 2: LOF values for Yeast instances; the red line representing 10% of the dataset

Having the highest 10% of points (having highest LOF scores) in the training sample will improve classifier performance. In this paper, the training sample will have 66% of points and the testing sample will have 34% of points. Algorithm 1 contains the pseudo code of BRBS algorithm. In general, the BRBS algorithm makes a ranking of the points according to the importance score which is computed by LOF algorithm (Lines 2 and 3). Hereafter, a holdout technique with scoring points modification is used to partition the dataset into two parts (Lines 4 and 5). Line 4 may classify noise points as border points. Therefore, it is recommended to employing one of noise removing algorithms before BRBS is applied.

Algorithm 1: pseudo code of BRBS algorithm

```
    Algorithm BRBS (D , MinPts)
    Input: D , MinPts
    Output: Train, Test
1.  Begin
2.  For each d ∈ D do
3.      $L(d)$ = Compute $LOF_{MinPts}(d)$ using Equ. 2 ;
4.  Train = select 66% of points having highest L(D) score;
5.  Test = select 34% of points having minimum L(D) score;
6.  Return Train, Test;
7.  End
```

BRBS uses LOF idea but apply LOF in an innovative way and have several advantages over existing sampling algorithm. It could be look for BRBS from different points of views like:

- *Ease of use*: several sampling algorithms have many parameters to set which are more meaningful to the data miner than to the user and hence not easy to set. BRBS classifies a point as a border point (or a rare point) depending on one control variables (i.e. MinPts). The higher MinPts value is the more border points. The best value for the parameter is selected using K-dist plot [24] curve which sorts the points according to distances. After drawing the curve, the knee in the curve reflects the best value to be selected making the algorithm easy to use.
- *Not random sample*. The proposed sampling algorithm generates same sample every time it is executed since it selects top n% instances as a training sample having highest LOF score. This feature is not available in random sampling (with or without replacement). In addition, it is considered as unsupervised sampling algorithm; however it is biased towards selecting rare class instances.
- *Space coverage*. In BRBS, choosing the points with highest LOF scores ensures rare classes are represented. In addition, the points at different classes' borders are selected since their density is different from their neighbors' densities. These points represent about 10% of the dataset. Hence, it should select extra more than 50% of points to be included in the training sample.

4 Experimental Results

The main goal of this section is to do experiments to test the performance of the proposed algorithm. To achieve this goal, three sets of experiments were conducted. The first set focuses on selecting instances with the algorithm. The second set of experiments was to compare the algorithm with two most common algorithms which are hold-out sampling algorithm and cross validation algorithm. The third set of experiments conducted to evaluate the root mean square error (RMSE) of the algorithm. All experiments are conducted by R language [25] and Weka [26].

The experiment design that is used to test the performance of BRBS is shown in Figure 3 to obtain a valid comparison. A representative test set (*DS*) including both easy and hard points is kept apart to make the test fair. The size of *DS* is 20% of the given dataset. Then, the rest of the points (*DT*) are used as training set to learn two sets of classifiers, the first in the standard way to be used as a baseline and the second set of classifiers are learnt with the BRBS. Then, the two sets of classifiers are evaluated against the test set (DS). This setup is used with hold-out, cross-validation, and BRBS. Three different classification algorithms were selected (as base classifiers) to compare with, which are J48, Naïve Bayes, and Multi-layer perceptron. The reason for selecting these algorithms is that they used different strategies for dealing with decision boundary regions.

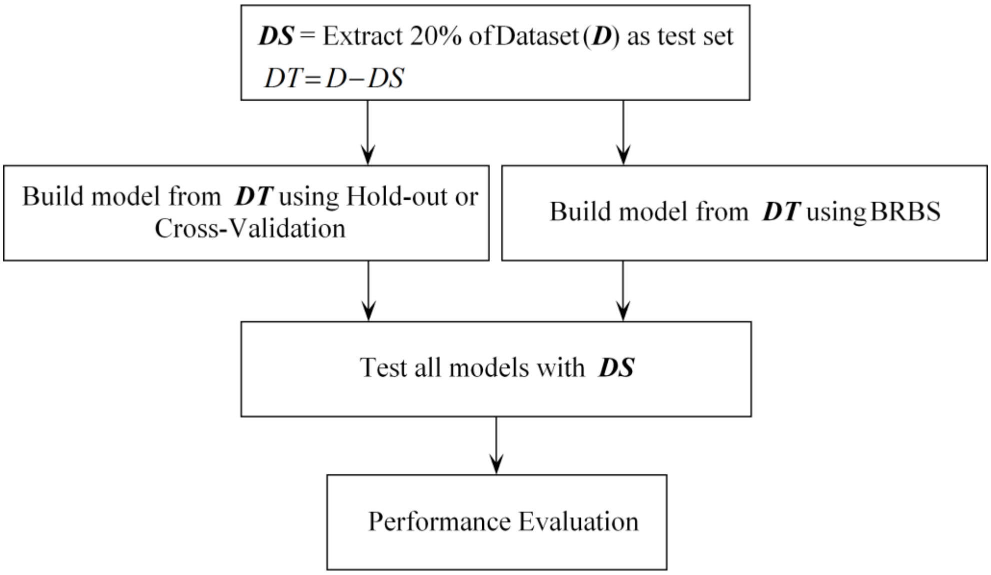

DS = Extract 20% of Dataset (D) as test set
DT = D − DS
Build model from DT using Hold-out or Cross-Validation
Build model from DT using BRBS
Test all models with DS
Performance Evaluation

Figure 3. Experiment design used in this paper

In order to compare BRBS and different algorithms, several measures had been used which are accuracy, recall, precision, the F measure. The accuracy is computed using (*TP+TN*)/(*TP+TN+FP+FN*) [19] where TP and TN are correctly classified points (positive and negative points, respectively) while FP and FN are incorrectly classified points (positive and negative points, respectively). Recall (*TP*)/(*TP +FN*) measures the proportion of actual positives that are correctly classified. Precision is the ratio of recognized positives which are correctly classified; it is computed by(*TP*)/(*TP+FP*). The F-measure is a mixture of recall and precision and is calculated as 2 * (*Precision * Recall*/(*Precision + Recall*)) [27]. In the presence of imbalanced datasets, it is more appropriate to use F-measure.

Ten different datasets have been used from UCI repository [28] as can be seen in Table 1. These datasets have different number of features and ranging from binary to multi class classification problems. In addition, these datasets are from different fields. Some of the selected datasets have rare classes like Ecoli, Yeast, and Glass; we consider classes occurred less than 10% as rare.

Table 1: The used dataset in the experiments from UCI repository

Datasets	#Instances	#Features	#Classes
Diabetes	768	8	2
ionosphere	351	34	2
Parkinsons	197	23	2
Phishing	11055	31	2
Iris	150	4	3
Lung Cancer	32	56	3
User Knowledge Modeling (UKM)	258	5	4
Glass	214	9	6
Ecoli	336	7	8
Yeast	1484	8	10

The proposed sampling algorithm produces representative training samples for the classifiers. Each class, containing set of instances, has a set of border instances which are selected by the proposed algorithm. Furthermore, the selected samples are the more difficult instances to classify. According to 10 experiments (on datasets appeared in Table 1) to get samples from, the proposed algorithm successfully produces good samples. The percent of each class in these datasets before and after sampling is presented in Table 2. As can be seen in the table, the algorithm fairly took instances from all classes when selecting the sample. Moreover, it selects most instances in rare classes since these instances sometimes appeared to be noise. The rare classes' instances almost have biased features values so their LOF value tends to be high. Therefore the proposed algorithm selects them as can be noted from Table 2 where the instances of classes 3 and 4 from Ecoli dataset are selected entirely; more examples are bolded. It could be concluded that the proposed algorithm behavior in selecting training instances is biased toward selecting the instances from rare classes.

Comparison experiments using different classification algorithms and different sampling algorithms had been conducted to measure BRBS performance; Table 3 shows these results. The results for hold-out and cross validation (with 10 folds) are obtained from R with default set of parameters. The best value for each dataset and each classification algorithm is bolded except those values that are equal. As can be noted from the table that BRBS algorithm is better than most algorithms because BRBS increases the classifier's knowledge by presenting the best sample to be taught from. It should be noted that almost all classifiers were did worse with Lung Cancer dataset since the size of dataset is too small when are partitioned while the performance of BRBS was steady since it characterizes the sample at instance level. The averaged accuracy for different datasets is depicted in Figure 4 where each column represents the average of three classifiers' experiments. It can be concluded from the figure that applying BRBS is always outperforms the other algorithms. Moreover, BRBS works well on both balanced and imbalanced datasets (i.e. UKM, Glass, Ecoli, and Yeast). On average, BRBS accuracy outperforms hold-out by 6.7% and cross-validation by 8.1%.

Table 2: Percent of classes before and after sampling in the format before(after)

Datasets	Class Number									
	1	2	3	4	5	6	7	8	9	10
Diabetes	34.9 (24.5)	65.1 (42.2)								
ionosphere	35.9 (32.5)	64.1 (34.2)								
Parkinsons	24.6 (20)	75.4 (46.7)								
Phishing	55.7 (33.3)	44.3 (33.4)								
Iris	33.3 (24.7)	33.3 (22.7)	33.3 (19.3)							
Lung Cancer	28.1 (18.8)	40.6 (21.9)	31.3 (25)							
UKM	9.3 (4.3)	32.2 (19.8)	34.1 (24.8)	24.4 (17.8)						
Glass	32.7 (19.2)	7.9 (4.7)	4.2 (3.3)	35.5 (26.6)	13.6 (8.4)	6.1 (4.2)				
Ecoli	42.6 (29.2)	22.9 (17)	**0.6 (0.6)**	**0.6 (0.6)**	10.4 (5.7)	6 (2.1)	**1.5 (1.5)**	15.5 (10.1)		
Yeast	16.4 (10.2)	28.9 (20.1)	31.2 (23.5)	3 (1.5)	2.4 (0.8)	3.4 (2)	11 (7.6)	2 (0.7)	1.3 (0)	**0.3 (0.3)**

Table 3: Comparison in Accuracy measure between BRBS algorithm, hold-out, and CV algorithms with three different classification algorithms

Datasets	J48			Naïve Bayes			Multi-layer Perceptron		
	Hold Out	CV	BRBS	Hold Out	CV	BRBS	Hold Out	CV	BRBS
Diabetes	69.3	68	**75.2**	**75.8**	73.9	74.5	70.6	72.5	**73.8**
Ecoli	80.6	77.6	**85.1**	83.6	83.6	83.6	82.1	80.6	**86.6**
Glass	64.3	54.8	**71.4**	54.8	**59.5**	54.8	66.7	64.3	**76.2**
Ionosphere	91.4	87.1	**97.1**	**81.4**	80	80	88.6	82.9	**92.9**
Iris	93.3	83.3	93.3	93.3	86.7	93.3	96.7	86.7	**100**
Lung Cancer	16.7	83.3	**66.7**	16.7	33.3	**66.7**	50	50	**83.3**
Parkinsons	82.1	74.4	**89.7**	69.2	66.7	**71.8**	89.7	66.7	**92.3**
UKM	**98**	86.3	96.1	92.2	88.2	**94.1**	96.1	88.2	96.1
Yeast	56.1	52.4	56.1	56.4	56.1	**58.2**	57.4	51	**63.5**
Phishing	94.6	93.1	**95.5**	89.7	89.6	**90.1**	94.7	90.7	**95.6**
Average	74.6	76	**82.6**	71.3	71.7	**76.7**	79.3	73.3	**86**

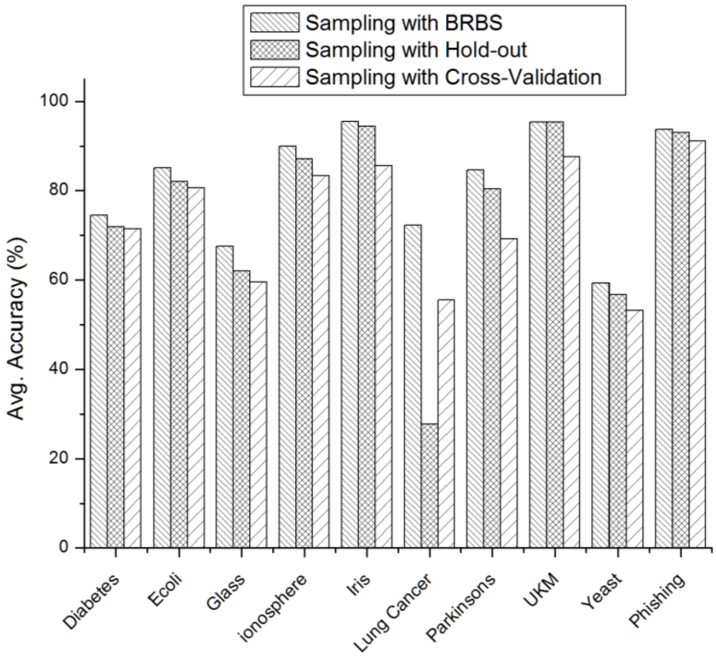


Figure 4: histogram represents a comparison of averaged accuracy measure for different sampling algorithms

For computing more performance comparison measures, three measures mentioned above are calculated which are the precision, recall, and F-measures for BRBS and Hold-out. These measures for four different datasets are shown in Table 4. In this table, we can observe that BRBS has the highest precision, recall, and F-measure values while comparing with Hold-out sampling.

Table 4: Comparison between BRBS and hold-out with precision, recall, and F measure

Datasets	Hold out			BRBS		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Diabetes	0.678	0.693	0.667	0.745	0.752	0.745
Ionosphere	0.918	0.914	0.915	0.974	0.971	0.972
Iris	0.933	0.933	0.933	0.933	0.933	0.933
Yeast	0.544	0.561	0.548	0.552	0.561	0.552
Average	0.768	0.775	0.766	0.801	0.804	0.801

The results of LOF algorithm are influenced by the parameter k indicating the nearest neighbours' number where estimating the density depends it. Determining the right size of *k* needs a little of experience. The accuracy of four different datasets (with J48 algorithm) was analyzed to show how the results depend on this parameter. These dataset are Diabetes, Ecoli, Glass, and Ionosphere. According to the results presented in Figure 5, where the value of *k* is in the interval $3 \leq k \leq 15$, it is easy to note how the accuracy value is affected by *k* parameter and every dataset has its best *k* value. Furthermore, some datasets has more than one value for *k* parameter by which we obtain highest accuracy.

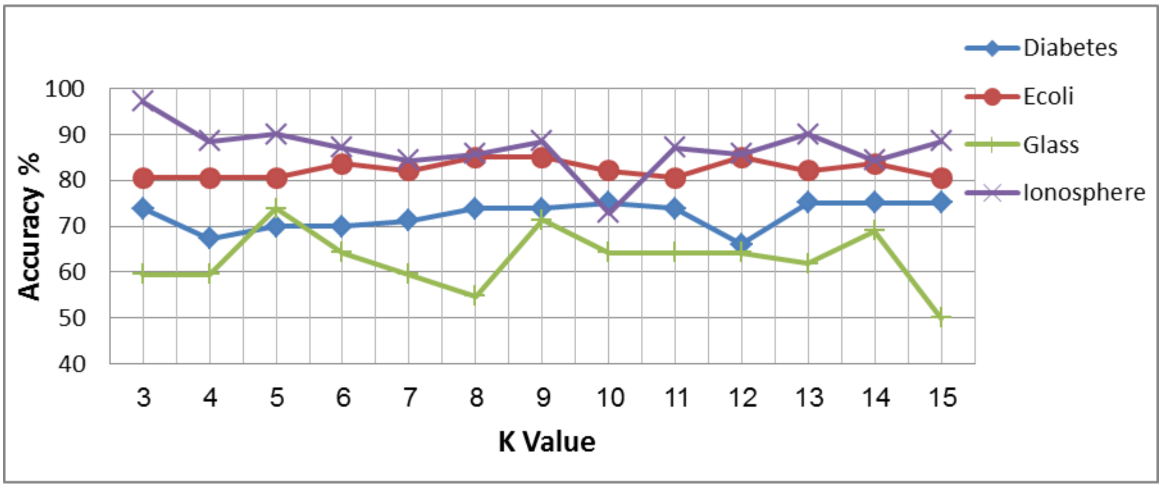


Figure 5: The influence of *k* parameter on accuracy value

Root mean square error (RMSE) [29] has been used here as another comparison measure to show the error rate in the BRBS algorithm and other algorithms. Equation of RMSE is stated in Equ. 3 where P is the predicted value and O is the observed one. It is well known that RMSE gives good indication as a comparison measure when the classification problem is binary. Hence, only four datasets were used in the experiment. Figure 6 depicts the results of RMSE as a comparison measure. Each column in the figure represents the average of three experiments with different sampling algorithms. The average of RMSE for all experiments was 0.33for BRBS, 0.36 for hold-out, and 0.41 for cross validation. It can be concluded from this figure that the RMSE results of BRBS algorithm is the minimum with respect to other algorithms.

$$RMSE = \sqrt[2]{\frac{1}{n}\sum_1^n (P_i - O_i)^2} \tag{3}$$

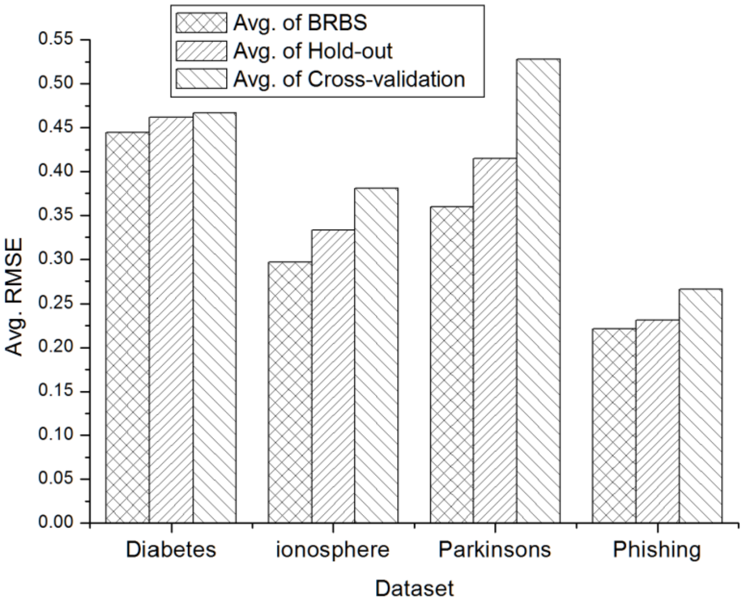


Figure 6: histogram represents a comparison of RMSE measure for different sampling algorithms

BRBS algorithm has good performance over several sampling algorithms. From the set of experiments, BRBS achieved highest accuracy with minimum RMSE values. Furthermore, the produced training sample covers the whole space paying attention to rare points and borderline points as noted in Table 2. BRBS is easy to use since it has one control parameter and every point in the dataset will get constant position in the samples because the points' importance score will be the same every time the algorithm is executed.

5 Conclusions

Several limitations exist in the state of the art of sampling algorithms like multi-control parameters, biasing towards density clusters only, and others. The margin points of different classes and minority class points have a great impact on classifier's model creation. Therefore, BRBS give these points more importance than other points in training sample building. It extracts margin points from decision boundary between classes and points from rare classes (using LOF as a scoring algorithm) to be included in the training sample to increase classifier experience. The main characteristics of BRBS are ease of use, not random samples, and space coverage.

Acknowledgements

The author recognizes and thanks the anonymous reviewer and the journal editor for their important comments and suggestions, which have significantly enhanced the paper.

References

[1] C. C. Aggarwal, *Data classification : algorithms and applications*. CRC Press, 2014.

[2] S. U. Ghumbre and A. A. Ghatol, "Heart Disease Diagnosis Using Machine Learning Algorithm," in *Proceedings of the International Conference on Information Systems Design and Intelligent Applications*, 2012, pp. 217–225.

[3] H. Becker, "Identification and Characterization of Events in Social Media," Columbia University, 2011.

[4] N. Patel and I. Sethi, "Multimedia Data Mining: An Overview," in *Multimedia Data Mining and Knowledge Discovery*, London: Springer London, 2007, pp. 14–41.

[5] A. Tanay, S. Roded, and S. Ron, "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 1, no. 1, pp. 24–45, 2004.

[6] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, 2005, pp. 78–87.

[7] J. Kim and M. Hastak, "Social network analysis: Characteristics of online social networks after a disaster," *Int. J. Inf. Manage.*, vol. 38, no. 1, pp. 86–96, 2018.

[8] S. O. Al-Mamory, "Classification performance enhancement using boundary based sampling algorithm," in *Annual Conference on New Trends in Information and Communications Technology Applications*, 2017, pp. 186–191.

[9] E. B. Kong and T. G. Dietterich, "Error-Correcting Output Coding Corrects Bias and Variance," *Mach. Learn. Proc.*, pp. 313–321, 1995.

[10] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold, "Efficient biased sampling for approximate clustering and outlier detection in large data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 5, pp. 1170–1187, Sep. 2003.

[11] A. P. Appel, A. A. Paterlini, E. P. M. de Sousa, A. J. M. Traina, and C. Traina, "A Density-Biased Sampling Technique to Improve Cluster Representativeness," pp. 366–373, 2007.

[12] C. R. Palmer and C. Faloutsos, "Density Biased Sampling: An Improved Method for Data Mining and Clustering," *ACM SIGMOD Int. Conf. Manag. Data*, no. 82, pp. 82–92, 2000.

[13] F. Ros and S. Guillaume, "DENDIS: A new density-based sampling for clustering algorithm," *Expert Syst. Appl.*, vol. 56, pp. 349–359, 2016.

[14] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE : Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[15] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Classification of imbalanced data by combining the complementary neural network and SMOTE algorithm," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6444 LNCS, no. PART 2, pp. 152–159, 2010.

[16] G. Liu, Y. Yang, and B. Li, “Fuzzy rule-based oversampling technique for imbalanced and incomplete data learning,” *Knowledge-Based Syst.*, vol. 158, pp. 154–174, 2018.

[17] G. Douzas, F. Bacao, and F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE,” *Inf. Sci. (Ny).*, vol. 465, pp. 1–20, 2018.

[18] H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning,” *LNCS*, vol. 3644, pp. 878 – 887, 2005.

[19] T. P, M. Steinbach, A. Karpatne, and K. Vipin, *Introduction to data mining*, 2nd ed. Pearson, 2018.

[20] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[21] D. M. Hawkins, *Identification of Outliers*. Dordrecht: Springer Netherlands, 1980.

[22] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

[23] M. M. . Breuniq, H.-P. . Kriegel, R. T. . Ng, and J. . Sander, “LOF: Identifying density-based local outliers,” *SIGMOD Rec. (ACM Spec. Interes. Gr. Manag. Data)*, vol. 29, no. 2, pp. 93–104, 2000.

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[25] S. Urbanek and M. Plummer, “R: The R Project for Statistical Computing.” [Online]. Available: https://www.r-project.org/. [Accessed: 15-Apr-2019].

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, p. 10, Nov. 2009.

[27] Y. F. Roumani, J. H. May, D. P. Strum, and L. G. Vargas, “Classifying highly imbalanced ICU data,” *Health Care Manag. Sci.*, vol. 16, no. 2, pp. 119–128, Jun. 2013.

[28] UC Irvine, “UCI Machine Learning Repository.” [Online]. Available: https://archive.ics.uci.edu/ml/index.php. [Accessed: 02-Mar-2019].

[29] D. G. Fox, “Judging Air Quality Model Performance,” *Bull. Am. Meteorol. Soc.*, vol. 62, no. 5, pp. 599–609, 1981.

Inteligencia Artificial, 22(64) (2019), 47-62
doi: 10.4114/intartif.vol22iss64pp47-62

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Dealing with Incompatibilities among Procedural Goals under Uncertainty

Mariela Morveli-Espinoza[1], Juan Carlos Nieves[2], Ayslan Possebom[3], and Cesar Augusto Tacla[1]

[1] Graduate Program in Electrical and Computer Engineering (CPGEI),
Federal University of Technology - Paraná (UTFPR), Curitiba - Brazil
morveli.espinoza@gmail.com, tacla@utfpr.edu.br

[2] Department of Computing Science
Umeå University, Umeå - Sweden
jcnieves@cs.umu.se

[3] Federal Institute of Parana, Paranavai - Brazil
possebom@gmail.com

Abstract

By considering rational agents, we focus on the problem of selecting goals out of a set of incompatible ones. We consider three forms of incompatibility introduced by Castelfranchi and Paglieri, namely the terminal, the instrumental (or based on resources), and the superfluity. We represent the agent's plans by means of structured arguments whose premises are pervaded with uncertainty. We measure the strength of these arguments in order to determine the set of compatible goals. We propose two novel ways for calculating the strength of these arguments, depending on the kind of incompatibility that exists between them. The first one is the logical strength value, it is denoted by a three-dimensional vector, which is calculated from a probabilistic interval associated with each argument. The vector represents the precision of the interval, the location of it, and the combination of precision and location. This type of representation and treatment of the strength of a structured argument has not been defined before by the state of the art. The second way for calculating the strength of the argument is based on the cost of the plans (regarding the necessary resources) and the preference of the goals associated with the plans. Considering our novel approach for measuring the strength of structured arguments, we propose a semantics for the selection of plans and goals that is based on Dung's abstract argumentation theory. Finally, we make a theoretical evaluation of our proposal.

Keywords: Argumentation, Goals Selection, Uncertainty, Arguments Strength, Goals Conflicts.

1. Introduction

An intelligent agent may in general pursue multiple procedural goals at the same time[1]. In this situation, some conflicts between goals could arise, in the sense that it is not possible to pursue them simultaneously. Reasons for not pursuing some goals simultaneously are generally related to the fact that plans for reaching such goals may block each other. Consider the well-known "cleaner world" scenario, where a set of robots have the task of cleaning the dirt of an environment. Although the main goal of the

[1] A goal is procedural when there is a set of plans for achieving it. These goals are also known as achievement goals [8].

ISSN: 1137-3601 (print), 1988-3064 (on-line)

robots is to clean the environment, during the execution of this task they may pursue some other goals. Furthermore, consider that there exist uncertainties in both actions and sensing.

According to Castelfranchi and Paglieiri [11], at least three forms of incompatibility could emerge:

- *Terminal incompatibility*: Suppose that at a given moment one of the robots – let us call him BOB – has a technical defect; hence, BOB begins to pursue the goal "*going to the workshop to be fixed*". Recall that BOB is already pursuing the goal of cleaning the environment; however, if BOB wants to be fixed he has to stop cleaning. Hence, BOB cannot pursue both goals at the same time because the plans adopted for each goal lead to an inconsistency, since he needs to be operative to continue cleaning and non-operative to be fixed.

- *Instrumental or resource incompatibility*: It arises because the agents have limited resources. Suppose that BOB is in slot (2,4) and detects two dirty slots, slot (4,2) and slot (4,8). Therefore, he begins to pursue two goals: (i) cleaning slot (4,2) and (ii) cleaning slot (4,8); however, he only has battery for executing the plan of one of the goals. Consequently a conflict due to resource battery arises, and BOB has to choose which slot to clean.

- *Superfluity*: It occurs when the agent pursues two goals that lead to the same end. Suppose that BOB is in slot (2,2) and he detects dirt in slot (4,5), since it is far from its localization, he has no certainty about the kind of dirt and he begins to pursue the goal "*cleaning slot (4,5)*". Another cleaner robot – TOM – also detects the same dirty slot and he has the certainty that it is liquid dirt; however, TOM's battery is quite low, whereby he sends a message to BOB to mop slot (4,5). Thus, BOB begins to pursue the goal "*mopping slot (4,5)*". It is easy to notice that both goals have the same end, which is that slot (4,5) to be cleaned.

Argumentation is an appropriate approach for reasoning with inconsistent information [16]. The process of argumentation is based on the construction and the comparison of arguments (considering the so-called attacks among them). Argumentation has been applied for practical reasoning for the generation of desires and plans (e.g., [2][3][20][31]). In [3] and [31], the authors represent the agent's plans by means of arguments (these arguments are called instrumental arguments) and the conflicts between plans are expressed in form of attacks.

In our example, we can have an argument A representing a plan p, an argument B representing another plan p', and both attack each other. The question is: what argument will be selected? According to [5], one can measure the strength of the arguments to refine the notion of acceptability (selection) of arguments. Thus, each argument is measured and a strength value is assigned to it. Then, the arguments' strengths determine the preference of one of them.

In [3], [20], [26], and [31], the authors use instrumental arguments to represents plans and define possible attacks; however, the agent's beliefs are not pervaded with uncertainty. Besides, in [3], [20], and [31] the actions are not taken into account in the structure of the arguments. The strength of an instrumental argument is measured in [3] based on the worth of the goals that make it up and the cost of the plan with respect to the resources it needs to be achieved.

Other related works are focused on dealing with uncertainty in structured arguments. Hunter [21] assign a probability distribution over the models of the language, which are used to give a probability distribution to the arguments that are constructed using classical logic. Haenni et al. [19] also applies the approach of assigning a probability distribution over the models models of logical arguments. In [4], Amgoud and Prade propose a possibilistic logic framework where arguments are built from an uncertain knowledge base and a set of prioritized goals. In [27], Nieves et al. present an argumentation approach based on the ASP's language for encoding knowledge under imprecise or uncertain information. Chesñevar et al. [12] present P-DeLP (Possibilistic Defeasible Logic Programming), which combines features of argumentation theory and logic programming and incorporates a possibilistic treatment of uncertainty. This work was extended in [1] by applying also fuzzy techniques. Thus, the authors use PGL+, a possibilistic logic over Gödel logic extended with fuzzy constants. Schweimeier and Schroeder [32] also use fuzzy logic and propose an argumentation framework with fuzzy unification to handle uncertainty. Nevertheless, all these works are not related to goals conflicts and do not represent plans by means of arguments.

Against this background, the aim of this article is to study and propose a way of measuring the strength of instrumental arguments whose premises are pervaded of uncertainty. This will lead us to determine the set of non-conflicting plans and non-conflicting goals the agent can continue pursuing. Thus, the research questions that are addressed in this article are:

1. How to measure the strength of an instrumental argument considering that its premises have uncertain elements?, and
2. Given that we use instrumental arguments to determine the incompatibilities between goals, how the uncertainty of the elements of instrumental arguments impact on determining the set of compatible goals?

In addressing the first question, we use a coherence-based probability logic approach [30]. We assign and/or calculate a probabilistic interval for each element of the argument and the interval of the argument is calculated based on the uncertainty of its premises. Lastly, the argument's strength is calculated from this interval. The reason to choose an interval is that it can be used to represent a range of possible values and/or the aggregation of several perceptions. For example, suppose that the dirty sensor of a cleaner robot has a sensitivity error for distinguishing between a dirty slot and a stain on the slot when he is far from the observed slot. In this case, it would be better to represent the probability that such slot is dirty by using an interval. Regarding the second question, we use Dung's argumentation semantics in order to obtain the set of compatible goals. Thus, the main contributions of this article are:

- A way of measuring the strength of structured arguments whose premises are pervaded with uncertainty,
- Two different ways of measuring the strength of instrumental arguments: (i) considering their logical structure and (ii) considering the necessary resources.
- A three-dimensional representation of the logical strength, which allows the agent to compare the argument in more than one way. To the best of our knowledge, the suggested three-dimensional strength of arguments is the first one in its kind for leading with arguments that are pervaded with uncertainty, and
- A way of selecting goals based on abstract argumentation semantics.

This article has also a practical contribution since it can be applied to real engineering problems. As it can be seen in the example, this kind of approach can be used in robotic applications (e.g., [15][17][33]) in order to endow a robot with a system that allows him to recognize and decide about the goals he should pursue. Another possible application is in the spatial planning problem, which aims to rearrange the spatial environment in order to meet the needs of a society [24]. As space is a limited resource, it causes that the planner finds conflicts in the desires and expectations about the spatial environment. These desires and expectation can be modeled as a set of restrictions and conditions, which can be considered as goals (e.g., suitability, dependency, and compatibility)[6]. Thus, the planner can be seen as a software agent that has to decide among a set of conflicting goals. Although these conflicting goals are not goals the agent wants to achieve, as in the case of the robot, the agent may use this approach in order to resolve the problem and suggest a possible arrangement of the spatial environment. It could also be applied during a design process, in which inconsistencies among design objectives may arise and this results in design conflicts [10]. In this case, agents may represent designers that share knowledge and have conflicting interests. This results in a distributed design system that can be simulated as an Multi-Agent System [13][18]. This last type of application involves more than one agent; however, since there is a conflict among goals, it can be resolved by applying the proposed approach. Another interesting application is in the medication adherence problem. In [22], the authors propose a coach intelligent agent that is in charge of supporting the medical management of patients. This agent has autonomous reasoning capabilities that allow him to deal with long-term goals in the settings of medication plans.

The rest of the article is organized as follows. Next section presents some necessary technical background related to probabilistic logic. In Section 3, the main building blocks on which this approach is based are defined. Section 4 is devoted to the kinds of attacks that may occur between arguments. In

Section 5, we study and present the strength calculation proposal. Section 6 is focused on the definition of the argumentation framework and on studying how to determine the set of compatible goals by means of argumentation semantics. We present the evaluation of our proposal in terms of fulfilling the postulates of rationality in Section 7. Finally, the conclusions and future work are presented in Section 8.

2. Probabilistic background

In this section some necessary technical background is presented. It is based on probabilistic logic inference in the settings of [23] and [29].

Let $\mathcal{L}$ be a propositional vocabulary that contains a finite set of propositional symbols. $\wedge$ and $\neg$ denote the logical connectives conjunction and negation. An *event* is defined as follows. The propositional constants *false* and *true*, denoted by $\bot$ and $\top$, respectively, are events. An atomic formula or *atom* is an event. If ϕ and ψ are events, then also $\neg\phi$ and $(\phi\wedge\psi)$. A *conditional event* is an expression of the form $\psi|\phi$ and a *conditional constraint* is an expression of the form $(\psi|\phi)[l, u]$ where $l, u \in [0, 1]$ are real numbers. The event ψ is called the consequent (or head) and the event ϕ its antecedent (or body). *Probabilistic formulas* are defined as follows. Every conditional constraint is a probabilistic formula. If F and G are probabilistic formulas then also $\neg F$ and $(F \wedge G)$.

One can distinguish between classical and purely probabilistic constraints. *Classical conditional constraints* are of the kind $(\psi|\phi)[1, 1]$ or $(\psi|\phi)[0, 0]$, while *purely probabilistic conditional constraints* are of the form $(\psi|\phi)[l, u]$ with $l < 1$ and $u > 0$.

An event ϕ is *conjunctive* iff ϕ is either $\top$ or a conjunction of atoms. A *conditional event* $\psi|\phi$ is conjunctive (respectively, *1-conjunctive*) iff ψ is a conjunction of atoms (respectively, an atom) and ϕ is conjunctive. A conditional constraint $(\psi|\phi)[l, u]$ is conjunctive (respectively, 1-conjunctive) iff $\psi|\phi$ is conjunctive (respectively, 1-conjunctive).

Conjunctive conditional constraints $(\psi|\phi)[l, u]$ with $l \leq u$ are also called *probabilistic Horn clauses*, from which can be defined *probabilistic facts* and *probabilistic rules*, which are of the form $(\psi|\top)[l, u]$ and $(\psi|\phi)[l, u]$, respectively, where $\phi \neq \top$.

We use the coherence-based probability logic to propagate the uncertainty of the premises to the conclusion, more specifically, we use probabilistic `MODUS PONENS`. We denote the probabilistic closure `MODUS PONENS` inference by $\vdash_P$. Finally, the calculation of the conclusion interval is given by [29]:

$$\{(\psi|\phi)[l, u], (\phi|T)[l', u']\} \vdash_P (\psi|T)[l * l', 1 - l' + u * l']$$

3. Basics of the proposal

In this section, we present the main mental states of the agent; and define a class of structured arguments that represent plans.

In this article, the main mental states of an agent are the following finite bases:

- $\mathcal{B}$ is a finite set beliefs,
- $\mathcal{A}$ is a finite base of the actions,
- $\mathcal{G}$ is a finite base of the goals, and
- $\mathcal{R}es$ is a finite base of the resources of the agent.

Elements of $\mathcal{B}$ and $\mathcal{A}$ are probabilistic facts and elements of $\mathcal{G}$ and $\mathcal{R}es$ are atomic formulas. It holds that $\mathcal{B}, \mathcal{A}$, $\mathcal{R}es$, and $\mathcal{G}$ are pairwise disjoint. Let $\mathcal{B}^* = \{b|(b|\top)[l, u] \in B\}$ and $\mathcal{A}^* = \{a|(a|\top)[l, u] \in A\}$ be the projections sets of $\mathcal{B}$ and $\mathcal{A}$, respectively. That is, the elements of $\mathcal{B}^*$ and $\mathcal{A}^*$ are atomic formulas, which have their correspondent probabilistic conditional constraints in $\mathcal{B}$ and $\mathcal{A}$, respectively. Furthermore, the agent is also equipped with a function `PREF` $:\mathcal{G} \rightarrow [0, 1]$, which returns a real value that denotes the preference value of a given goal (0 stands for the null preference value and 1 for the maximum one) and a resource summary structure $\mathcal{R}_{sum} \subseteq \mathcal{R}es \times \mathbb{R}^+$ where the first component of a pair is a resource and the second one is the available amount of such resource. Let `AVAILABLE_RES` : $\mathcal{R}es \rightarrow \mathbb{R}^+$ be a function that returns the current amount of a given resource.

The agent has also a set of probabilistic plans that allows him to achieve his goals. In order to analize the possible incompatibilities that could arise among them, we express the plans in terms of instrumental arguments. Thus, the basic building block of an instrumental argument is a probabilistic plan rule, which includes, in the premise, a set of beliefs, a set of goals and a set of actions. All these elements are necessary for the plan to be executed and the goal in the conclusion of the rule to be achieved.

Definition 1 ***(Probabilistic plan rule)*** *A probabilistic plan rule is denoted by a probabilistic rule* $(\psi|\phi)[l,u]$ *such that* $\phi = b_1 \wedge ... \wedge b_n \wedge g_1 \wedge ... \wedge g_m \wedge a_1 \wedge ... \wedge a_l$ *and* $\psi = g$ *where* $b_i \in \mathcal{B}^*$ *(for all* $1 \leq i \leq n$*),* $g_j \in \mathcal{G}$ *(for all* $1 \leq j \leq m$*),* $a_k \in \mathcal{A}^*$ *(for all* $1 \leq k \leq l$*), and* $g \in \mathcal{G}$*. In order to avoid cycles, we require that* $\psi \neq g_1$ *...* $\psi \neq g_m$*. Besides, the number of elements of* ϕ *is finite.*

A probabilistic plan rule expresses that if $b_1 \wedge ... \wedge b_n$ are true at a certain degree, $g_1 \wedge ... \wedge g_m$ are achieved at a certain degree, and $a_1 \wedge ... \wedge a_l$ are accurately performed at a certain degree then g is achieved at a certain degree.[2].

Finally, let $\mathcal{PR}$ be the base containing the set of probabilistic plan rules.

Example 1 *Considering the scenario presented in the introduction section, let us introduce some examples of probabilistic plan rules. Suppose that the environment is a square of 4 rows by 4 columns. So, for the environment to be completely clean, all the slots have to be clean.*

First of all, let us present the beliefs, actions, and goals that are part of the premises of the probabilistic plan rules. Beliefs: $has\_refill, \neg full\_trashcan$*, and* $solid\_dirt\_1\_3$*. Action:* go_2_2*. Goals:* $be\_oper, clean\_1\_1$*, ...,* $clean\_4\_4$*,* $clean, sweep\_4\_4, in\_wshop$ *and* be_fixed*. We use goal clean to refer to the environment as a whole and we use goals* $clean\_1\_1, ..., clean\_4\_4$ *to refer to each slot of the environment. Thus, to achieve the goal clean, all the dirty slots have to be cleaned. Below, we present some probabilistic plan rules:*
- $(clean \mid be\_oper \wedge clean\_1\_1 \wedge ... \wedge clean\_4\_4)[1,1]$
- $(sweep\_4\_4 \mid \neg full\_trashcan \wedge solid\_dirt\_4\_4 \wedge go\_4\_4)[0{,}7, 0{,}9]$
- $(be\_fixed|\neg be\_oper \wedge has\_refill \wedge in\_wshop)[1,1]$

We use probabilistic facts and probabilistic plan rules to build probabilistic instrumental arguments, which represent complete plans. Like in [31], we represent this type of argument using a tree structure; however, in our definition the root is made up of a probabilistic plan rule and the leaves are either beliefs or actions. We can consider these last elements as elementary arguments, since they do not generate sub-trees. We can say that this definition of instrumental argument is new in the state of the art.

Definition 2 ***(Elementary probabilistic argument)*** *An elementary probabilistic argument is a tuple* $\langle H, (\psi|\top)[l,u]\rangle$ *where either* $(\psi|\top)[l,u] \in \mathcal{A}$ *or* $(\psi|\top)[l,u] \in \mathcal{B}$ *and* $H = \emptyset$*.*

Function `CLAIM` returns the claim ψ of a given elementary probabilistic argument. Unlike beliefs and actions, the goals that make up the premise of a probabilistic plan rule generate a tree-structure. Considering that plans may have sub-plans, it is natural that arguments may also have sub-arguments.

Definition 3 ***(Probabilistic instrumental argument, or complete plan)*** *A probabilistic instrumental argument is a tuple* $\langle \mathcal{T}, g\rangle$*, where* $\mathcal{T}$ *is a finite tree such that:*

- *The root of the tree is a structure of the form* $\langle H, g\ [l_g, u_g]\rangle$ *where:*
  - $H = (g|b_1 \wedge ... \wedge b_n \wedge g_1 \wedge ... \wedge g_m \wedge a_1 \wedge ... \wedge a_l)[l,u]$,
  - $l_g, u_g \in [0,1]$ *are real numbers that represent the upper and lower probabilities of* g*.*
- *Since* $H = (g|b_1 \wedge ... \wedge b_n \wedge g_1 \wedge ... \wedge g_m \wedge a_1 \wedge ... \wedge a_l)[l,u]$*, it has exactly* $(n+m+l)$ *children, such that* $\forall b_i$ $(1 \leq i \leq n)$ *and* $\forall a_k$ $(1 \leq k \leq l)$ *there exists an elementary probabilistic argument, and* $\forall g_j$ $(1 \leq j \leq m)$ *there exists a probabilistic instrumental argument, we can call these last arguments of sub-arguments.*

[2] Achievement goals represent a desired state that an agent wants to reach [14].

- $H, (b_1|\top[l_{b_1}, u_{b_1}]), \ldots, (b_n|\top[l_{b_n}, u_{b_n}]), H_{g_1}, ..., H_{g_m}, (a_1|\top[l_{a_1}, u_{a_1}]), \ldots, (a_l|\top[l_{a_l}, u_{a_l}]) \vdash_P g[l_g, u_g]$.

Let ARG be the set of all arguments[3] that are associated to the goals in $\mathcal{G}$. We assume that each goal has at least one argument. It is also important to mention that there could be more than one argument for a given goal. Function SUPPORT(A) returns the set of elementary probabilistic arguments, the main root of the argument, and the roots of the sub-arguments of A, CLAIM(A) returns the claim g of A, and SUB(A) returns the set of sub-arguments of A.

In order to obtain the probabilistic interval of the claim of an argument, it has to be applied the probabilistic MODUS PONENS from the leaves to the root.

Figure 1 shows the tree of the argument A whose claim is goal *be_fixed* with the sub-arguments B whose claim is goal $\neg$*be_oper*, and C whose claim is goal *in_wshop*. The values of the probabilistic intervals of the main argument and the sub-arguments are calculated using probabilistic MODUS PONENS.

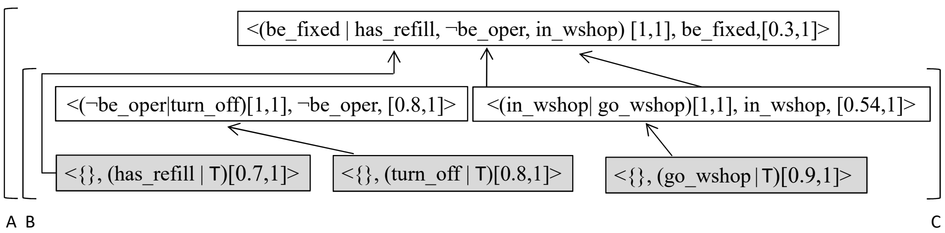

<(be_fixed | has_refill, ¬be_oper, in_wshop) [1,1], be_fixed,[0.3,1]>
<(¬be_oper|turn_off)[1,1], ¬be_oper, [0.8,1]>
<(in_wshop| go_wshop)[1,1], in_wshop, [0.54,1]>
<{}, (has_refill | T)[0.7,1]>
<{}, (turn_off | T)[0.8,1]>
<{}, (go_wshop | T)[0.9,1]>
A
B
C

Figura 1: Tree of probabilistic arguments A, B, and C. Gray-filled rectangles represent the leaves of the tree.

4. Attacks between arguments

In this section, we focus on the identification of attacks between arguments, which will lead to the identification of incompatibility among goals. The kind of attack depends on the form of incompatibility. The conflicts between arguments are defined over ARG and are captured by the binary relation $\mathcal{R}_x \subseteq$ ARG $\times$ ARG (for $x \in \{t, r, s\}$) where each sub-index denotes the form of incompatibility. Thus, t denotes the attack for terminal incompatibility, r the attack for resource incompatibility, and s the attack for superfluity. We denote with (A, B) the attack relation between arguments A and B. In other words, if $(A, B) \in \mathcal{R}_x$, it means that argument A attacks argument B.

4.1. Terminal incompatibility attack

We define the terminal incompatibility in terms of attacks among arguments. An argument A attacks another argument B when the claim of any of the sub-arguments of A is the negation of the claim of any of sub-arguments of B, and both arguments correspond to plans that allow to achieve different goals.

Definition 4 ***(Support rebuttal -*** $\mathcal{R}_t$***)*** *Let* $A, B \in$ ARG, $[H, \psi] \in$ SUPPORT(A) *and* $[H', \psi'] \in$ SUPPORT(B). *We say that* $(A, B) \in \mathcal{R}_t$ *occurs when:*

- CLAIM(A) $\neq$ CLAIM(B)*,*
- $\psi = \neg\psi'$ *such that* $\psi, \psi' \in \mathcal{B}$ *or* $\psi, \psi' \in \mathcal{A}$*, or* $\psi, \psi' \in \mathcal{G}$*.*

Sub-arguments of arguments that are involved in a support rebuttal are also involved in a support rebuttal between them and with the main arguments. Formally:

- *If* $(A, B) \in \mathcal{R}_t$ *and* $\exists C \in$ SUB(B) *and* $\exists D \in$ SUB(A)*, then* $(C, D) \in \mathcal{R}_t$ *and* $(D, C) \in \mathcal{R}_t$*.*
- *If* $(A, B) \in \mathcal{R}_t$ *and* $\exists C \in$ SUB(B)*, then* $(A, C) \in \mathcal{R}_t$ *and* $(C, A) \in \mathcal{R}_t$*.*

[3] Hereafter, for sake of simplicity, we use only argument to refer to a probabilistic instrumental argument.

Finally, it holds that $\mathcal{R}_t$ is symmetric.

Example 2 *Let $\mathcal{G} = \{clean\_1\_3, be\_fixed\}$ be two goals that robot* BOB *is currently pursuing. Figure 1 depicts the arguments A, B, and C. Figure 2 shows arguments D whose claim is goal $clean\_1\_3$ and E whose claim is goal be_oper. Since $\texttt{CLAIM}(D) = \neg\texttt{CLAIM}(B)$, the following support rebuttals arise: $\{(A, D), (D, A), (A, E), (E, A), (B, D), (D, B), (B, E), (E, B), (C, D), (D, C), (C, E), (E, C)\} \in \mathcal{R}_t$.*

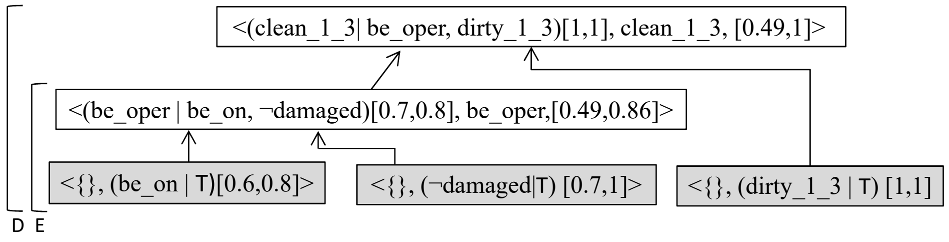

<(clean_1_3| be_oper, dirty_1_3)[1,1], clean_1_3, [0.49,1]>
<(be_oper | be_on, ¬damaged)[0.7,0.8], be_oper,[0.49,0.86]>
<{}, (be_on | ⊤)[0.6,0.8]>
<{}, (¬damaged|⊤) [0.7,1]>
<{}, (dirty_1_3 | ⊤) [1,1]
D
E

Figura 2: Probabilistic instrumental argument for goal *clean_1_3*.

4.2. Resources incompatibility attack

First of all, let us denote with $\texttt{LIST\_RES\_ARG}(A)$ the list of resources along with their respective amounts that are necessary to perform the plan represented by an argument A. We assume that each probabilistic plan rule has a list of the resources that are necessary for its performance. Thus, $\texttt{LIST\_RES\_ARG}(A)$ is the sum up of the list of every probabilistic plan rule that makes up argument A.

Now, we define when a set of instrumental arguments attack each other considering the resources. We evaluate sets of arguments that need the same resource and that are related to plans that allow to achieve different goals. For instance, if three arguments need a certain resource res, we compare the current amount of that resource in $\mathcal{R}_{sum}$ with the total amount of res the three arguments need. We use function $\texttt{NEED\_RES} : \texttt{ARG} \times \mathcal{R}es \rightarrow \mathbb{R}^+$ to figure out the amount of a resource that a given argument needs. Finally recall that function AVAILABLE_RES returns the current amount of a given resource.

Definition 5 ***(Resource attack)*** *Let A and B be two arguments that need a same resource res, such that $A, B \in \texttt{ARG}$ and $\texttt{CLAIM}(A) \neq \texttt{CLAIM}(B)$. We say that $(A, B) \in \mathcal{R}_r$ when $\texttt{NEED\_RES}(A, res) + \texttt{NEED\_RES}(B, res) > \texttt{AVAILABLE\_RES}(res)$. It holds that $\mathcal{R}_r$ is symmetric.*

4.3. Superfluity attack

Superfluity can be defined in terms of the superfluous attack. In this attack, the claims of arguments are evaluated. Thus, an argument A attacks another argument B when they have the same claim. Since each argument may have sub-arguments, this attack is also inherited by the sub-arguments.

Definition 6 ***(Superfluous attack - $\mathcal{R}_s$)*** *Let $A, B \in \texttt{ARG}$. We say that $(A, B) \in \mathcal{R}_s$ occurs when:*

- $\texttt{CLAIM}(A) = \texttt{CLAIM}(B)$,
- $\texttt{SUPPORT}(A) \neq \texttt{SUPPORT}(B)$.

Like in the terminal support rebuttal, the sub-arguments of arguments that are involved in a superfluous attack are also involved in a superfluous attack. Finally, it holds that $\mathcal{R}_s$ is symmetric.

5. Strength calculation

In this section, the way for calculating the strength of instrumental arguments is presented. We present two different approaches for the strength calculation: (i) considering only the logical structure of the instrumental argument, and (ii) considering the necessary resources.

5.1. Considering the logical structure

We base on the approach of Pfeifer [28] to calculate the strength of the arguments. This approach uses the values of the probabilistic interval of the claim of the arguments to make the calculation and is based on two criteria: the *precision* of the interval and the *location* of it. Thus, the higher the precision of the interval is and the closer to 1 the location of the interval is, the stronger the argument is. We use the notions of precision, location and the combination of both to measure the arguments from different point of views. Then, we next present a three dimensional measure of the arguments strength.

Definition 7 ***(Logical Strength)*** *Let $A = \langle \mathcal{T}, g \rangle$ be an argument and $\langle H, g[l_g, u_g] \rangle$ be the root of $\mathcal{T}$. The logical strength of A –denoted by* $\texttt{STRENGTH}(A)$*– is a three-dimensional vector $\langle \texttt{CO}(A), \texttt{PR}(A), \texttt{LO}(A) \rangle$ where:*

$$\begin{aligned}
&\texttt{LO}(A) = \left(\tfrac{l_g+u_g}{2}\right)\\
&\texttt{PR}(A) = 1 - (u_g - l_g)\\
&\texttt{CO}(A) = (1 - (u_g - l_g)) \times \left(\tfrac{l_g+u_g}{2}\right)
\end{aligned}$$

Example 3 *Consider the arguments of Figure 1. The strength values of the main argument and its sub-arguments are:* $\texttt{STRENGTH}(A) = \langle 0{,}195, 0{,}3, 0{,}65 \rangle$*,* $\texttt{STRENGTH}(B) = \langle 0{,}72, 0{,}8, 0{,}9 \rangle$*, and* $\texttt{STRENGTH}(C) = \langle 0{,}42, 0{,}54, 0{,}77 \rangle$*.*

5.2. Considering the resources

For measuring the strength of an argument considering the resources, we take into account the preference value of all the goals involved in the probabilistic instrumental argument, the value of the combination of precision and location of its probabilistic interval, and the cost of achieving the goal in the claim of the argument. Recall that function $\texttt{NEED\_RES}(A, res)$ returns the amount of resource res that argument A needs. Thus, the cost of an argument A is calculated as follows: $\texttt{COST}(A) = \sum_{i=1}^{i=n} \texttt{NEED\_RES}(A, res_i)$, where n is the quantity of resources argument A needs.

Definition 8 ***(Utility)***[4] *Let $A = \langle \mathcal{T}, g \rangle$ be an instrumental argument, the utility strength of A –denoted by* $\texttt{UTILITY}(A)$*– is calculated as follows:*

$$\texttt{UTILITY}(A) = \sum_{g_i \in \texttt{SUPPORT}(A)} \texttt{PREF}(g_i) + \texttt{COMB}(A) - \texttt{COST}(A) \qquad (1)$$

Notice that we have two positive components and one negative. Figure 3 shows the behaviour of the utility strength[5], the X axis denotes the sum of the positive components, the Y axis denotes the utility strength value, and each columns group denotes the preference plus the combined strength value. Thus, in the graphic, we can notice that:

- Most of the strength values are negative, this is because the values assigned to the preference of the goals and the combined strength of the argument are real values between zero and one, while the values of the cost are positive real numbers with (possibly) values greater than one.

- In each group of columns, the strongest arguments are displayed with the darkest blue. In the first columns group, this color does not appear because both values are zero.

- The greater the sum of the preference and the combined strength is and the lower the cost is, the greater the utility strength is. In the figure, the highest value of the strength is given when the sum of the positive components is 10 and the cost is 0. Otherwise, the lowest value of the utility strength occurs when the sum of the positive elements is 0 and the cost is 40.

[4] Like in [31], we call this kind of strength of utility.

[5] For a graphical representation of the behaviour of the logical strength, the reader is referred to [28] pag. 11.

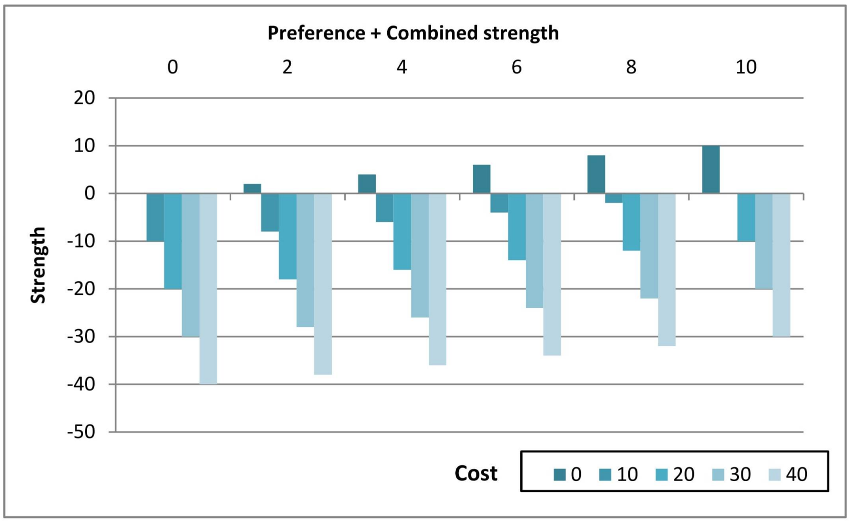


Figura 3: Example values for demonstrating the behaviour of the utility strength.

5.3. Preference between arguments

We can now compare two arguments based on these values. This comparison will determine the preference of an argument over another one. Taking into account these three dimensions is specially useful when there is a tie in the value of $\texttt{CO}(A)$. Consider, for example, that $\texttt{PR}(A) = 0{,}6$ and $\texttt{LO}(A) = 0{,}4$, and $\texttt{PR}(B) = 0{,}4$ and $\texttt{LO}(B) = 0{,}6$; hence, $\texttt{CO}(A) = \texttt{CO}(B) = 0{,}24$. In this case, the agent may determine the strongest argument comparing the other values.

Definition 9 ***(Preferred argument)*** *Given two arguments A and B. Considering the logical strength, an argument A is more preferred than argument B (denoted by $A \succ B$) if:*
- $\texttt{CO}(A) > \texttt{CO}(B)$*, or*
- $\texttt{CO}(A) = \texttt{CO}(B)$ *and* $\texttt{LO}(A) = \texttt{LO}(B)$ *and* $\texttt{PR}(A) > \texttt{PR}(B)$*, or*
- $\texttt{CO}(A) = \texttt{CO}(B)$ *and* $\texttt{PR}(A) = \texttt{PR}(B)$ *and* $\texttt{LO}(A) > \texttt{LO}(B)$*.*

Considering the utility strength, A is more preferred than B (denoted by $A \succ_R B$) if $\texttt{UTILITY}(A) > \texttt{UTILITY}(B)$*.*

The election of which value the agent has to compare first (either the precision value or the location one) depends on his interests.

Regarding the logical strength of an argument and the preference relation, the following property shows the relation that exists between the logical strength of a main argument and the logical strength of its sub-arguments.

Property 1 *Let $A = \langle \mathcal{T}, g \rangle$ be an argument and A_1, ..., A_n all the sub-arguments of A. For all A_i, $A_i \succeq A$, where $1 \leq i \leq n$.*

In the Example 3, one can notice that the logical strength of both sub-arguments is greater than the logical strength of the main argument.

6. Goals selection

In this section, we present an argumentation framework which integrates all the arguments and attacks in a unique framework and will be used to determine the set of compatible goals.

Definition 10 ***(Argumentation framework)*** *An argumentation framework is a tuple $\mathcal{AF} = \langle \texttt{ARG}, \mathcal{R} \rangle$, where $\mathcal{R} = \mathcal{R}_t \cup \mathcal{R}_s \cup \mathcal{R}_r$.*

Regarding $\mathcal{R}$, it could happen that two arguments attack each other in more than one way. For example, suppose that G and F are two arguments such that $(G, F) \in \mathcal{R}_t$ and $(G, F) \in \mathcal{R}_s$. In these cases, we consider multiple attacks between two arguments as a unique attack in $\mathcal{AF}$.

Hitherto, we have considered that all attacks are symmetrical. However, the strength values of the arguments allow the agent to break such symmetry. Therefore, depending on these values some attacks may be considered successful. Thus, the process of goals selection starts by modifying the attack relation $\mathcal{R}$ taking into account the successful attacks.

Definition 11 ***(Successful attack)***[6] *Let $A, B \in \texttt{ARG}$ be two arguments, we say that A successfully attacks B when $A \succ B$ or $A \succ_R B$.*

Example 4 *(Cont. Example 2) Let $\mathcal{AF}' = \langle\{A, B, C, D, E\}, \{(D, A), (B, D), (C, D), (E, A), (B, E), (E, C)\}\rangle$ be the AF after considering the successful attack definition. Figure 4 shows the graph representation of this framework and the three-dimensional strength of the arguments.*

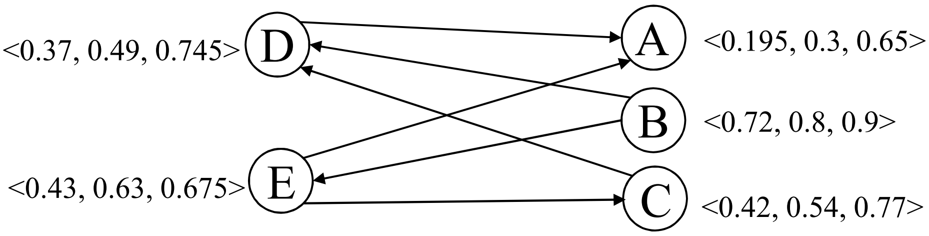


Figura 4: Graph representation of the argumentation framework $\mathcal{AF}$. Nodes represent the arguments and edges the attacks. The strength of each argument is located next to each node.

The next step of the selection process is applying an argumentation semantics on the resultant AF, that is, after considering the successful attacks.

In argumentation theory, acceptability semantics are in charge of returning sets of arguments called extensions which are internally consistent. In order to obtain the set of goals that have no conflicts among them, we will apply the notion of conflict-freeness over the set of arguments to guarantee that no incompatible argument (i.e., plan) is returned by the semantics, and consequently no incompatible goal. Another notion, that we believe is important, is related to the number of compatible goals the agent can continue pursuing; in this way, the idea is to maximize this number. Thus, we propose to apply a semantics based on the notion of conflict-freeness and that also returns those extensions that maximize the number of goals to be pursued.

Definition 12 ***(Semantics)*** *Given $\mathcal{AF} = \langle \texttt{ARG}, \mathcal{R}' \rangle$ where $\mathcal{R}' \subseteq \mathcal{R}$ is the modified attack relation after considering the successful attack. Let $\mathcal{E} \subseteq \texttt{ARG}$:*

- *$\mathcal{E}$ is conflict-free if $\forall A, B \in \mathcal{E}$, $(A, B) \notin \mathcal{R}'$. Let $\mathcal{CF}$ be the set of all the conflict-free extensions,*
- $\texttt{MAX\_GOAL} : \mathcal{CF} \rightarrow \mathcal{CF}'$, *where $\mathcal{CF}' = 2^{\mathcal{CF}}$. This function takes as input a set of conflict-free sets and returns those maximal (w.r.t set inclusion) sets that allow the agent to achieve the greatest number of pursuable goals. Sub-goals are not taken in to account in this function.*
- $\texttt{MAX\_UTIL} : \mathcal{CF}' \rightarrow 2^{\mathcal{CF}'}$. *This function takes as input the set of conflict-free sets $\mathcal{CF}'$ and returns those with the maximum utility for the agent in terms of preference value. The utility of each extension is calculated by summing up the preference value of the main goals of the extension. In this function, sub-goals are not taken in to account either.*

The final step of the selection process is to obtain the set of compatible goals from the set of compatible plans.

[6] In other works, it is called a defeat relation [25].

Definition 13 ***(Projection function)*** *Let $\mathcal{CF}''$ be a set of extensions returned by* `MAX_UTIL`*. Function* `COMP_GOALS` $: \mathcal{CF}'' \rightarrow 2^{\mathcal{G}}$ *takes as input an extension of $\mathcal{CF}'$ and returns the set of compatible goals that are associated to the arguments in the extension.*

Notice that function `COMP_GOALS` is applied to each extension of $\mathcal{CF}'$; hence, there could be more than one different set of compatible goals. In such case, the agent has to choose the set of compatible goals he will continue to pursue according to his interests.

Example 5 *(Cont. Example 4) After applying the conflict-free semantics the conflict-free extensions are $\mathcal{CF} = \{\{\}, \{C\}, \{B\}, \{B, C\}, \{A\}, \{A, C\}, \{A, B\}, \{A, B, C\}, \{E\}, \{D\}, \{D, E\}\}$. Then we apply* `MAX_GOAL`$(\mathcal{CF})$ *and we obtain: $\mathcal{CF}' = \{\{A, B, C\}, \{D, E\}\}$. Extension $\{A, B, C\}$ allows the agent to achieve the top-goal be_fixed whereas extension $\{D, E\}$ allows the agent to achieve the top-goal $clean\_1\_3$. Since there is a tie to break, the agent employs the preference value of the main goal. Assume that* `PREF`$(\{D, E\}) = 0{,}9$ *and* `PREF`$(\{A, B, C\}) = 0{,}75$*. Consequently,* `MAX_UTIL`$(\{A, B, C\}, \{D, E\}) = \{D, E\}$*.*

Finally, we apply the projection function: `COMP_GOALS`$(\{D, E\}) = \{clean\_1\_3, be\_oper\}$*.*

7. Evaluation of the approach

In this section, we evaluate the proposed approach and prove that it satisfies the rationality postulates proposed in [9]. Firstly, let us define the following notation:
$\texttt{BEL}(\mathcal{E}) = \bigcup_{A \in \mathcal{E}}(\texttt{BODY}(\texttt{SUPPORT}(A)) \cap \mathcal{B})$,
$\texttt{ACT}(\mathcal{E}) = \bigcup_{A \in \mathcal{E}}(\texttt{BODY}(\texttt{SUPPORT}(A)) \cap \mathcal{A})$
$\texttt{GOA}(\mathcal{E}) = \bigcup_{A \in \mathcal{E}}(\texttt{BODY}(\texttt{SUPPORT}(A)) \cap \mathcal{G})$

It is important to understand the following definition before presenting the results because both consistency and closure are specified based on justified conclusions. A goal that is the conclusion of an argument in any extension can be regarded as a justified conclusion, even if it is not in all extensions. From a more restrictive point of view, a goal can be regarded as a justified conclusion when it is the conclusion of an argument that belongs to all the extensions.

Definition 14 ***(Justified conclusions)*** *Let $\mathcal{AF} = \langle \texttt{ARG}, \mathcal{R} \rangle$ be a general AF and $\{\mathcal{E}_1, ..., \mathcal{E}_n\}$ $(n \geq 1)$ be its set of extensions under the conflict-free semantics.*

- $\texttt{CONCS}(\mathcal{E}_i) = \{\texttt{CLAIM}(A) | A \in \mathcal{E}_i\} (1 \leq i \leq n)$.
- $\texttt{Output} = \bigcap_{i=1,...,n} \texttt{CONCS}(\mathcal{E}_i)$.

$\texttt{CONCS}(\mathcal{E}_i)$ *denotes the justified conclusions for a given extension $\mathcal{E}_i$ and* `Output` *denotes the conclusions that are supported by at least one argument in each extension.*

An important property required in [9] is direct consistency. An argumentation system satisfies direct consistency if its set of justified conclusions and the different sets of conclusions corresponding to each extension are consistent. This property is important in our approach because it guarantees that the agent will only pursue non-conflicting goals.

Theorem 1 ***(Direct consistency)*** *Let $\mathcal{AF} = \langle \texttt{ARG}, \mathcal{R} \rangle$ be a general AF and $\mathcal{E}_1, ..., \mathcal{E}_n$ its conflict-free extensions. $\forall \mathcal{E}_i, i = 1, ..., n$, it holds that:*

1. *The set of beliefs* $\texttt{BEL}(\mathcal{E}_i)$ *is a consistent set of literals.*
2. *The set of actions* $\texttt{ACT}(\mathcal{E}_i)$ *is a consistent set of literals.*
3. *The set of goals* $\texttt{GOA}(\mathcal{E}_i)$ *is a consistent set of literals.*
4. *The set of goals* $\texttt{GOA}(\mathcal{E}_i)$ *has no superfluous conflicting goals.*

Proof 1 *Let $\mathcal{E}_i$ be an extension of $\mathcal{AF}$.*

1. Let us show that $\texttt{BEL}(\mathcal{E}_i)$ *is a consistent set of literals. Suppose that* $\texttt{BEL}(\mathcal{E}_i)$ *is inconsistent. This means that* $\exists b, \neg b \in \texttt{BEL}(\mathcal{E}_i)$. *Consider that* $\exists\ A, B \in \mathcal{E}_i$ *such that* $[\{\}, b] \in \texttt{SUPPORT}(A)$ *and* $[\{\}, \neg b] \in \texttt{SUPPORT}(B)$. *This means that there is a supports rebuttal between A and B due to b and $\neg b$. Since it contradicts the fact that $\mathcal{E}_i$ is conflict-free, we can say that* $\texttt{BEL}(\mathcal{E}_i)$ *is consistent.*

2. Let us show that $\texttt{ACT}(\mathcal{E}_i)$ *is a consistent set of literals. Suppose that* $\texttt{ACT}(\mathcal{E}_i)$ *is inconsistent. This means that* $\exists\ a, \neg a \in \texttt{ACT}(\mathcal{E}_i)$. *Consider that* $\exists A, B \in \mathcal{E}_i$ *such that* $[\{\}, a] \in \texttt{SUPPORT}(A)$ *and* $[\{\}, \neg a] \in \texttt{SUPPORT}(B)$. *This means that there is a supports rebuttal between A and B due to a and $\neg a$. Since it contradicts the fact that $\mathcal{E}_i$ is conflict-free, we can say that* $\texttt{ACT}(\mathcal{E}_i)$ *is consistent.*

3. Let us show that $\texttt{GOA}(\mathcal{E}_i)$ *is a consistent set of literals. Suppose that* $\texttt{GOA}(\mathcal{E}_i)$ *is inconsistent. This means that* $\exists\ g, \neg g \in \texttt{GOA}(\mathcal{E}_i)$. *Consider that* $\exists\ A, B \in \mathcal{E}_i$ *such that* $[\{\}, g] \in \texttt{SUPPORT}(A)$ *and* $[\{\}, \neg g] \in \texttt{SUPPORT}(B)$. *This means that there is a supports rebuttal between A and B due to g and $\neg g$. Since it contradicts the fact that $\mathcal{E}_i$ is conflict-free, we can say that* $\texttt{GOA}(\mathcal{E}_i)$ *is consistent.*

4. Let us show that $\texttt{GOA}(\mathcal{E}_i)$ *has no superfluous conflicting goals. Suppose that* $\exists g, g' \in \texttt{GOA}(\mathcal{E}_i)$ *such that g and g' are superfluous goals. Since g and g' are superfluous goals, it means that* $\exists A, B \in \mathcal{E}_i$ *such that* $\texttt{CLAIM}(A) = \texttt{CLAIM}(B) = g = g'$ *and* $\texttt{SUPPORT}(A) \neq \texttt{SUPPORT}(B)$. *Hence, there is a superfluous attack between A and B. This contradicts the fact that $\mathcal{E}_i$ is conflict free.*

Finally, since all the extensions obtained from $\mathcal{AF}$ are consistent, then $\texttt{OUTPUT}$ *is also consistent.*

Next property is closure, the idea of closure is that the set of justified conclusions of every extension should be closed under the set of plan rules $\mathcal{PR}$. That is, if g is a conclusion of an extension and there exists a plan rule $g \to g'$, then g' should also be a conclusion of the same extension. Next definition states the closure of the set of plan rules of the agent.

Definition 15 ***(Closure of*** $\mathcal{PR}$***)*** *Let* $\mathcal{F} \subseteq \mathcal{B}^* \cup \mathcal{A}^* \cup \mathcal{G}$. *The closure of $\mathcal{F}$ under the set $\mathcal{PR}$ of plan rules, denoted by* $\mathcal{C}l_{\mathcal{PR}}(\mathcal{F})$, *is the smallest set such that:*

- $\mathcal{F} \subseteq \mathcal{C}l_{\mathcal{PR}}(\mathcal{F})$
- *If* $(g|b_1 \wedge ... \wedge b_n \wedge g_1 \wedge ... \wedge g_m \wedge a_1, ... \wedge a_l)[l, u] \in \mathcal{PR}$ *and* $b_1, ..., b_n, g_1, ..., g_m, a_1, ..., a_l \in \mathcal{C}l_{\mathcal{PR}}(\mathcal{F})$ *then* $g \in \mathcal{C}l_{\mathcal{PR}}(\mathcal{F})$.

If $\mathcal{F} = \mathcal{C}l_{\mathcal{PR}}(\mathcal{F})$, *then $\mathcal{F}$ is said to be closed under the set $\mathcal{PR}$.*

In our approach closure is important because it guarantees that all the goals that can be inferred from $\mathcal{PR}$ be evaluated in terms of their possible conflicts, which in turn guarantees that the agent only will pursue non-conflicting goals.

Theorem 2 ***(Closure)*** *Let $\mathcal{PR}$ be a set of probabilistic plan rules,* $\mathcal{AF} = \langle \texttt{ARG}, \mathcal{R}' \rangle$ *be an argumentation framework built from $\mathcal{PR}$.* $\texttt{Output}$ *is its set of justified conclusions, and* $\mathcal{E}_1, ..., \mathcal{E}_n$ *its extensions under the conflict-free semantics. $\mathcal{AF}$ satisfies closure iff:*

1. $\texttt{CONCS}(\mathcal{E}_i) = \mathcal{C}l_{\mathcal{PR}}(\texttt{CONCS}(\mathcal{E}_i))$ *for each* $1 \leq i \leq n$.
2. $\texttt{Output} = \mathcal{C}l_{\mathcal{PR}}(\texttt{Output})$.

Proof 2 *Let us call* $\texttt{ARG}_{Cl}$ *the arguments that can be built from* $\texttt{BEL}(\mathcal{E}) \cup \texttt{ACT}(\mathcal{E}) \cup \texttt{GOA}(\mathcal{E})$ *and $\mathcal{PR}$.*

Given that $\texttt{CONCS}(\mathcal{E}_i) = \mathcal{C}l_{\mathcal{PR}}(\texttt{CONCS}(\mathcal{E}_i))$, *we will proof that* $\forall \mathcal{E}_i, i = 1, ..., n$, *it holds that* $\mathcal{E}_i = \texttt{ARG}_{Cl}$; *thus, we will first proof that* $\mathcal{E}_i \subseteq \texttt{ARG}_{Cl}$ *and then that* $\texttt{ARG}_{Cl} \subseteq \mathcal{E}_i$.

1. $\mathcal{E}_i \subseteq \texttt{ARG}_{Cl}$: *This is trivial.*
2. $\texttt{ARG}_{Cl} \subseteq \mathcal{E}_i$: *Suppose that* $\texttt{ARG}_{Cl} \nsubseteq \mathcal{E}_i$; *hence,* $\exists A \in \texttt{ARG}_{Cl}$ *such that* $A \notin \mathcal{E}_i$. *Since* $A \notin \mathcal{E}_i$, *it means that* $\exists B \in \mathcal{E}_i$ *such that* $(B, A) \in \mathcal{R}'$ *or* $(A, B) \in \mathcal{R}'$. *There are three situation in which this happens, one for each form of incompatibility:*

a*) When the attack is a supports rebuttal ($(B, A) \in \mathcal{R}_t$ or $(A, B) \in \mathcal{R}_t$): This means that $\exists[H, \psi] \in$ $\mathtt{SUPPORT}(B)$ and $\exists[H', \psi'] \in \mathtt{SUPPORT}(A)$ such that $\psi = \neg\psi'$, considering that $\psi, \psi' \in \mathcal{B}$ or $\psi, \psi' \in \mathcal{G}$ or $\psi, \psi' \in \mathcal{A}$. Since both A and B are built from $\mathtt{BEL}(\mathcal{E}_i)$, $\mathtt{ACT}(\mathcal{E}_i)$, and $\mathtt{GOA}(\mathcal{E}_i)$ this would mean that $\mathtt{BEL}(\mathcal{E}_i)$, $\mathtt{ACT}(\mathcal{E}_i)$, or $\mathtt{GOA}(\mathcal{E}_i)$ are inconsistent, which contradicts the first, second, and third items of Theorem 1.*

b*) When it is a superfluous attack ($(B, A) \in \mathcal{R}_s$ or $(A, B) \in \mathcal{R}_s$): Let $g = \mathtt{CLAIM}(A)$ and $g' = \mathtt{CLAIM}(B)$. Since there is a superfluous attack between A and B, it means that g and g' are superfluous goals. Given that both arguments are built from $\mathtt{BEL}(\mathcal{E}_i)$, $\mathtt{ACT}(\mathcal{E}_i)$, and $\mathtt{GOA}(\mathcal{E}_i)$, we can say that their conclusions are also part of $\mathtt{GOA}(\mathcal{E}_i)$, i.e. $g, g' \in \mathtt{GOA}(\mathcal{E}_i)$. This contradicts the last item of Theorem 1, which proofs that there is no superfluous conflicting goals in $\mathtt{GOA}(\mathcal{E}_i)$.*

The last property our proposal should satisfy is indirect consistency. This property means that (i) the closure under the set of probabilistic plan rules of the set of justified conclusions is consistent, and (ii) for each extension, the closure under the set of probabilistic plan rules of its conclusions is consistent.

Theorem 3 ***(Indirect consistency)*** *Let $\mathcal{PR}$ be a set of probabilistic plan rules and $\mathcal{AF} = \langle \mathtt{ARG}, \mathcal{R}' \rangle$. $\mathtt{OUTPUT}$ is its set of justified conclusions, and $\mathcal{E}_1, ..., \mathcal{E}_n$ its conflict-free extensions under the conflict-free semantics. $\mathcal{AF} = \langle \mathtt{ARG}, \mathcal{R}' \rangle$ satisfies indirect consistency iff:*
- $Cl_{\mathcal{PR}}(\mathtt{CONCS}(\mathcal{E}_i))$ is consistent for each $1 \leq i \leq n$.
- $Cl_{\mathcal{PR}}$ ($\mathtt{OUTPUT}$) is consistent.

Proof 3 *Based on Proposition 7 defined in [9][7], we can say that our proposal satisfies indirect inconsistency since it satisfies closure and direct consistency.*

8. Conclusions and future work

This article presents: (i) a way for measuring instrumental arguments which components are pervaded of uncertainty and (ii) an argumentation-based approach for selecting compatible goals from a set of incompatible ones. We use abstract argumentation theory because we have noticed that the problem of goals selection can be compared to the problem of calculating an extension in abstract argumentation. Therefore, we have adapted some concepts of abstract argumentation to our problem.

With respect to the research questions presented in the introduction, we can now state the following:

1. In order to represent the uncertainty of the elements of an argument, we use probabilistic intervals, which express the certainty degree of the beliefs, actions, and goals that made up the argument. Since the structure of an argument is a tree, we apply probabilistic `MODUS PONENS` from the leaves to the root to obtain the probabilistic interval of the goal that is the claim of the argument. We represent the logical strength of an argument by means of a three-dimensional vector, which includes the values of the precision and the location of the interval of the claim, and the combined value of both precision and location. Besides, we calculate the utility strength of an argument based on the resources that are necessary for performing the plan associated to the arguments. Both types of strength determine the successful attacks in the argumentation framework.

2. The uncertainty of the elements that make up an argument impacts on the final set of compatible goals in the following way: the set of compatible goals depends on the attack relation among the arguments. We consider that an attack is successful when an argument is more (or equal) preferred than other. Since the preference relation is determined using the logical strength, this means that the certainty degree determines if an attack is successful or not.

Comparing with related work, we can state the following:

[7] Extracted from [9]: "**Proposition 7**. Let $\langle A, Def \rangle$ be an argumentation system. If $\langle A, Def \rangle$ satisfies closure and direct consistency, then it also satisfies indirect consistency.".

- In [3], [20], and [31], they use arguments to represent plans, determine conflicts, and select compatible plans. The main differences of our proposal with these approaches are: (i) the structure of our arguments includes beliefs and actions, which are not considered in the related works, and (ii) they deal with the conflicts that are similar to the attack defined for terminal incompatibility; however, the superfluous attacks not taken into account.

- Unlike our proposal, [3], [20], [26], and [31] do not consider uncertainty in the elements of the arguments.

- Our approach is based on probabilistic logic; however, unlike [21] and [19], we use an interval representation of the probabilistic value of the components of the arguments.

- Finally, with respect to the strength calculation of arguments, in [3], it is measured based on the worth of the goals that made up it and the cost of the plan in relation to the resources it needs to be achieved. We calculate the strength based on the probabilistic values and propose a three-dimensional measure that allows to evaluate an argument from different perspectives, which is an advantage of our proposal.

A direct future work of this research is its practical application on the medical adherence problem that is being tackled by researchers of the Department of Computing Science together with the Department of Community Medicine and Rehabilitation of the Umea University. In this project, a Medication Coach Intelligent Agent (MCIA) is being developed [22]. Such agent perceives the environment through a smart augmented reality device –the Microsoft HoloLens– and has autonomous reasoning capabilities. We believe that the proposed approach fits perfectly in this project because the MCIA has to deal with uncertain information perceived through the HoloLens sensors and uses such information during his reasoning cycle, which includes the selection of goals.

Another future (more theoretical) research direction is related to probability logic. We have used the coherence-based probability logic to propagate the uncertainty from the support to the claim of the argument. We would like to study other approaches of probabilistic logic in order to obtain tighter intervals, if possible.

Finally, we plan to implement this proposal in Java and want to study how to integrate it with JASON [7], a BDI-based platform for developing intelligent agents.

Acknowledgements

The first author is founded by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

Referencias

[1] Teresa Alsinet, Carlos I Chesnevar, Lluís Godo, Sandra Sandri, and Guillermo Simari. Formalizing argumentative reasoning in a possibilistic logic programming setting with fuzzy unification. *International Journal of Approximate Reasoning*, 48(3):711–729, 2008.

[2] Leila Amgoud. A formal framework for handling conflicting desires. In *ECSQARU*, volume 2711, pages 552–563. Springer, 2003.

[3] Leila Amgoud, Caroline Devred, and Marie-Christine Lagasquie-Schiex. A constrained argumentation system for practical reasoning. In *International Workshop on Argumentation in Multi-Agent Systems*, pages 37–56. Springer, 2008.

[4] Leila Amgoud and Henri Prade. Using arguments for making decisions: A possibilistic logic approach. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 10–17. AUAI Press, 2004.

[5] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3):413–436, 2009.

[6] Saeed Behzadi and Ali A Alesheikh. Introducing a novel model of belief–desire–intention agent for urban land use planning. *Engineering Applications of Artificial Intelligence*, 26(9):2028–2044, 2013.

[7] Rafael H Bordini and Jomi F Hübner. Bdi agent programming in agentspeak using jason. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 143–164. Springer, 2005.

[8] Lars Braubach, Alexander Pokahr, Daniel Moldt, and Winfried Lamersdorf. Goal representation for bdi agent systems. In *ProMAS*, volume 3346, pages 44–65. Springer, 2004.

[9] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms, 2007.

[10] Baris Canbaz, Bernard Yannou, and Pierre-Alain Yvars. Preventing design conflicts in distributed design systems composed of heterogeneous agents. *Engineering Applications of Artificial Intelligence*, 28:142–154, 2014.

[11] Cristiano Castelfranchi and Fabio Paglieri. The role of beliefs in goal dynamics: Prolegomena to a constructive theory of intentions. *Synthese*, 155(2):237–263, 2007.

[12] Carlos I Chesñevar, Guillermo R Simari, Teresa Alsinet, and Lluís Godo. A logic programming framework for possibilistic argumentation with vague knowledge. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 76–84. AUAI Press, 2004.

[13] Camelia Chira, Ovidiu Chira, and Thomas Roche. Multi-agent support for distributed engineering design. In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 155–164. Springer, 2005.

[14] Mehdi Dastani, M Birna Van Riemsdijk, and Michael Winikoff. Rich goal types in agent programming. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 405–412. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[15] Angela Davids. Urban search and rescue robots: from tragedy to technology. *IEEE Intelligent systems*, 17(2):81–83, 2002.

[16] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[17] Luis Emmi, Mariano Gonzalez-de Soto, Gonzalo Pajares, and Pablo Gonzalez-de Santos. New trends in robotics for agriculture: integration and assessment of a real fleet of robots. *The Scientific World Journal*, 2014:21 pages, 2014.

[18] LQ Fan, A Senthil Kumar, Bhat Nikhil Jagdish, and Shung-Hwee Bok. Development of a distributed collaborative design framework within peer-to-peer environment. *Computer-Aided Design*, 40(9):891–904, 2008.

[19] Rolf Haenni. Probabilistic argumentation. *Journal of Applied Logic*, 7(2):155–176, 2009.

[20] Joris Hulstijn and Leendert WN van der Torre. Combining goal generation and planning in an argumentation framework. In *NMR*, pages 212–218, 2004.

[21] Anthony Hunter. A probabilistic approach to modelling uncertain logical arguments. *International Journal of Approximate Reasoning*, 54(1):47–81, 2013.

[22] Martin Ingeson, Madeleine Blusi, and Juan Carlos Nieves. Smart augmented reality mhealth for medication adherence. In *Proceedings of the First Joint Workshop on AI in Health, co-located with AAMAS 2018*, pages 157–168, 2018.

[23] Gabriele Kern-Isberner and Thomas Lukasiewicz. Combining probabilistic logic programming with the power of maximum entropy. *Artificial Intelligence*, 157(1-2):139–202, 2004.

[24] Arend Ligtenberg, Monica Wachowicz, Arnold K Bregt, Adrie Beulens, and Dirk L Kettenis. A design and application of a multi-agent system for simulation of multi-actor spatial planning. *Journal of Environmental Management*, 72(1):43–55, 2004.

[25] Sanjay Modgil and Henry Prakken. The aspic+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[26] M Morveli-Espinoza, Juan Carlos Nieves, A Possebom, Josep Puyol-Gruart, and Cesar Augusto Tacla. An argumentation-based approach for identifying and dealing with incompatibilities among procedural goals. *International Journal of Approximate Reasoning*, 105:1–26, 2019.

[27] Juan Carlos Nieves, Mauricio Osorio, and Ulises Cortés. Modality-based argumentation using possibilistic stable models. *CMNA VII-Computational Models of Natural Argument*, 2007.

[28] Niki Pfeifer. On argument strength. In *Bayesian Argumentation*, pages 185–193. Springer, 2013.

[29] Niki Pfeifer and Gernot D Kleiter. Inference in conditional probability logic. *Kybernetika*, 42(4):391–404, 2006.

[30] Niki Pfeifer and Gernot D Kleiter. Framing human inference by coherence based probability logic. *Journal of Applied Logic*, 7(2):206–217, 2009.

[31] Iyad Rahwan and Leila Amgoud. An argumentation based approach for practical reasoning. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 347–354. ACM, 2006.

[32] Ralf Schweimeier and Michael Schroeder. Fuzzy unification and argumentation for well-founded semantics. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 102–121. Springer, 2004.

[33] Gurkan Tuna, V Cagri Gungor, and Kayhan Gulez. An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, 13:54–68, 2014.

Inteligencia Artificial 22(64), (2019), 63-84
doi: 10.4114/intartif.vol22iss64pp63-84

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

An Efficient Probability Estimation Decision Tree Postprocessing Method for Mining Optimal Profitable Knowledge for Enterprises with Multi-Class Customers

Janapati Naga Muneiah[1,2,A] and Ch D V Subba Rao[3,B]

[1] Research Scholar, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India.
[2] Associate Professor, Department of Computer Science and Engineering, Chadalawada Ramanamma Engineering College, Tirupati, Andhra Pradesh, India
[3] Professor, Department of Computer Science and Engineering, Sri Venkateswara University College of Engineering, Tirupati, Andhra Pradesh, India.
[A] nagamuni513@gmail.com, [B]chdvsrao@svuniversity.ac.in

Abstract Enterprises often classify their customers based on the degree of profitability in decreasing order like C_1, C_2, ..., C_n. Generally, customers representing class C_n are zero profitable since they migrate to the competitor. They are called as attritors (or churners) and are the prime reason for the huge losses of the enterprises. Nevertheless, customers of other intermediary classes are reluctant and offer an insignificant amount of profits in different degrees and lead to uncertainty. Various data mining models like decision trees, etc., which are built using the customers' profiles, are limited to classifying the customers as attritors or non-attritors only and not providing profitable actionable knowledge. In this paper, we present an efficient algorithm for the automatic extraction of profit-maximizing knowledge for business applications with multi-class customers by postprocessing the probability estimation decision tree (PET). When the PET predicts a customer as belonging to any of the lesser profitable classes, then, our algorithm suggests the cost-sensitive actions to change her/him to a maximum possible higher profitable status. In the proposed novel approach, the PET is represented in the compressed form as a Bit patterns matrix and the postprocessing task is performed on the bit patterns by applying the bitwise AND operations. The computational performance of the proposed method is strong due to the employment of effective data structures. Substantial experiments conducted on UCI datasets, real Mobile phone service data and other benchmark datasets demonstrate that the proposed method remarkably outperforms the state-of-the-art methods.

Keywords Data mining, Knowledge Engineering and Applications, Machine Learning: Methods and Applications, actionable knowledge discovery, profit maximization.

1 Introduction

Most of the service providing sectors such as Mobile Phone service, Internet, Banking, Insurance, IT Services, Retail, etc. are encountering the crisis due to certain classes of their customers. The prime reason for massive losses taking place in these sectors is due to the attritors, a class of customers who close their account and shift to the competitor over a period of time [1] and this phenomenon is called attrition or churning. Apart from attritors and non-attritors still, there are other classes of customers who stay with the service provider but inactive in using the services and lead to very insignificant profits and uncertainty. Most often, enterprises classify their customers based on the amount of profit gained from them over a certain period of time and perceive them in a certain order of priority based on the degree of profitability. Among these multiple classes of customers in the decreasing order

ISSN: 1137-3601 (print), 1988-3064 (on-line)

of profitability those who stand at the first tier yields highest profit and the last tier yields very less or zero profit. Most of the times last tier customers are attritors. There are many reasons for attrition or lesser profitability nature of the customers and some of them are huge industry deregulations, low service levels, high tariffs, not updating with the technology, etc. If necessary actions are taken, then, a customer who is predicted to be an attritor or of any less profitable class can be converted as a non-attritor and high profitable.

It is a known fact that retaining an existing customer is cheaper than finding a new customer [2]. However, enterprises consistently try hard to retain their customers by organizing customer retention campaigns and provide suitable offers to the likely attritors. In some businesses, they also take actions such as changing the current plan, reducing interest rates on loans etc. to convert other inactive classes of customers as active and more profitable. It is a difficult task to detect probable attritors, and less profitable classes of customers among a large number of customers through campaigns. Hence, a machine learning model for attrition or less profitability prediction is required.

Some of the researchers of the machine learning community have focused on the attrition problem and they treated the attrition prediction as a classification problem and limited only on constructing a classification model using the customers' base. Thereafter, they mainly concentrated on improving the technical interestingness measures like accuracy and AUC, etc. of the constructed model [3-8]. When an existing customer's record is given as input to the constructed model; it only classifies the customer as a probable attritor or non-attritor. This knowledge is not useful to the enterprise since it does not suggest any actions to change the customer from a less profitable class to a higher profitable one and no direct benefit is obtained. Hence, some manual work by the business expert has to be performed on the model to find the actions to change the customer from a less profitable class to a higher profitable one.

A customer's sample can be re-classified by changing the values of the required attributes. Here the idea of *changing a customer's class* means, changing her/him from a lesser or zero profitability category to a higher profitability category. Till now, very less study has been done on extracting profit maximizing knowledge automatically from the machine learning models. Though some of the past research has addressed this problem, in that they treated the problem as a 2-class problem and presented the methods to convert the customer from class C_2 (attritor) to class C_1 (non-attritor) only [9-13]. Moreover, there is no specific focus on computational performance while achieving the objective of profit maximization.

When the customers are of more than two classes then, enterprises perceive them in the decreasing order of profitability as C_1, C_2, C_3,..., C_n classes. Class C_1 customers are highest profitable and in most of the applications, class C_n customers are attritors. In the service oriented business sectors, in one set-up, customers are classified as platinum class (C_1), gold class (C_2), iron class (C_3), and lead class (C_4), where n=4, in the decreasing order of profitability [14]. In one context of the Retail industry, customers are categorized as true friends (high loyal and highest profitable), butterflies (low loyal and high profitable), barnacles (high loyal and low profitable), and strangers (low loyal and lowest profitable) [15]. In another circumstance, they are classified as stay customers, discount customers, impulsive customers, need based customers, and wandering customers or churn customers (zero profitable) [16]. In such kind of applications [17], if a customer is predicted as belonging to class C_m ($m \leq n$) then, it is necessary and beneficiary to the enterprise to re-classify her/him to another class C_k ($k<m$ and $k \geq 1$) by applying the required actions. An action is changing an attribute's value of a customer. For instance, in the wireless mobile phone sector, changing the *data plan* of a customer from one category to another is an action.

To address these limitations and challenges, we propose an efficient algorithm namely EDest_Leaf_Finder (EDLF). Our research considers the attrition avoidance and profit maximization problem as a multi-class classification problem and employs a decision tree specifically probability estimation decision tree (PET) which is built using the customers' profiles. With the help of the PET if a customer is predicted to be as an attritor or a low profitable class customer, then our method tries to convert her/him as non-attritor and higher profitable class customer by changing the values of the required attributes of the customer. However, efforts are made only to change the values of flexible attributes, whose values are possible to change (eg. Service Level, data plan, etc. in Telecom sector, discount rate, interest rate, etc. in other sectors), and not the values of the non-flexible attribute's whose values cannot be changed (eg. Gender, age, and Income of the customer, etc.).

In the process of achieving a profit maximizing solution, the constructed PET is represented in the compressed form as a Bit patterns matrix (BPM). Thereafter, PET is used only to find the class label and class probability estimation of a customer's sample. All the remaining postprocessing work is performed on the BPM only rather than on the actual form of the PET. EDLF follows a novel approach and employs bitwise AND operations on the elements of the 2-D array (BPM) for attaining the objective. Due to the effective organization of the bit patterns and efficient usage of the array data structures and bitwise AND operations, the computational achievement of the proposed method is strong. The proposed method is applicable to the business domains/applications with n number of classes of customers, where $n \geq 2$, and it is designed such that it does not leave any option and provides an optimal solution with a maximum possible net profit for a customer who is predicted as attritor/less profitable.

Even when it is not possible to re-classify the customer from class C_m to C_k (k<m), the proposed method tries to find a profit enhancing solution.

In summary, this paper focuses on providing a profit maximization solution for the service providing business sectors which classify their customers into multi-classes. When a customer is predicted as attritor/less profitable, the proposed method suggests customized actions to reclassify her/him such that the net profit obtained is the maximum. The solution is provided while achieving the remarkable computational performance by employing efficient data structures. We have illustrated the working of the proposed method using a synthetic dataset from the Banking sector. By conducting experiments on synthetic data, real-world data belonging to Mobile phone service, UCI and other benchmark datasets, the efficiency of the proposed method has been compared with a single tree based [9] and also ensemble tree based state-of-the-art methods [12,13]. These experiments demonstrate that our method achieves remarkable computational performance and outperforms the state-of-the-art methods with respect to runtimes and profits.

The rest of the paper is organized as follows: In section 2 we review the literature and discuss the related work. In section 3, we provide the preliminaries and discuss mining profitable knowledge from PET using our algorithm EDLF for 2-class, 3-class problems and finally, a mathematical model has been formulated to compute the net profit for multi-class applications. Performance evaluation of EDLF is presented in section 4 by comparing its computational times and profits with state-of-the-art methods. In Section 5, we have given the conclusions and discussed the possibilities for future work.

2 Literature Survey and Related work

Earlier, many researchers of machine learning and data mining have studied the attrition problem and handled it as a classification problem. They have built various data mining models for attrition prediction. However, till today there is more focus on building churn prediction models only rather than automatic extraction of the profitable actionable knowledge from the model. And most of this attrition prediction research has only targeted on improving the performance metrics like accuracy, AUC, and robustness, etc. of the constructed model [3-8]. Recently, Sivasankar and Vijaya have presented [18] a hybrid method for building a churn prediction model which is based on the combination of classification and clustering. They claimed that the predictive accuracy of their method is high. Bahnsen et al. [19] presented a new measure based on monetary constraints for evaluating the effectiveness of the attrition methodology. Their method makes use of several fixed offers which is also dependant on product based cost and likeliness of offer acceptance by the customer. As different classes of customers have a varied monetary effect, they projected within the customers that the charge assessed and specified by the new methodology will be different.

The method proposed by Eugen Stripling et al. [20] has taken the profit into account and hence included the profitability concept in the churn prediction model. They have introduced a genetic algorithm based classifier for churn prediction which also incorporated the notion of profit maximization. The study of Pednault et al. [21] has observed the sequential nature of some of the CRM problems and addressed the issue of sequential decision making using the Markov decision process. Yang et al. [9] presented a single decision tree based cost-sensitive profit-maximizing method viz. Leaf_Node_Search that suggests an optimal destination leaf for an instance which is predicted as belonging to an un-loyal class(attrition). Their algorithm follows a conventional tree traversal approach for searching the solution that leads to more computational time when the size of the prediction model is huge.

Instead of class probabilities, Nasrin Kalanat et al. introduced [10, 11] fuzzy based methods which makes use of fuzzy membership to mine the profitable actions from the data. Their methods build the fuzzy decision tree and postprocess the tree to mine the actions to improvise the fuzzy profit by taking fuzzy costs into account. Sebastiaan Höppner, et al. introduced a new classification approach [22] for mining the profitable knowledge for 2-class applications of attrition prevention. Their approach integrates the profit calculation measure within the decision tree construction phase itself.

To change one input sample from an undesired class to a desired one, Zhicheng Cui et al. proposed a cost-sensitive method [12] to extract actionable knowledge from additive tree models(ATM) such as Random forest. They have formulated the problem as an Integer Linear Programming (ILP) problem. Qiang LU et al. [13] also adopted ATM classifiers to address the problem of mining the actionable knowledge for achieving a maximum net profit from each individual. They have formulated the problem of extracting optimal actions from the ATM as a state space graph search problem and solved it by the state space search algorithm. Further, to reduce search time and achieve the runtime performance, they presented a sub-optimal search algorithm with an acceptable and rational heuristic function. However, ATM's are complex and lacks the interpretability. When dataset size is huge, the model built using algorithms like Random forest can be enormously huge and deep. Consequently, search time for finding an optimal solution highly increases.

Gao and Yao [23] presented a method which follows a three phase model. Trisecting is the first phase where a universe of samples is partitioned into three disjoint sections. The second phase is the acting step where distinct action plans for the three sections are discussed. In the third phase, they discussed the method of changing the samples from undesired to the desired zone. To extract actionable knowledge, Cao et al. proposed four types of generic frameworks [24, 25] which incorporate domain knowledge to some extent. They studied on providing solutions for the applications of various domains using a framework which not only considers technical interestingness but also domain-specific expectations.

Though research on actionable knowledge discovery from data mining models is limited, some researchers have even shown their interest in surveying the existing methods and reviewed the issues with the present research [26]. Zhang et al. studied the rate of prediction, prediction capability and withholding capacity [27] and proposed a method for obtaining profit. Nasrin Kalanat and Eynollah Khanjari proposed a new cost-sensitive method [28] for mining actionable knowledge from graph data belonging to social networks where there can be relationships between the objects. To the best of our knowledge, all the existing research has treated the problem as a 2-class problem only. However, some of the studies [12, 13] followed a tricky method while dealing with multi-class datasets eventually treated as a binary classification problem.

3 Automatic extraction of profit maximizing knowledge from PETs

3.1 Modelling using a PET

Automatic extraction of actionable knowledge for profit maximization in the context of business problems is considered as a classification problem. PET has been used for modelling in our research. The motivation for using decision trees is, they are simple and easy to comprehend. They also work well for high dimensional data and also accuracy is in general high. The research in this paper can also be explained at ease with the help of the decision tree. In the business sectors, customers' profiles are described by a large number of different types of attributes like income, age, gender, education and nationality, service usage time, service plan, service level, rate of interest on loans, etc. To build the decision tree using the customers' profiles, C4.5 technique [29] has been used since it is one of the best off-the-shelf classifiers. C4.5 has become highly popular after ranked as #1 algorithm in the field of data mining [30]. Normally, CRM datasets tend to be very large. When the model is built on such datasets using the other effective decision tree induction methods like Random forest, the size of the model is high and moreover, those ensemble methods produce multiple trees. Consequently, the CPU time for achieving our objective significantly increases and also the process to attain the solution turn out to be more complex.

The proposed algorithm is designed to work on a single tree. Because of all these reasons, C4.5, a benchmark algorithm has been used in our research for model construction. C4.5 uses gain ratio [29] as the splitting criterion for attribute selection at a node during the tree construction. The ratio of Information gain and Split information with respect to an attribute *A* results in gain ratio as shown in Eq. (1). Information gain, as given in Eq. (2), is the amount of information that can be gained if attribute *A* is chosen as a splitting attribute at a node. If the entropy concerning an attribute is 0 then the Information gain can be maximum. Entropy is the expected information required to classify a sample in the dataset *D* as given in Eq. (3). Information gain method favours the attribute with maximum outcomes and selects it as a splitting attribute. Gain ratio measure overcomes this drawback by applying a type of normalization to Information gain using split information. The amount of Split information with respect to an attribute *A* denotes the potential information generated by splitting the training dataset *D* into *o* number of partitions, belonging to the *o* number of outcomes of a test on attribute *A*. Computation of Split Information with respect to an attribute *A* is given in Eq. (4).

$$\text{Gain Ratio (A)} = \frac{\text{Info Gain(A)}}{\text{Split Information(A)}} \quad \text{where} \tag{1}$$

$$\text{Information Gain (A)} = \text{Entropy(D)} - \sum_{j=1}^{o}\left(\frac{|D_j|}{|D|} \times \text{Entropy}(D_j)\right), \tag{2}$$

$$\text{Entropy(D)} = \left(-\sum_{i=1}^{|C|}(P_i * \log_2 P_i)\right) \text{ and} \tag{3}$$

$$\text{Split Information}_A\text{(D)} = -\sum_{j=1}^{o}\left(\frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)\right) \tag{4}$$

During the tree construction, the attribute computed as possessing the highest gain ratio is determined as the splitting attribute at a node of consideration. In the given equations, $|C|$, *o* and $|D|$ are the number of classes, number of outcomes of an attribute *A* and total number of records in the training dataset respectively. Various steps in C4.5 algorithm for tree construction are described in the pseudocode in Algorithm 1.

The proposed algorithm though depends on the predicted class label for providing an optimal solution, it strictly makes use of the probability estimation of each class which can be offered from the decision trees. For the induction of probability estimation decision tree, the extension of C4.5 viz., C4.4 algorithm is adopted. The tree output by C4.4 estimates the probability of belonging to each class for an instance fallen into a particular leaf node. C4.4 is totally based on the frequency and adopts the maximum likelihood estimate for probability estimation [31]. Further, Laplace correction is also applied to overcome extreme probabilities scenarios. The probability that a customer's instance X which has fallen into a leaf node belongs to class C is given in Eq. (5).

$$P(C/X) = \frac{\sum_{j=1}^{|D|} \delta(R_j,\ C) + 1}{|D| + |C|} \tag{5}$$

If the j^{th} customer's record, R_j, belongs to class C, then $\delta(R_j,C)=1$, or else 0. After the PET is constructed, its effectiveness is evaluated using the required metrics like accuracy, Area Under Receiver Operator Characteristic Curve (AUC), etc.

Algorithm 1. C4.5. Construct a decision tree using the instances of data partition of D

```
Input   : An attribute-valued training data partition D associated with class labels
Output  : A decision tree

create a node N ;
if all the instances in D are of the same class, C then return N as a leaf node labeled with the class C;
if attribute_list is empty then return N as a leaf node labeled with the majority class in D;
                    // attribute_list - List of available attributes at that node or partition
Apply Attribute_selection_method, Gain ratio (D, attribute_list) to find the best Splitting_Attribute
 and label the node N with Splitting_Attribute;
if Splitting_Attribute is discrete_valued then
   attribute_list=attribute_list- Splitting_Attribute; // Splitting_Attribute is removed from the attribute_list
for each outcome j of Splitting_Attribute   // partition the instances and grow subtrees for each partition
    let $D_j$ be the set of data tuples in D satisfying outcome j;   // a partition
    if $D_j$ is empty then
          attach a leaf labeled with the majority class in D to node N;
    else attach the node returned by C4.5($D_j$, attribute list) to node N;
 end for
 return N;
```

3.2 Finding Destination leaf for profit maximization

When a customer's instance is input to the PET it reaches one of the leaf nodes which represent a class label and also the probability of belonging to each class. Based on the class label and the class probability information, the degree of customer's loyalty and the amount of profit that the service provider can make from the customer is predicted. If a customer is predicted as attritor/less profitable, then, it is required to change her/him as a non-attritor and maximum profitable. To achieve such profit maximizing solution, we introduce our algorithm EDLF. When a customer's instance X falls into a leaf node of the PET that represents attrition/less profitability then, our proposed method finds an optimal destination leaf (defined below) with a maximum net profit by applying necessary actions. Providing the optimal solution is the goal. While finding the optimal destination leaf for a customer, EDLF algorithm considers each potential leaf node and then a leaf with a maximum net profit will be chosen.

Definition 1 (Source). The leaf node of the PET in which a customer's sample X falls into based on its attributes' values is defined as a Source(L_S).
Definition 2 (Desirable leaf). Irrespective of the number of classes of customers that the dataset has, only the leaf node possessing highest class C_1 probability among all the leaf nodes is a Desirable leaf.
Definition 3 (Candidate). A leaf node which estimates a profit more than that of the L_S is a Candidate (L_C). Amount of profit obtained from a customer if she/he falls into a leaf is computed with respect to the probabilities of all classes except class C_n represented by that leaf.
Definition 4 (Action). An action Actn is defined as a change in the value of an attribute *A* which is denoted as Actn = $\{A, p \rightarrow q\}$, q=1, 2, ..., o where $q \neq p$. p and q are original and changed values of *A* respectively. o is the number of outcomes of attribute *A*. If *A* is continuous valued attribute then o=2.

Definition 5 (Action set). If an instance is required to fall from L_S into an L_C of the PET, then a set of actions are required. When the path along root to L_C of the PET is considered, it contains say *len* number of decision nodes (non-class attributes). The attribute-value pairs along the path of L_C are $\{(A_i=q_i)$, i=1,2,...,*len*$\}$. For the instance X, the corresponding attribute-value pairs are $\{(A_i=p_i)$, i=1,2,...,*len*$\}$. In the act of shifting the instance from L_S to L_C, a set of actions in the form $Ast=\{(A_i,\ p_i \rightarrow q_i)$, i=1, 2, ..., *len*$\}$, where $p_i \neq q_i$, are required.
Definition 6 (Net Profit). To make the instance fall into an L_C from the L_S, an action set is required. For each action Actn=$\{A,\ p \rightarrow q\}$ in *Ast*, some cost is incurred. For each attribute a cost matrix is maintained where the entries in them are provided by the domain expert. Total cost for all actions in *Ast*, i.e. Tot_Cost is calculated. Net profit is then computed as the difference in the profit when the instance X is in L_C and in L_S minus Tot_Cost.
Definition 7 (Destination). A leaf node in which a customer's sample has been eventually shifted to, from the L_S with a maximum Net Profit is the Destination (L_D).
Definition 8 (Goal). If the L_S is not a Desirable leaf then, the Goal is finding the action set, *Ast*, to make the instance X fall into Destination L_D and symbolized as (X, $L_S \rightsquigarrow L_D$, Max_NetProfit) where Max_NetProfit is the Net Profit obtained which is eventually a maximum value.

With our proposed method, the tree is represented as a compressed model in the form of a 2-D array. Research framework of our method is presented in Figure 1. The BPM representing PET, input instance and the output given by the PET for the instance are inputs to our proposed algorithm. The dataset is preprocessed before it is used for PET construction. Prior to applying the proposed method on the input instance, the constructed PET is evaluated using 10-fold cross validation. If the input instance does not fall into the Desirable leaf of the PET, then the proposed method determines the Destination for it. This is achieved by applying the bitwise AND operations on the attributes' values of the input instance X and the Candidate which are organised as an array data structure.

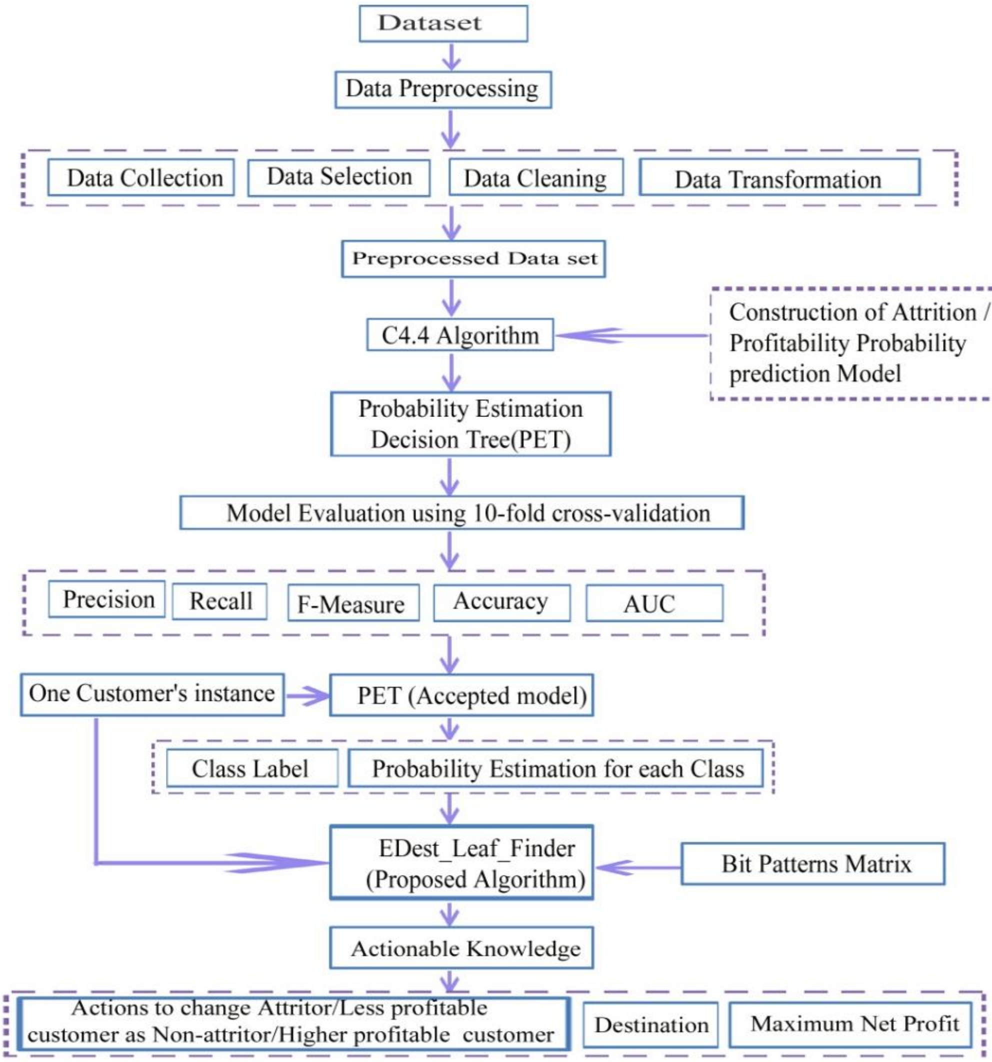

Dataset
Data Preprocessing
Data Collection
Data Selection
Data Cleaning
Data Transformation
Preprocessed Data set
C4.4 Algorithm
Construction of Attrition / Profitability Probability prediction Model
Probability Estimation Decision Tree(PET)
Model Evaluation using 10-fold cross-validation
Precision
Recall
F-Measure
Accuracy
AUC
One Customer's instance
PET (Accepted model)
Class Label
Probability Estimation for each Class
EDest_Leaf_Finder (Proposed Algorithm)
Bit Patterns Matrix
Actionable Knowledge
Actions to change Attritor/Less profitable customer as Non-attritor/Higher profitable customer
Destination
Maximum Net Profit

Figure. 1 Research framework of EDLF

3.2.1 Profit maximization for 2-class problems

There are many business sectors which generally classify their customers as Positive and Negative classes. The Positive class customers are loyal and stay with the enterprise and contribute profits. On the other hand, Negative class customers, who are familiarly called as attritors or churners, leave the enterprise and go for the service of the competitor. Banking sector is highly facing the attrition problem due to a number of options to the customer. Hence, for illustration of our proposed work, a synthetic dataset in Table 1 belonging to the Banking service sector, where each customer belongs to either of the two classes Positive/Negative is considered. In this sector, customer's profiles are included different attributes such as socio demographic (e.g., age, income, gender, job status, education and nationality, etc.), service levels provided to the customer, behavioural information (e.g., duration of service usage, amount of revenue generated, etc), etc. This information is used to predict the customer's nature.

The details of the synthetic dataset are shown in Table 2. This dataset consists of 17 records composed of four significant discrete attributes of the customers and a class label viz., Positive (C_1), Negative (C_2). Among 17 records of the dataset, 64.71% are Positive and 35.29% are Negative. When this data set is given as input to the C4.4 algorithm, the PET shown in Figure 2 is obtained as output. The induced PET depicts what sort of customer's features lead to either the class C_1 or C_2 and also the probability of belonging to each class. Thus, for a customer's instance, the PET can provide a class label and also the probability of belonging to each class. To provide more clarity, each leaf node of the PET in Figure 2 is associated with class label and also the probability of belonging to each class.

Table 1 Synthetic Bank customers dataset

RID	Service Level	Income Of Customer	Gender	Rate_Of_ Interest	Class
1	Low	High	Male	Low	Negative
2	Low	High	Male	High	Negative
3	High	High	Male	Low	Positive
4	Medium	Medium	Male	Low	Positive
5	Medium	Low	Female	Low	Positive
6	Medium	Low	Female	High	Negative
7	High	Low	Female	High	Positive
8	Low	Medium	Female	Low	Negative
9	Low	Low	Female	Low	Positive
10	Medium	High	Female	Low	Positive
11	Low	Medium	Female	High	Positive
12	High	Medium	Male	High	Positive
13	High	High	Female	Low	Positive
14	Medium	Medium	Male	High	Negative
15	Medium	Low	Male	Low	Positive
16	High	High	Male	Low	Positive
17	Low	Medium	Male	High	Negative

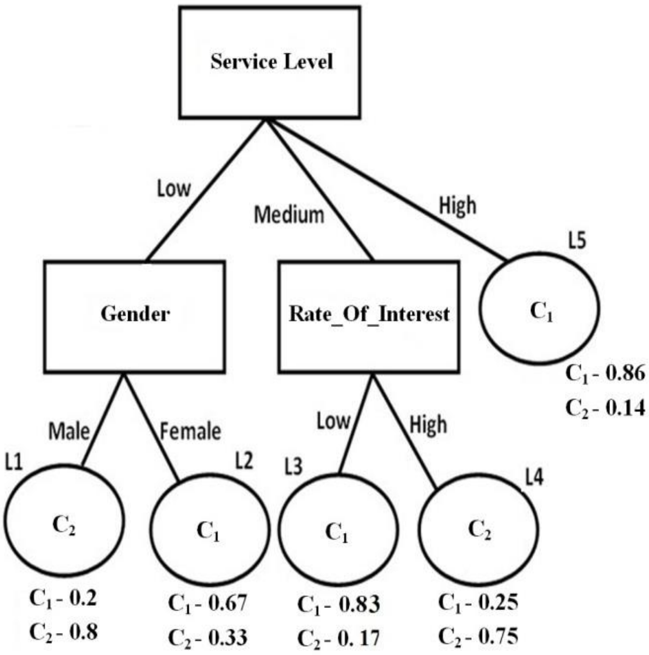


Figure. 2 PET representing the dataset in Table 1

Table 2: Features of synthetic Bank customers dataset in Table 1

Attribute	Description	Flexible/ Non-flexible	Values
Service Level	Level of service provided to the customer	Flexible	Low, Medium, High
Income Of Customer	Customer's annual income	Non-flexible	Low, Medium, High
Gender	Gender of the customer	Non-flexible	Male, Female
Rate_Of_Interest	Interest Rate on the loans taken by a customer	Flexible	Low, High
Class Label	Whether customer left (Negative) or stayed (Positive) with the Service provider	Flexible	Positive, Negative

3.2.2 Bit patterns matrix representation of the PET

EDLF is based on bitwise operations and array data structures. Normally bitwise operations are utilized and incorporated when we are required to encode or decode data in a packed manner and need the computations to be performed quickly. Bitwise AND can be used to execute set intersection and bitwise OR to execute set union. We use the former point which is being employed in our proposed method. To increase efficiency arrays are used in the course of attaining the Goal since; accessing array elements can be done in a faster way.

The BPM, B[ ][ ] is formed by *p* number of *bit patterns vector*s where *p* is the number of leaf nodes representing one of the class labels among C_1, C_2, ..., C_{n-1} and *n* is the number of customers' classes. Hence, BPM for the PET in Figure 2 is formed with the bit patterns vectors of the leaf nodes L2, L3, and L5, since, n=2. Bit patterns of the leaf nodes representing class C_2 are not necessary to furnish in the BPM since; no customer will be changed to such leaf nodes. For each of these *p* leaf nodes of the PET, the path along the root to a leaf is considered and each decision node's (non-class label attribute's) value in that path is represented in terms of the fixed size *bit pattern*. Bit patterns are formed for all the attributes present in that path and also for the attributes not present in the path but in the tree. However, the method considers only the attributes which are present in the PET and not all the attributes in the dataset. If an attribute has *v* outcomes, then that attribute's value is represented with a *v-bit pattern*. For each value of an attribute, among *v* bits of the pattern, only one bit is set to 1 and all the remaining bits are set to 0. The number of *v-bit patterns* in each row of the BPM is *q* where *q* is the number of distinct decision nodes in the PET. For the PET shown in Figure. 2, bit patterns of the attributes are formed as shown below.

Gender (Male-10, Female-01), *Service Level* (low-100, medium-010, high-001), *Rate_Of_Interest* (low-10, high- 01). If an attribute is not found in the path from root to a leaf then, its bit pattern is represented with all 1's for our computational requirements. For the PET in Figure 2, L3 is one of the leaf nodes. Fixed size bit patterns vector for the path from the root to leaf L3 is formed as: Attribute values to reach L3 shall be *Gender*=NULL, *Service Level*=medium, *Call Charges*=low. Thus, *bit patterns vector* of the path of L3 is (11, 010, 10). Similarly, bit pattern vectors of the other potential Candidates L3 and L5 are also formed.

The BPM of the PET in Figure 2, whose order is *p*×*q*, is shown in Figure 3. The bit patterns in each row of BPM are arranged in the order of *Gender*, *Service Level* and *Rate_Of_Interest*. Class label and class probabilities estimations of the rows (leaf nodes) in BPM are maintained as furnished in Table 3, which are needed during Net Profit computation. With this approach, once the PET is represented in the form of BPM, PET is used only to find the class label and class probability estimation of the input instance. The total remaining process is performed on the 2-D array formed by fixed length bit patterns vectors. However, the method of representing the PET as a BPM is the same irrespective of the number of classes of customers. After the PET is represented as a BPM, EDLF starts its post processing. It considers bit pattern vectors of all the Candidates and determines the Destination among them. In the PET in Figure 2 only the leaf node L5 is a Desirable leaf.

$$\begin{array}{rcl} & \begin{array}{ccc} \textit{Gender} & \textit{Service Level} & \textit{Rate\_Of\_Interest} \end{array} & \\ B = & \begin{pmatrix} 01 & 100 & 11 \\ 11 & 010 & 10 \\ 11 & 001 & 11 \end{pmatrix} & \begin{array}{l} \leftarrow L2 \\ \leftarrow L3 \\ \leftarrow L5 \end{array} \end{array}$$

Figure. 3 Bit patterns matrix representation of the Candidates of PET in Figure 1

Table 3: Class label and class probabilities Estimation of the leaf nodes of the PET in Figure 1

Leaf	Class	Class-C_1 probability	Class-C_2 probability
L2	C_1	0.67	0.33
L3	C_1	0.83	0.17
L5	C_1	0.86	0.14

3.2.3 Determining the Candidates and Destination for an instance in 2-class problems

In the case of 2-class problems, say for a customer's instance X Source is L_S, representing class C_k, where k=1(Positive) or k=2 (Negative), which is not a Desired leaf. Then, in these two contexts, the proposed method determines the Destination among all the probable Candidates, L, as:

$$k=1 :\ (X,\ L_S \rightsquigarrow \text{Max\_NetProfit}\ ((\forall L)\ ((L \in C_1)\ \wedge\ (Pc_1(L) > Pc_1(L_S)))))$$

$$k=2 :\ (X,\ L_S \rightsquigarrow \text{Max\_NetProfit}\ ((\forall L)\ (L \in C_1)))$$

In the above two cases, $Pc_1(L)$, $Pc_1(L_S)$ are class C_1 probabilities of the probable Candidate, L, and Source respectively. In the case when k=1, if L represents class C_1, and its C_1 class probability is more than that of the C_1 probability of L_S, then L is a Candidate. In the other case when k=2, any leaf node L representing class C_1 is a Candidate. In each of these cases, among all the Candidates one which yields maximum Net Profit is chosen as the Destination.

In the case of 2-class problems, when a customer's instance has fallen into a leaf node then the profit obtained from her/him is computed with respect to the class C_1 only. If the probability of belonging to class C_1 is 1.0 (customer is 100% of Positive class) for a leaf node, then enterprise makes a certain amount of profit P_x from the customer who has fallen into this leaf. For example, if an instance X has fallen into the leaf L2 of the PET in Figure 2, then 0.67*1000=$670 is made by the service provider when P_x value is considered as $1000. If X is shifted to a Candidate then, the profit obtained after shifting is calculated in terms of the difference of class C_1 probabilities in Source and Candidate which is the net gain in the class C_1 probability after the shifting. If X is shifted from L1 to L5 then the profit is 860-200=$660. Eventually, the Net Profit, N_P, obtained after transformation of X from L_S to an L_C is formulated in Eq. (6). By subtracting the total cost of actions incurred for this transformation, the EDLF computes the Net Profit.

$$N_P = P_x * (P_{C_1}(L_C) - P_{C_1}(L_S)) - \text{Tot\_Cost} \tag{6}$$

For illustration, a customer's sample X is considered, where the attributes' values are *Gender*=Male, *Service Level*=Low, *Rate_Of_Interest*=High. X falls into the leaf node L1 of the PET in Figure 2 which is not a Desirable leaf. Bit patterns vector representation of instance X is (10, 100, 01). Then, the Net Profits, if X is shifted to each of the Candidates, are computed and the one which produces the maximum is the Destination. For the case of X, Candidates are L2, L3, and L5.

For shifting X to a Candidate, a set of actions are required. To confirm whether an action is required or not on an attribute, a bitwise AND operation is performed between the bit patterns of X and the Candidate on the corresponding attribute's values. If the bitwise ANDing results in *False* then, it denotes that an action is required. Consequently, during finding the Destination for a customer's sample X, only the attributes, after bitwise ANDing, which are resulted as *False* are considered for further steps. Cost for this action will be obtained from the cost matrix of the corresponding attribute. *True* cases need not be considered since the result will be *True* in two cases. The first case, if a particular attribute's value is the same for the customer's instance and also the Candidate's path, then there is no need to change the value of that attribute. Second, if an attribute is not present in the path of a Candidate then, the case of changing that attribute's value does not arise. To make this case *True*, that nonpresent attribute's value's bit pattern is represented with all 1's.

In the process of attaining the Goal, bitwise AND operations are performed between X and the Candidates. The results after the bitwise AND operations are shown in Table 4 where the operations resulted in *True* and *False* are denoted as T and F respectively. During the Net Profits computation, every action cost is considered as $100 and P_x value is taken as $1000. Then, L1$\rightsquigarrow$L2: Not possible since a non-flexible attribute's (Gender) value has to be changed since 10 AND 01=*False*. L1$\rightsquigarrow$L3: Ast={(*Service Level*, Low→Medium), (*Rate_Of_Interest*, High→Low)}, N_P=1000*(0.83-0.2)-(100+100)=$430. L1$\rightsquigarrow$L5:Ast={(*Service Level*, Low→High)}, N_P=1000*(0.86-0.2) -100=$560. Goal : (X, L1$\rightsquigarrow$L5, $560). In this scenario, by shifting the customer X from L1 to L5 a maximum amount of Net Profit is obtained and no other option can improve the Net Profit more than this amount.

As the other case, for the sample Y={*Gender*=Female, *Service Level*=Low, *Rate_Of_Interest*=High}, Goal is (Y, L2$\rightsquigarrow$L5, $90) with Ast={(*Service Level*, Low→High)}. In this case, for Y, the leaf nodes L3 and L5 are the Candidates. Finally, the sample Y is shifted from a leaf node representing class C_1 to another leaf node representing class C_1 but possessing a higher probability of C_1.

Further, to achieve extra computational performance, attributes order in the bit patterns vector is formed such that in each row of BPM, bit patterns of all the non-flexible attributes are placed first followed by flexible attributes'. In the BPM shown in Figure 3, bit pattern of *Gender* is the first element in every row. This is done irrespective of the order of the attributes along the path from the root to leaves. The reason for this arrangement is for early stopping i.e. along the path of a Candidate, if any non-flexible attribute's value is required to be changed, then that Candidate can be ignored in that stage itself so that unnecessary extra computations are avoided. This situation is observed while performing L2 & X on the attribute *Gender* where we have stopped and not performed bitwise ANDing on the remaining attributes of the L2 as shown in Table 4.

Table 4: Bitwise AND (**&**) operation between instance X and Candidates

L2 & X			**L3 & X**			**L5 & X**		
10	100	01	10	100	01	10	100	01
01	100	11	11	010	10	11	001	11
F			T	F	F	T	F	T

3.3 Profit maximization for 3-class problems

In real time, several business sectors categorize their customers into three classes based on the amount of profit obtained from them within a period of time. In one scenario of retail Banking, customers are categorized as Medium-high profit customers (C_1), Low profit customers (C_2), Unprofitable customers (C_3) [17]. The amount of profit m ade from the class C_1 and class C_2 customers is high and mediocre respectively, while from class C_3 customers profit contribution is zero or negative. In the case of 3-class applications, when an instance X falls into a leaf node, then, that instance most likely contains the qualities of all the three classes but with different ranks.

In this scenario, as both the classes C_1 and C_2 are profitable, profit is calculated with respect to these two classes. For example, L2 in Figure 4 is one of the leaf nodes of the hypothetical PET representing a 3-class dataset that comprises the class probabilities C_1=0.8, C_2=0.1, C_3=0.1. This means for a customer who has fallen into L2 contains the qualities of all the three classes in different degrees. If class C_1 probability is 1.0 then the service provider earns an amount of P_{X_1} and if class C_2 probability is 1.0 then an amount of P_{X_2} is made. Let the assumption is P_{X_1}=\$1000 and P_{X_2}=\$500. An instance X is considered for illustration and assumed as fallen into the leaf node L2. If X remains in L2, then the amount of profit made is 0.8*1000 + 0.1*500 =\$850.

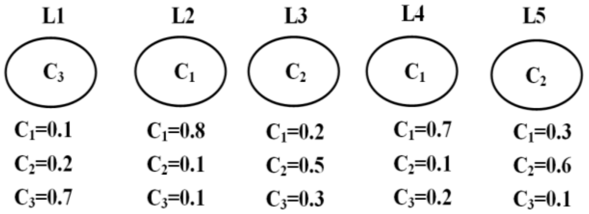


Figure. 4 Leaf nodes of a hypothetical PET

While computing the profit (Pt) resulted after migration of the instance from Source to a Candidate, sum of the differences of class C_1 probabilities in Source and Candidate and C_2 probabilities in Source and Candidate is considered as given in Eq. (7). Profit is calculated with respect to the gain in the C_1 and C_2 class probabilities. The amount of profit (Pt) and Net Profit (N_P) obtained from a customer after changing her/him from L_S to an L_C is computed as given in Eqs. (7) and (8) respectively. In Eq. (7), $Pc_2(L_C)$, $Pc_2(L_S)$ are class C_2 probabilities of the Candidate and Source respectively.

$$Pt = (P_{X_1}*(P_{C_1}(L_C) - P_{C_1}(L_S))) + (P_{X_2}*(P_{C_2}(L_C) - P_{C_2}(L_S))) \quad (7)$$

$$N_P = Pt - Tot\_Cost \quad (8)$$

3.3.1 Determining the Candidates and Destination for an instance in 3-class problems

In 3-class problems, for the customer's instance X, say the L_S is representing class C_k (k=1 or 2 or 3) which is not a Desired leaf. For each of these three scenarios, our method determines the Destination among the probable Candidates, L, as:

k=1: (i) $(X, L_S \rightsquigarrow$ Max_NetProfit $((\forall L) ((L \in C_1) \wedge (Pc_1(L) > Pc_1(L_S)))))$.

k=2: (i) $(X, L_S \rightsquigarrow$ Max_NetProfit $((\forall L) (L \in C_1)))$,

(ii) $(X, L_S \rightsquigarrow$ Max_NetProfit $((\forall L) ((L \in C_2) \wedge (Pt>0))))$.

k=3: (i) $(X, L_S \rightsquigarrow$ Max_NetProfit $((\forall L) (L \in C_1)))$,

(ii) $(X, L_S \rightsquigarrow$ Max_NetProfit $((\forall L) (L \in C_2)))$.

In the case of k=1, L_S represents class C_1 and then, another leaf representing C_1 but with a higher probability of C_1 is a Candidate. In the case of k=2, the leaf nodes representing C_1 are the Candidates of a top priority since they can provide more profit than the other options. If it is not possible to shift to any of the Candidates representing class C_1 (due to the actions on non-flexible attributes or due to the high action costs, no Candidate is yielding a positive Net Profit), then, the other option is finding the best one among the Candidates representing class C_2. In this context, the leaf node which can provide a profit (Pt), computed using Eq. (7), more than that of the Source is a Candidate. In the case of k=3, our method first tries to shift the instance X from L_S to a leaf representing class C_1. If this is not possible, then it tries to shift to a leaf representing class C_2. Hence, all the leaf nodes representing either class C_1 or class C_2 are the Candidates. A Candidate which yields maximum Net Profit is the Destination. However, the Net Profit from a Candidate also depends on the cost of actions.

For example, for a customer's instance X, Source is L1 of the hypothetical PET in Figure 4. Then, the Candidates are L2, L4, L3, and L5. P_{X_1} and P_{X_2} are considered as \$1000 and \$500 respectively. For simplicity, it has been assumed that to shift Y to any of the Candidates, it requires two actions and every action requires a cost of \$100. First priority goes to L2 and L4. L2 yields a Net Profit (1000 * (0.8-0.1) + 500 * (0.1-0.2)) – 200=\$450. L4 yields a Net Profit of \$350. Then, the Goal can be achieved by making X fall into the leaf L2, which produces a maximum Net Profit. Goal : (X, L1 $\rightsquigarrow$ L2, \$450). For this sample, since a Candidate representing class C_1 and producing some positive Net Profit is found, Net Profits w.r.t. L3 and L5 are not computed. As another example, let Y is an instance whose Source is L3. Then the Candidates for Y are L2, L4, and L5. If it is not possible to change Y to either L2 or L4 then, L5 can be the Candidate and also Destination. In this case, it is assumed that one action is needed to shift Y from L3 to L5 and hence, stood as Destination leading to the Goal (Y, L3$\rightsquigarrow$L5, \$50).

3.4 Profit maximization for n-class problems

In various business applications, there are datasets composed of multi-classes like four, five and more number of customer classes [14-16]. In CRM, in one scenario customers are classified as high stay (contributes highest level of profitability), latent stay, spurious stay, and low stay. In such applications, when there are *n* number of classes of customers ($n\geq2$) then, they are perceived as arranged in the decreasing order of degree of profitability viz., C_1, C_2, ..., C_n where C_n represents the class of attritors. In the multi-class scenario, a customer who falls into any of the leaf nodes L of the PET is most likely to have the qualities of all the n classes in different degrees. In such a case, except class C_n, all the other classes are profit yielding in different amounts. Hence, the amount of profit earned from a customer if she/he falls into a leaf L is:

$$\sum_{k=1}^{n-1} (P_{X_k} * P_{C_k}(L)) \qquad (9)$$

In Eq. (9), P_{X_k} is the amount of profit earned by the enterprise if probability of belonging to class C_k for the customer is 1.0 and P_{C_k} is the probability that the customer belongs to class C_k. Profit (Pt) obtained from a customer X by shifting her/him from a Source L_S to a Candidate L_C is computed w.r.t all the class probabilities except class C_n probability. Pt is generalized as shown in Eq. (10) and the Net Profit (N_P) is shown in Eq. (11).

$$Pt = (\sum_{k=1}^{n-1} (P_{X_k} * P_{C_k}(L_C)) - \sum_{k=1}^{n-1} (P_{X_k} * P_{C_k}(L_S)))$$

$$Pt = \sum_{k=1}^{n-1} (P_{X_k} * (P_{C_k}(L_C) - P_{C_k}(L_S))) \qquad (10)$$

$$N_P = Pt - Tot\_Cost \qquad (11)$$

3.4.1 Determining the Candidates and Destination for an instance in n-class problems

In the case of n-class problems, for a customer X if the L_S is not a Desirable leaf representing class C_k, where k=1, 2, .., n then, for each of these n scenarios, the proposed method searches and finds the Destination among all the probable Candidates, L, as:

k=1 : (i) (X, $L_S \rightsquigarrow$ Max_NetProfit ((∀L) (($L \in C_1$) ∧ ($Pc_1(L) > Pc_1(L_S)$))))
⋮
k=m : (i) (X, $L_S \rightsquigarrow$ Max_NetProfit ((∀L) ($L \in C_1$))),
(ii) (X, $L_S \rightsquigarrow$ Max_NetProfit ((∀L) ($L \in C_2$))),
⋮
(m) (X, $L_S \rightsquigarrow$ Max_NetProfit((∀L)(($L \in C_m$) ∧ (Pt>0)))).
⋮
k=n : (i) (X, $L_S \rightsquigarrow$ Max_NetProfit ((∀L) ($L \in C_1$))),
(ii) (X, $L_S \rightsquigarrow$ Max_NetProfit ((∀L) ($L \in C_2$))),
⋮
(n-1) (X, $L_S \rightsquigarrow$ Max_NetProfit ((∀L) ($L \in C_{n-1}$))).

In the case of k=m (m>1 and m<n), L_S represents class C_m and then all the leaf nodes representing the classes C_1 through C_{m-1} are the Candidates. Then, EDLF tries to find a Destination by giving priority to the Candidates in the order C_1, C_2,..., C_{m-1}. If any Candidate of the higher profitable class provides some positive Net Profit then the algorithm stops and does not go for the Candidates of the other lower profitable classes. When even the Candidates of the class C_{m-1} cannot yield a positive Net Profit, then, the leaves representing class C_m which can give the profit more than that of the L_S are the Candidates. In the case of k=n, the leaves representing any of the classes C_1, C_2, ..., C_{n-1} are the Candidates. If none of the Candidates representing class C_i (i=1, 2, ..., n-2) can yield a positive Net Profit, then only the algorithm tries a Candidate representing class C_{i+1}.

To achieve much better computational performance, the bit patterns vectors in the matrix are arranged in the order of class labels of the leaf nodes starting from class C_1 and ending with C_{n-1} so that unwanted Candidates need not be processed. Further, we don't consider arranging all the rows representing one class C_k (k=1, 2, ..., n-1) of B[ ][ ] in the descending order of the values of class C_1 probabilities. The reason is that there is no guarantee that a Candidate possessing higher class C_1 probability can yield maximum Net Profit among all the Candidates of that class. This is because; Net Profit also depends on the number of actions and cost of actions. Hence, all the Candidates representing one class are required to be processed to find the optimal one. For illustration, we have presented Table 5 which represents the organization of Bit patterns vectors of 7 leaf nodes of an imaginary PET in the order C_1, C_2, and C_3 where n=4. Among 10 leaves, 3 are representing class C_4. Hence, the BPM contains 7 rows. PET contains 5 input attributes A1, A2, A3, A4, and A5. Out of them A2 and A4 are non-flexible and remaining are flexible. It can be observed that patterns within one class are not arranged based on class C_1 probabilities. Bit patterns of the non-flexible attributes A2 and A4 occupied the first two positions in each row.

As discussed in the above sections, for finding the Destination and computing the Net Profit from one customer in the multi-class scenario, a method is devised and presented in algorithm 2. We have designed the algorithm such that it handles all the three different cases where the Source represents one among them i.e. 1. $L_S \in C_1$ 2. $L_S \in C_m$, and 3. $L_S \in C_n$. The overall working of the algorithm is briefly described in the four steps below:

1. Initialize Max_NetProfit , c, Dest **(line 1).** c is the class of a row (leaf) of BPM represented for a path from the root to a leaf which obviously starts from C_1. Therefore c=1.
2. Case 1 : **k = 1** i.e. $\mathbf{L_S \rightsquigarrow C_1}$. If the Source ($L_S$) represents class C_1, then, find Destination among the leaf nodes having better C_1 class probability than that of the L_S **(lines 2-15).**
3. Case 2 : **k < n** i.e. $\mathbf{L_S \rightsquigarrow C_m}$. If the L_S represents class C_m where m>1 and m<n, then, find Destination from the Candidates of class C_1, if not possible from the Candidates of class C_2, ..., and if not possible from the Candidates of class C_{m-1} **(lines 17-33)** and if not possible from the Candidates of class C_m **(lines 34-47).**
4. Case 3 : **k = n** i.e. $\mathbf{L_S \rightsquigarrow C_n}$. If the L_S represents class C_n then, find Destination from the Candidates of class C_1, if not possible from the Candidates of class C_2, ..., and if not possible from the Candidates of class C_{n-1} **(lines 49-68).**

The methods Profit( ) and Tot_Cost( ) computes profit when X is shifted from L_S to L_C and total cost incurred for actions respectively. These methods are presented in algorithm 3 and algorithm 4.

Table 5: Organization of BPM of an imaginary PET representing 4-class data

S. No.	Leaf No.	Class	Class C_1 Probability	Class C_2 Probability	Class C_3 Probability	Class C_4 Probability	BPM				
							A2	A4	A1	A3	A5
1	L1	C_1	0.90	0.04	0.03	0.03	**10**	**111**	**100**	**111**	**111**
2	L5	C_1	0.70	0.10	0.10	0.10	**11**	**111**	**010**	**100**	**111**
3	L8	C_1	0.80	0.05	0.10	0.05	**11**	**100**	**001**	**111**	**111**
4	L2	C_2	0.05	0.75	0.05	0.15	**01**	**111**	**100**	**111**	**100**
5	L9	C_2	0.10	0.80	0.05	0.05	**11**	**010**	**001**	**111**	**111**
6	L3	C_3	0.20	0.05	0.70	0.05	**01**	**111**	**100**	**111**	**010**
7	L6	C_3	0.05	0.05	0.85	0.05	**11**	**111**	**010**	**010**	**111**

Algorithm 2: ***EDest_Leaf_Finder (EDLF)***

Inputs : - One customer's instance X
- Output given by PET for input instance X i.e. Source L_S information for X in PET including **class k** and each class probabilities
- Bit patterns vectors Matrix B representing PET
- n (number of customers' classes)
- Cost matrices for all the flexible attributes in the PET
- P_{X_k} for all k values.

Output : - Destination
- Net Profit

```
Max_NetProfit ← 0,   c ← 1, Dest ← $L_S$ ;    // c = 1. Rows in B obviously starts with class $C_1$, Dest - Destination for X.
if  k = 1   then                          // $L_S \in C_1$
    for  i ← 1  to  p  do        // p is number of  rows in B
        if  $L_i \notin C_1$   then    // If the class represented by Leaf $L_i$ (given in B[ ][ ]'s i^th row) is not $C_1$
            break;
        else
            if  $Pc_1(L_i) > Pc_1(L_S)$    then
                $N_{Pi}$ ← Profit($L_S$,  $L_i$) - Tot_Cost(X, $L_i$);    //  $N_{Pi}$ - Net profit w.r.t. Candidate-i
                if  $N_{Pi}$ > Max_NetProfit  then
                    Max_NetProfit ← $N_{Pi}$ ;
                    Dest ← $L_i$;
                endif
            endif
        endif
    endfor
else
    if  k < n   then                          // $L_S \in C_m$
        for  i ← 1  to  p  do
            if  c < k   then
                if  $L_i \in C_c$   then        // If the class represented by Leaf $L_i$ is c. c is initially 1 since, classes start from $C_1$.
                    $N_{Pi}$ ←  Profit($L_S$,  $L_i$) - Tot_Cost(X, $L_i$);    //  $N_{Pi}$ - Net profit w.r.t. Candidate-i
                    if  $N_{Pi}$ > Max_NetProfit   then
                        Max_NetProfit ← $N_{Pi}$ ;
                        Dest ← $L_i$ ;
                    endif
                else
                    if  Max_NetProfit > 0   then      // Eventually if some positive profit got with class $C_1$.
                        break;
                    else
                        c ← c + 1;
                        i  ← i - 1;
                    endif
                endif
            else
                if  c = k   then           // If  $L_i \in C_m$ and  $L_S \in C_m$
                    Pt = Profit($L_S$,  $L_i$);
                    if  Pt > 0   then
                        $N_{Pi}$ ←  Pt - Tot_Cost(X, $L_i$);
                        if  $N_{Pi}$ > Max_NetProfit    then
                            Max_NetProfit ← $N_{Pi}$ ;
                            Dest ← $L_i$;
                        endif
                    endif
                else
                    break;
                endif
            endif
        endfor
    else                                // $L_S \in C_n$
        for  i ←1  to  p  do
            if  c < n   then      //  If the class represented by Leaf $L_i$ is less than $n$  i.e. $C_n$
                if  $L_i \in C_c$   then
                    $N_{Pi}$ ←  Profit($L_S$,  $L_i$) - Tot_Cost(X, $L_i$);
                    if  $N_{Pi}$ > Max_NetProfit   then
                        Max_NetProfit ← $N_{Pi}$ ;
                        Dest ← $L_i$;
                    endif
                else
                    if  Max_NetProfit > 0   then
                        break;
                    else
                        c ← c+1;
                        i  ← i-1;
                    endif
                endif
            endif
        endfor
    endif
endif
Output (Dest, Max_NetProfit) ;
```

Algorithm 3: Profit calculation for multi-class problems

```
Profit ($L_S$,  $L_C$)
  begin
    Pt ← 0;
    for  k ←1  to  n-1  do
      Pt ←  Pt + ( $P_{X_k}$ *  ($P_{C_k}$($L_C$)  -  $P_{C_k}$($L_S$)) )
    end for
    return (Pt);
  end
```

Algorithm 4: Calculation of total cost of actions for Source to Candidate transformation

```
Tot_Cost(X, $L_i$ )
    begin
        for  j ← 1  to  q   do     // q is the number of  columns in the Matrix B
            T_Cost ← 0;        //  T_Cost - Total cost  of actions
            if (B(i, j) & $X_j$ )  is  False  then   // Bitwise AND operation between $i^{th}$ Candidate's $j^{th}$ attribute's values of Bit pattern vector and X
                if  attr[j]  is a  non-flexible attribute  then    //  attr[j] – $j^{th}$ attribute
                    skip this Candidate-i and continue with next one   // X can't be shifted to Candidate-i
                else
                    T_Cost ←  T_Cost + attr[j]_Cost_matrix(corresponding value) ;
                endif
            endif
        end for
        return(T_Cost);
    end
```

4 Experimental Evaluation

In order to evaluate the performance of the proposed algorithm, extensive experiments are conducted using several multi-class UCI datasets, real Telecom dataset, and other benchmark datasets. We have compared the performance of the proposed method with a single crisp tree based method, and two ensemble tree based methods. The proposed method exhibited a similar performance in all the experiments and outperformed the state-of-the-art methods.

4.1 Performance analysis of EDLF

The strength of EDLF algorithm lies in the effective usage of two vibrant points which are suitable for the research discussed in this paper. First is, it completely counts on bitwise AND operations. In computer programming, a bitwise operation is performed in between bit patterns at the level of their individual bits. For performing comparisons and other calculations efficiently, bitwise operations are far preferable as they straight away executed by the processor. After compilation of any bitwise operation, irrespective of the programming language used for implementation, it is translated as a single assembly language instruction in all most all assembly languages though that assembly language is overly simplified. When that assembly instruction turn comes for execution, it takes only one CPU cycle for evaluation. Thus the bitwise operations are basic and quick operations which can most significantly enhance the computation performance. Second is, performing the bitwise operations on the elements of a 2-dimensional array. Using the array data structure for storing and accessing a group of elements can greatly enhance the computational performance since an array is an efficient data structure for this purpose. In the context of accessing the elements, arrays are simple and efficient data structures than any other data structures like linked lists, etc. Moreover, accessing any single element in an array is done in a faster manner and in the constant time O(1).

To compute the Net Profit obtained from a customer X it is necessary to process the elements of the BPM which contains *p* rows and *q* columns. For complexity analysis, we consider that on average, the number of rows and the number of columns in BPM is *p*. In the case of successful search (Destination found), best case runtimes can be observed if any of the Candidates of class C_1 yields a positive Net Profit for the input instance X. In this case, the remaining Candidates of the classes C_2, C_3, ..., C_{n-1} need not be processed and the algorithm's time complexity is O(1). If L_S for an instance X represents class C_n and if the Destination is found among the

Candidates of class C_{n-1} only and if most of the attributes in each row of BPM are flexible and their values are required to be changed then, the proposed algorithm exhibits worst case runtime i.e. O(p^2). For successful search, average case time complexity is also O(p^2).

In the case of unsuccessful search (Destination not found), if each row's first element represents a non-flexible attribute and if that attribute's value has to be changed, then the algorithm exhibits the best case with runtime O(p). Another best case scenario for unsuccessful search arises when L_S represents class C_1 and if the Destination is not found among the Candidates of class C_1. If L_S represents class C_n and if no Candidates of the classes C_1 through C_{n-1} yields a positive Net Profit, and if majority number of attributes in each row of BPM are flexible and their values are required to be changed then the proposed algorithm exhibits worst case runtime O(p^2). During the unsuccessful search, average case time complexity when L_S represents class C_n is O(p^2) and if L_S represents class C_1 it is O(1). However, in real life, time complexities in the order of quadratic are acceptable and the customer's class changing problem can be solvable in a finite amount of time.

4.2 Experimental setup

To evaluate the efficiency of our algorithm, datasets which are appropriate to the research discussed in this paper are used. To construct a PET with C4.4 algorithm, Weka (version of 3.8) source code in Java, has been used. All numeric attributes in the datasets used in the experiments are discretized (except to the datasets in section 4.4) by applying the supervised filter viz., *Discretize* in Weka which is based on the class information, via MDL method [32]. The constructed PET's performance is evaluated using 10-fold cross-validation and the precision, recall, F-measure, accuracy and also AUC to figure out the probability estimation capability [33] are recorded for each dataset. Then, the proposed method, EDLF, is implemented in Java programming language and the experiments are conducted on a dual core Pentium 4, 2.5 GHz processor with 4GB RAM running on Windows7 Operating System.

We have used the UCI and other standard datasets to demonstrate the efficiency of the proposed method. However, we have incorporated the required features to these datasets such that they fit to demonstrate our research. During the experiments on UCI and other benchmark datasets, when working on 2-class datasets, one appropriate class has been considered as Positive (C_1) and the other one as Negative (C_2). Then, while computing the Net Profit, P_x value is set to \$1000. While applying the proposed method on datasets composed of multi-classes (n>2), we assumed and treated the classes aptly in the descending order of profit and computed the Net Profit. If the dataset is composed of n-classes viz., C_1, C_2, ..., C_k, ..., C_n, and if L_S represents class C_k ($k \geq 1$ and $k<n$), then a profit value of \$(1000/k) has been considered. However, for all datasets, if L_S purely represents class C_n, then a profit value of \$0 has been considered during Net Profit calculations. Action costs are taken appropriately for the flexible attributes in the range \$[0-200]. Based on their characteristics, some of the attribute are considered as non-flexible.

4.3 Comparison with single crisp tree based method

We first compare the computational performance of the proposed method with Leaf_Node_Search [9]. To our knowledge, this is the state-of-the-art method based on a single crisp tree and best fits for comparison with our method. Though Leaf_Node_Search algorithm is designed to work for 2-class problems, we have made required modifications to it such that it works for multi-class datasets and conducted experiments on UCI datasets, and a real Telecom dataset.

4.3.1 Experiments on UCI ML Datasets

For comparing runtimes and Net Profits of EDLF with Leaf_Node_Search, experiments are performed on 10 UCI datasets [34] with multi-classes that supports classification task and have sufficient examples. The details of the datasets used in the experiments along with the information of the PET constructed using each dataset and the PETs' technical evaluation measures are furnished in Table 6. The experimental setup which is discussed in section 4.2 is employed. The methods are implemented in the Java programming language.

Table 6: UCI datasets used in the experiments and the evaluation metrics of the PETs

Data set	No. of Instances	No. of Attributes	No. of Classes	Precision	Recall	F-measure	Accuracy (%)	AUC
Anneal	898	39	6	0.924	0.923	0.920	92.31	0.961
Autos	205	26	7	0.846	0.839	0.839	83.90	0.940
Balance Scale	625	4	3	0.658	0.696	0.673	69.60	0.768
Connect-4	67557	42	3	0.791	0.808	0.797	80.84	0.868
German	1000	20	2	0.706	0.721	0.710	72.10	0.696
Glass	214	10	7	0.743	0.738	0.732	73.83	0.852
Heart-c	303	14	5	0.791	0.789	0.787	78.87	0.801
Hypothyroid	3772	30	4	0.994	0.995	0.994	99.46	0.990
Nursery	12960	8	5	0.970	0.971	0.970	97.05	0.995
Solar	1066	12	6	0.723	0.745	0.728	74.48	0.916

The two methods try to find the Destination. Each time a set of instances are taken randomly from one dataset and given as input to the algorithms and the total runtime for finding the Destination for each of those instances, and total Net Profit obtained from all those set of instances are recorded. The graphs presented in Figure 5(a) through 5(j) describe the runtime behavior of the proposed method and Leaf_Node_Search on 10 UCI datasets. In all the graphs, x-axis and y-axis represent the number of instances taken as input from one dataset and the total runtime respectively. All the graphs shown in Figure 5(a) through 5(j) describes the fact that on all datasets, runtime of EDLF is significantly less as compared with Leaf_Node_Search whose runtime is increasing exponentially. Leaf_Node_Search allows a large number of candidate actions to be considered, complicating the computation. The computational complexity of Leaf_Node_Search is high since it is necessary to perform more number of primitive operations(comparisons). If the average number of Candidates, average number of attributes, and average length of path from the root to a Candidate is considered as *p* then the runtime of Leaf_Node_Search is O(p^3). The time complexity of Leaf_Node_Search is cubic whereas the time complexity of the proposed method is polynomial with degree 2.

When the training data size is huge, the constructed PET's size can be large and deep. Then, the average length of each path from the root to a Candidate and the number of Candidates increases. When the tree size increases, then there can be a possibility for the increase in the number of actions. This eventually increases the computational time since the number of bit patterns in each row of the Bit pattern matrix gets increased and more bitwise operations need to be performed for finding the Destination. In the same scenario, Leaf_Node_Search requires higher run times than EDLF since it has to identify the attribute and then its value. However, Net Profit also decreases in this case as the Tot_Cost value increases. Hence, while working on large datasets with too many number of attributes, runtimes of both the methods increase. This fact is observed with the Connect-4 dataset, where the constructed tree contains 15952 nodes. However, this increase is high with Leaf_Node_Search. When more number of non-flexible attributes are encountered along a path, then, there is a possibility for early stopping of the processing which causes a reduction in the runtime of EDLF. The potential use of bitwise operations, best use of the array data structures and placing non-flexible attributes' bit patterns as the starting elements in each row of BPM are substantially helping improve the efficiency of EDLF.

Table 7 presents the Net Profits produced by the Leaf_Node_Search and EDLF on each of the 10 UCI datasets used in the experiments. On all the 10 UCI datasets, our algorithm is generating a higher total Net Profit in comparison with Leaf_Node_Search. Leaf_Node_Search produced a total Net Profit of $10695525 and on the other hand, EDLF produced $12000760 which is 12.2% more than that of the Net Profit produced by the Leaf_Node_Search. Leaf_Node_Search only tries to change an instance from a leaf node with class C_k to C_j (j<k) and stops if these transformations are not possible. On the other hand, in the same scenario, the proposed method tries to shift the instance from Source representing class C_k to a Candidate representing class C_k that can yield some positive Net Profit. Though a Candidate contains the higher probability of class C_1, along its path from the root, if it contains more non-flexible attributes, then there is a chance for discarding that Candidate. If more Candidates are of such kind, then, eventually there is a chance for the drop in the Net Profit. In the other case, if more the number of actions required for shifting an instance from L_S to L_C then more the chances for an increase in total cost that leads to reduction in Net Profit.

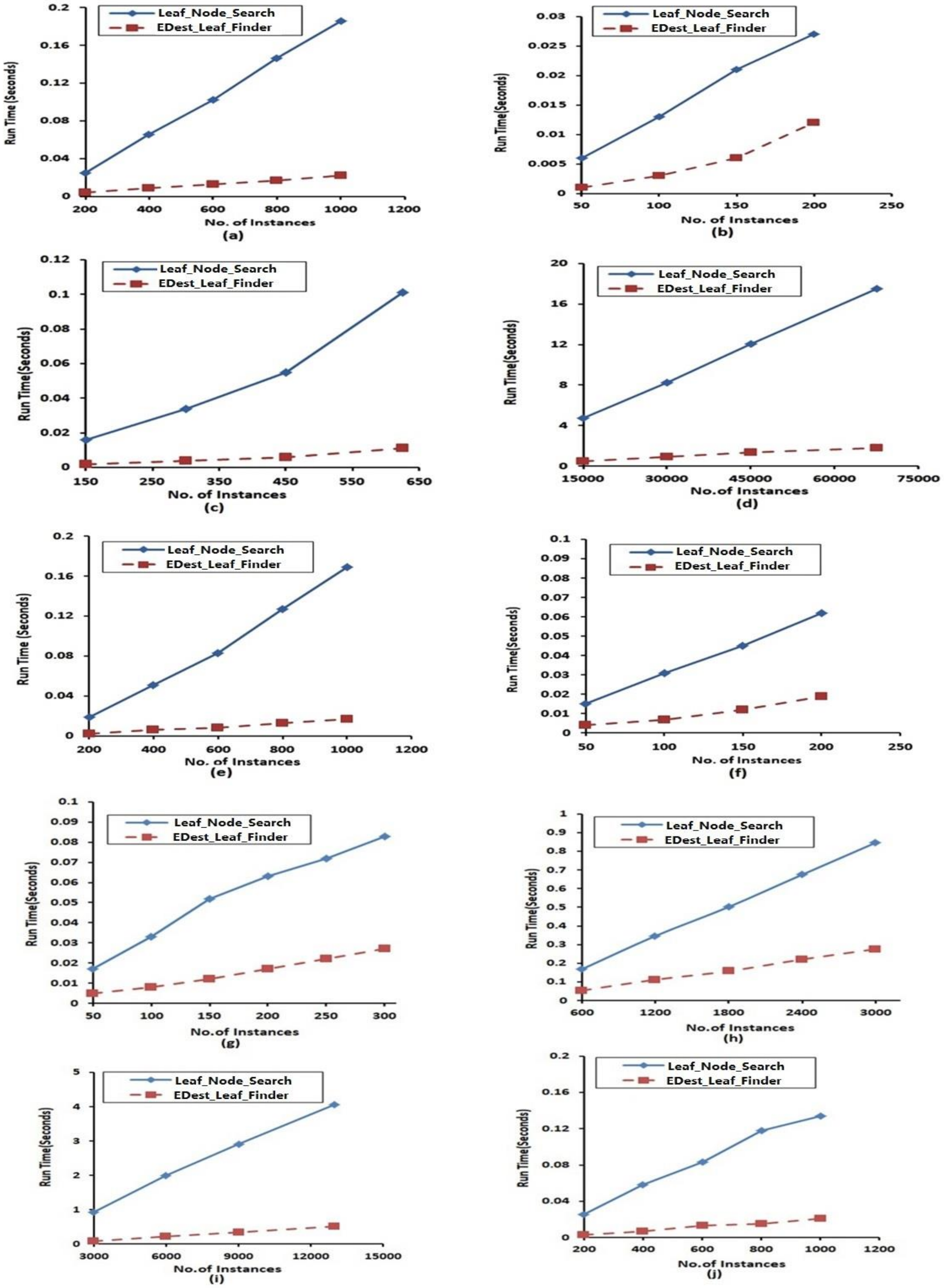


Figure. 5 Runtime comparisons of EDLF and Leaf_Node_Search on UCI datasets.
(a) Anneal (b) Autos (c) Balance Scale (d) Connect-4 (e) German (f) Glass (g) Heart-c (h) Hypothyroid (i) Nursery (j) Solar

Table 7: Net profits comparison of EDLF and Leaf_Node_Search on UCI datasets

Name of the Dataset	Net Profit($)	
	Leaf_Node_ Search	EDLF
Anneal	140610	153070
Autos	53400	67860
Balance Scale	105490	117330
Connect-4	8903200	9981700
German	169025	191080
Glass	57820	67070
Heart-c	49300	56440
Hypothyroid	228650	247620
Nursery	877580	995520
Solar	110450	123070
Total	10695525	12000760

4.3.2 Experiments on real data

Mobile phone service sector is facing more problems due to attrition which is leading to huge losses. Hence, we have considered and performed experiments on the large real time Telecom dataset obtained from an operator in India and demonstrated the performance of EDLF against the Leaf_Node_Search. This dataset has 15000 records of the customers where 2500 customers are classified as Negative and 12500 are as Positive. Each customer's instance is described by 40 input attributes which are divided into four categories viz., Socio-demographic (age, Gender, job status, and education etc.), behavioural (calling behaviour, short message usage, data usage, and frequency of calls made to customer care center, etc.), tariffs levied (Call Charges, Data Charges, etc.), and Service Levels (Customer complaint resolving, internet speed, network coverage, and call drop rate reduction, etc.). Among the 40 attributes, 19 are flexible and the remaining 21 are non-flexible. The decision attribute is a class label that denotes whether the subscriber has retained or left the service provider within a period of 3 months.

To maintain the balance in the distribution of class labels of the subscribers' records in the dataset and to avoid predicting many examples as Positive, we have randomly chosen 2500 Positive samples from the dataset. Totally 5000 records are used in the experiments where the share of the Positive and Negative samples is equal. By giving these 5000 samples as input to C4.4 algorithm, a PET is constructed which contains 734 leaves. Among these leaves, 346 represent the class Positive and the rest of the 388 have Negative class label. However, we have also applied the required data pre-processing methods (eg. data cleaning, data transformation, etc.) before building the tree. 10-fold cross validation is used to evaluate the induced PET where the accuracy and AUC values are observed as 88.04% and 0.894 respectively. During profit calculations, action costs are considered in the range $[0-200] depending on the attribute. P_x value is considered as $1000. After this setup, we have applied our EDLF and also Leaf_Node_Search to compare their runtimes and Net Profits.

Table 8: Runtime and Net Profit comparisons of EDLF and Leaf_Node_Search on real Telecom dataset

#Records	Runtime(Seconds)		Net Profit($)	
	Leaf_Node_ Search	EDLF	Leaf_Node_ Search	EDLF
1000	0.2501	0.04659	104380	115090
2000	0.4113	0.08432	195330	218110
3000	0.6397	0.11457	256020	278430
4000	0.8511	0.15039	350790	387020
5000	1.2637	0.21231	432880	466580

Table 8 presents the runtimes and Net Profits comparison of EDLF and Leaf_Node_Search on the real Telecom dataset. Each time, a set of samples are given as input to the algorithms and the total runtime to determine Destination for each of those instances and the total Net Profit obtained from the set of samples are recorded. It has been noted that proposed method is notably outperforming Leaf_Node_Search in all cases. On average, the runtime for finding the optimal solution for one instance with Leaf_Node_Search is 0.0002269 seconds and with EDLF is 0.0000414 seconds which is 5.48 times less than that of the former method. For all cases, the total Net Profit produced by EDLF is 9.39% more than that of generated by the Leaf_Node_Search. In the case when 5000 records are given as input, the total Net Profit generated by EDLF is 7.78% more than that of produced by the Leaf_Node_Search. On average, Net Profit yielded for one instance by Leaf_Node_Search is \$92.33 and by the EDLF is \$101.40 which is 9.82% more than that of the one produced by Leaf_Node_Search.

When a minimal number of actions are required to be performed and those actions are not required on non-flexible attributes, then, high Net Profit values can be expected. In the cases where there is no possibility to shift instances from a leaf node with class C_k to another one with class C_j (j<k) EDLF generates more Net Profit than that of generated by the Leaf_Node_Search.

4.4 Comparison with ensemble tree based state-of-the-art methods

Computation times of proposed method are compared against two ensemble tree based state-of-the-art methods i.e. Suboptimal search [13] and Integer Linear Programming (ILP) [12] and the results are presented in Table 9. To the best of our knowledge, these two are the leading state-of-the-art methods for optimal action mining.

These state-of-the-art techniques find a set of actions that can convert the input sample from an undesired status to the desired one using additive tree Model (ATM) which are based on an ensemble of trees. With the same parameter setting, experiments are performed on nine datasets from LibSVM[1] website and UCI which are used in Suboptimal search and ILP's original experimental analysis. For runtime comparison, from each dataset 30 samples are randomly picked and given as input and the average computation time to provide the solution for each of these 30 samples is recorded. The main motive of our EDLF is to search and find a Destination among all the Candidates. Hence, no other leaf node produces profit more than that of the one produced as output. Therefore, we focus on comparing the methods with respect to the computation times. These ensemble tree based methods also provide the solution by treating the problem as 2-class. However, while processing the multi-class datasets, these techniques follow a tricky method by using the one-verses-remaining all strategy which eventually perceives the problem as 2-class. They set the desired class as Positive and all the remaining classes as Negative.

Table 9: Runtime comparison of proposed method and two state-of-the art methods on nine benchmark datasets

Dataset	#Instances	# Attributes	# Classes	Accuracy (%)	AUC	Runtime (Seconds)			$\frac{M3}{M2}$ (%)	$\frac{M3}{M1}$(%)
						ILP (M1)	Suboptimal search (M2)	EDLF (M3)		
A1a	32561	123	2	84.4	0.85	7.54	4.03	7530×10^{-5}	998.67×10^{-5}	1868.48×10^{-5}
Australian	690	14	2	85.6	0.89	108.03	1.87	95×10^{-5}	0.88×10^{-5}	50.80×10^{-5}
Breast cancer	683	10	2	95.0	0.95	31.01	1.46	89×10^{-5}	2.87×10^{-5}	60.96×10^{-5}
DNA	3386	180	3	92.8	0.94	35.39	4.43	6478×10^{-5}	183.05×10^{-5}	1462.3×10^{-5}
Heart	270	13	2	81.8	0.84	5.71	2.33	300×10^{-5}	52.54×10^{-5}	128.7×10^{-5}
Ionosphere	351	34	2	89.2	0.92	48.92	2.91	237×10^{-5}	4.84×10^{-5}	81.44×10^{-5}
Liver disorders	345	6	2	63.2	0.56	31.71	0.17	2.8×10^{-5}	0.088×10^{-5}	16.47×10^{-5}
Mushrooms	8124	22	2	100.0	1.00	3.69	2.71	4835×10^{-5}	1310.3×10^{-5}	1784.1×10^{-5}
Vowel	990	12	11	81.5	0.95	68.15	1.99	185×10^{-5}	2.71×10^{-5}	92.97×10^{-5}
				85.9 (Average)	0.88 (Average)	340.15 (Total)	21.9 (Total)	0.195668 (Total)	284×10^{-5} (Average)	616.24×10^{-5} (Average)

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets// datasets/

On average, the computational time of EDLF is 0.284×10^{-2}% of ILP and 0.61624×10^{-2}% of Suboptimal search. In most of the experiments, EDLF's computational times are drastically less than those of ILP and Suboptimal search. This fact is observed especially on the datasets Liver disorders (0.000088×10^{-2}% and 0.01647×10^{-2}%), Australian (0.00088×10^{-2}% and 0.0508× 10^{-2}%), Breast cancer (0.00287×10^{-2}% and 0.06096×10^{-2}%).

When the dataset size and dimensionality increase, the computational time of EDLF has increased to some extent and still very less than those of ILP and Suboptimal search. The runtimes on the datasets A1a (0.99867 × 10^{-2}% and 1.86848×10^{-2}%), DNA(0.18305×10^{-2}% and 1.4623×10^{-2}%) and Mushrooms(1.3103× 10^{-2}% and 1.7841×10^{-2}%) depict this observation. As the Suboptimal search and ILP applies their postprocessing method on an ensemble of trees built using Random forest classifier, they require more computation time to achieve the objective. On the other hand, the proposed method postprocesses and extracts knowledge from a single tree.

From all the experimental results, it can be concluded that the proposed method outperforms two different classes of state-of-the-art methods. However, all the other methods talk about the conversion of an instance from class C_2 to C_1 only and treat as a 2-class problem. Proposed EDLF tries for more optimization for acquiring maximum profits and fits well for multi-class problems.

5 Conclusions and Future Scope

The ultimate goal of the enterprises is to improve their profits. In this regard, they depend on the machine learning models for acquiring automated profit maximizing knowledge. In brief, in this paper, we have strictly focused on two issues. The first one is, providing a profit maximization solution for enterprises with multi-class customers who are facing losses and uncertainty due to attritors and other insignificant profitable classes of customers. Second, providing the solution while achieving remarkable computational efficiency. Our novel algorithm EDLF serves both the purposes. On the other hand, the limited research done in the past has treated this profit maximization problem as a 2-class problem and also there was no focus on computational achievement. We started our work by building a probability estimation decision tree (PET) using customers' profiles. Then, the PET is represented as a compressed form as a BPM on which the entire profit maximization course of action is performed. Customers are supposed to be in the decreasing order of profitability viz., C_1, C_2, ..., C_n. If a customer falls into any leaf nodes of the PET which is not a Desirable leaf then, the proposed method searches and finds another optimal leaf which provides a maximum Net Profit. Our method suggests the cost-sensitive actions, to shift the customer from Source to Destination, such that the solution is most optimum. When it is not possible to migrate a customer from a leaf node representing class C_k to other leaf node representing class C_j (j=1,2,.., k-1), then, the proposed method tries to shift the instance to a leaf node representing class C_k that can provide more profit than that of the Source. Moreover, the proposed method provides tailored profit maximizing actions for each individual customer. Afterwards, we have discussed the solution for a 2-class problem with the help of a synthetic Telecom dataset. Then, for the 3-class scenario, we have discussed computing the Net Profit with an example. Finally, we have generalized and formulated a mathematical model to provide a solution for n-class applications where $n \geq 2$.

Due to the merits like the potential use of array data structures, the best use of bitwise operations on the fixed length bit patterns, and the appropriate arrangement of the bit patterns and also the bit pattern vectors of the leaf nodes in a well-organized order, the computational achievement of EDLF is very impressive. The experiments performed on UCI datasets, real Telecom dataset, and other benchmark datasets prove that the proposed EDLF algorithm appreciably outperforms the other classes of state-of-the-art methods with respect to computational time and Net Profits.

As the next step, a study can be performed to improve the proposed algorithm such that the new approach can also find the class label/source leaf node information of the given input sample by using the BPM so that the induced PET can never be used in the postprocessing phase. This work can also be extended by incorporating ontological concepts and can be used for various other domains. We believe the proposed method can evolve as a remarkable method for profitable action extraction for the service providing sectors like Telecom, Banking, Internet, Retail, and online shopping, etc.

Acknowledgments

Sincere thanks are given to the anonymous reviewers for their valuable comments and suggestions.

References

[1] W. Kamakura, C. F. Mela, A. Ansari, A. Bodapati, P. Fader, R. Iyengar, R. Wilcox. Choice models and customer relationship management. Marketing Letters, 2005, 16(3–4): 279–291. http://dx.doi.org/10.1007/s11002-005-5892-2

[2] F. Reichheld. Prescription for cutting costs. Bain & Company, 2014

[3] Long Bing Cao, D. Luo and C. Zhang. Knowledge actionability: Satisfying technical and business interestingness. International Journal of Business Intelligence and Data Mining, 2007, 2(4):496-514

[4] G. Nie, W. Rowe, L. Zhang, Y. Tian and Y. Shi. Credit card churn forecasting by logistic regression and decision tree. Expert Syst Appl., 2011, 38:15273–15285. doi: 10.1016/j.eswa.2011.06.028

[5] Adnan Amin, Sajid Anwar, Awais Adnan, Muhammad Nawaz, Khalid Alawfi, Amir Hussain, Kaizhu Huang. Customer Churn Prediction in Telecommunication Sector using Rough Set Approach. Neurocomputing, 2017, 237:242–254. DOI: http://dx.doi.org/10.1016/j.neucom.2016.12.009

[6] Abbas Keramati, Hajar Ghaneei and Seyed Mohammad Mirmohammadi, Keramati. Developing a prediction model for customer churn from electronic banking services using data mining. Financial Innovation, 2016, 2:10. DOI 10.1186/s40854-016-0029-6

[7] Patrik Berger, Michal Kompan. User Modeling for Churn Prediction in E-Commerce. Intelligent Systems, IEEE, 2019, 34(2):44-52 March-April DOI: 10.1109/MIS.2019.2895788

[8] Thomas Verbraken, Wouter Verbeke and Bart Baesens. Profit optimizing customer churn prediction with Bayesian network classifiers. Intelligent Data Analysis, 2014, 18(1) : 3-24. DOI 10.3233/IDA-130625

[9] Quiang Yang, Jie Yin, Charles Ling and Rong Pan. Extracting Actionable knowledge using decision Trees. IEEE Transactions on Knowledge and Data Engineering, 2007,17(1) : 43-56. doi: 10.1109/TKDE.2007.9

[10] Nasrin Kalanat, Behrouz Minaei-Bidgoli. An optimized fuzzy method for finding action. Journal of Intelligent & Fuzzy Systems, 2016, 30(1):257-265. DOI:10.3233/IFS-151751

[11] Nasrin Kalanat, Shamsinejadbabaki, P. Saraee. A fuzzy method for discovering cost-effective actions from data. Journal of Intelligent & Fuzzy System, 2015, 28: 757–765. DOI:10.3233/IFS-141357.

[12] Zhicheng Cui, Wenlin Chen, Yujie He, Yixin Chen. Optimal Action Extraction for Random forests and Boosted Trees, In:Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015,179-188

[13] L. U. Qiang, Zhicheng Cui, Yixin Chen, Xiaoping Chen. Extracting optimal actionable plans from additive tree models. Frontiers of Computer Science, 1-14, 11(1), 2017, Pages 160-173. https://doi.org/10.1007/s11704-016-5273-4

[14] V. Zeithaml, R. T. Rust, K. N. Lemon. The customer pyramid Creating and serving profitable customers. California Management Review, 2001, 43(4), 118–142

[15] Reinartz, Werner, V. Kumar. The Mismanagement of Customer Loyalty. Harvard Business Review, 2002, 80 (7): 86-94, 125

[16] A. S. Dick, K. Basu. Customer loyalty: Toward an integrated conceptual framework. Journal of the Academy of Marketing Science, 1994, 22(2):99-113. https://doi.org/10.1177/0092070394222001

[17] R. Garland. Non-financial drivers of customer profitability in personal retail banking. Journal of Targeting, Measurement and Analysis for Marketing, 2002, Vol. 10:233, pp. 233-248. https://doi.org/10.1057/palgrave.jt.5740049

[18] E. Sivasankar, J. Vijaya. Hybrid PPFCM-ANN model:an efficient system for customer churn prediction through probabilistic possibilistic fuzzy clustering and artificial neural network. Neural Computing and Applications, May 2018, doi:10.1007/s00521-018-3548-4

[19] Alejandro Correa Bahnsen, Djamila Aouada, Björn Ottersten. A novel cost-sensitive framework for customer churn predictive modelling, Springer, Decision Analytics, 2015, 2(5). DOI 10.1186/s40165-015-0014-6

[20] E. Stripling, S. Vanden Broucke, K. Antonio, B. Baesens, M. Snoeck. Profit maximizing logistic model for customer churn prediction using genetic algorithms. Swarm and Evolutionary Computation, 2018, Vol. 40, Pages: 116-130. doi: 10.1016/j.swevo.2017.10.010

[21] E. Pednault, N. Abe, B. Zadrozny. Sequential Cost-Sensitive Decision Making with Reinforcement Learning. In: Proceedings of Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, 259-268, doi:10.1145/775047.775086

[22] H. Sebastiaan, E. Stripling, B. Baesens, S. Broucke, T. Verdonck. Profit driven decision trees for churn prediction. Computational Intelligence & Information Management, 2018. https://doi.org/10.1016/j.ejor.2018.11.072

[23] C. Gao, Y. Yao. Actionable Strategies in Three-way Decisions. Knowledge-Based Systems, 133(C), 141-155, 2017, doi: 10.1016/j.knosys.2017.07.001

[24] Long Bing Cao, Y. C. Zhao, H. F. Zhang, D. Luo, C. Q. Zhang, E. K. Park. Flexible frameworks for actionable knowledge discovery. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(9): 1299–1312. DOI: 10.1109/TKDE.2009.143

[25] Long Bing Cao. Domain Driven Data Mining, IEEE Transactions on Knowledge and Data Engineering, 2010, 22(6),755-769. DOI: 10.1109/TKDE.2010.32

[26] Long Bing Cao. Actionable knowledge discovery and delivery. WIREs Data Mining and Knowledge Discovery, 2012, 2:149-163. doi:10.1002/widm.1044

[27] Z. Zhaojing, W. Regina, Z. Weihong, L. Shizhan, L. Dong, and J. Hao. Profit Maximization Analysis Based on Data Mining and the Exponential Retention Model Assumption with Respect to Customer Churn Problems. In: IEEE International Conference on Data Mining Workshop (ICDMW), 2015, 1093-1097. DOI: 10.1109/ICDMW.2015.84

[28] Nasrin Kalanat, K. Eynollah. Action extraction from social networks. Journal of Intelligent Information Systems, 2019, https://doi.org/10.1007/s10844-019-00551-2

[29] Tom Mitchell. Machine Learning and Data Mining. Communications ACM, vol. 42, no.11, Nov. 1999, 30-36.

[30] W. Xindong, J. Vipin Kumar, R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, N. Angus, Bing Liu, S. Philip, Y. Z. Zhou, S. Michael, D. J. Hand, D. Steinberg. Top 10 algorithms in data mining. Knowledge and Information Systems, 2008,14: 1–37, DOI 10.1007/s10115-007-0114-2

[31] F. J. Provost, P. Domingos. Tree induction for probability-based ranking. Machine Learning, 2003, 52(3):199–215. https://doi.org/10.1023/A:1024099825458

[32] Fayyad, M. Usama, M. Irani, B. Keki. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, .hdl:2014/35171 Proceedings of the International Joint Conference on Uncertainty in AI, (Q334 .I571 1993),1022-1027

[33] J. Huang, C. X. Ling. Using AUC and Accuracy in Evaluating Learning Algorithms. IEEE Trans on Knowledge and Data Engineering, 2005, 17(3), 299-310. DOI**:** 10.1109/TKDE.2005.50

[34] C. L. Blake, C. J. Merz. UCI Repository of Machine Learning, www.ics.uci.edu/~mlearn/mlrepository.html, 1998

[35] Q. Yang, J. Yin, X. Ling, T. Chen. Postprocessing Decision Trees to Extract Actionable Knowledge. In:Proceedings of the Third IEEE International Conference on Data Mining IEEE, 2003, 685–688. DOI**:** 10.1109/ICDM.2003.1251008

Inteligencia Artificial, 22(64) (2019), 85-101
doi: 10.4114/intartif.vol22iss64pp85-101

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Uma Nova Meta-heurística Adaptativa Baseada em Vetor de Avaliações para Otimização de Portfólios de Investimentos

A New Adaptive Meta-Heuristic Based on a Vector Evaluated Approach for Portfolio Investiment Optimization

Letícia de Fátima Corrêa Costa[1], Omar Andres Carmona Cortes[2], João Pedro Augusto Costa[1]

[1]Programa de Pós-Graduação em Engenharia da Computação e Sistemas (PECS)
Universidade Estadual do Maranhão (UEMA)
São Luis, MA, Brasil
leticiadefsc@gmail.com

[2]Departamento de Computação (DComp)
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão (IFMA)
São Luis, MA, Brasil
omar@ifma.edu.br

Abstract This article describes a new adaptive metaheuristic based on a vector evaluated approach for solving multiobjective problems. We called our proposed algorithm Vector Evaluated Meta-Heuristic. Its main idea is to evolve two populations independently, exchanging information between them, *i.e.*, the first population evolves according to the best individual of the second population and vice-versa. The choice of which algorithm will be executed on each generation is carried out stochastically among three evolutionary algorithms well known in the literature: PSO, DE, ABC. In order to evaluate the results, we used an established metric in multiobjective evolutionary algorithms called hypervolume. Tests have shown that the adaptive metaheuristic reaches the best hyper-volumes in three of ZDT benchmarks functions and, also, in two portfolios of a real-world problem called portfolio investment optimization. The results show that our algorithm improved the Pareto curve when compared to the hypervolumes of each heuristic separately.

Resumo Este artigo descreve uma nova meta-heurística adaptativa baseada em vetor de avaliações para solucionar problemas multiobjetivos. A ideia do algoritmo é evoluir duas populações independentes que trocam informações entre si, ou seja, a evolução de uma população se dá com base no melhor indivíduo da outra população e vice-versa, sendo que cada população pode ainda escolher, também de forma independente, qual meta-heurística (PSO, ED e ABC) utilizar durante sua execução. Para avaliar os resultados utiliza-se o hiper volume que é uma métrica comum em algoritmos evolutivos multiobjetivos. Testes demonstraram que a nova meta-heurística pode encontrar os melhores hiper volumes tanto nas funções ZDT de benchmarks quanto na otimização de portfólios de investimentos. Além disso, a nova meta-heurística adaptativa consegue gerar melhores fronteiras de Pareto do que meta-heurísticas de avaliação vetorial estáticas.

Keywords: Metaheuristics, multiobjective, vector evaluated, ABC, PSO, DE, portfolio optimization.

Palavras-Chave: Meta-heurísticas, multiobjetivo, avaliação vetorial, ABC, PSO, ED, otimização de portfólios.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

1 Introdução

Usualmente, problemas do mundo real são naturalmente multiobjetivos, sendo formados por dois ou mais funções objetivo que conflitam entre si, ou seja, considerando duas funções quaisquer, a melhora em uma função implica na deterioração da outra. Para solucionar esse tipo de problema, métodos tradicionais, como por exemplo, os métodos baseados em gradiente, não são adequados [5], pois: *(i)* as funções precisam ser diferenciáveis, *(ii)* o espaço de busca é limitado a poucas variáveis; e *(iii)* os métodos clássicos tendem a não tratar as restrições. Nesse contexto, surgem os algoritmos evolutivos e de enxame (MOEA) como uma alternativa à resolução de problemas multiobjetivos (MOP), pois são nitidamente reconhecidos como adequados para esse tipo de problema, pois possuem a habilidade de encontrar várias soluções não dominadas em uma única execução [29].

A primeira aparição de um MOEA ocorreu em 1985 [23], quando David Shcaffer propôs o *Vector Evaluated Genetic Algorithm* (VEGA). O principal problema do VEGA é que ele tende a perder rapidamente a diversidade, pois pode haver a escolha de um super indivíduos nas operações genéticas. Dessa forma, não há uma boa distribuição das soluções sobre a curva de Pareto. Em 1993, Fonseca e Flemming [10] proporam um algoritmo multiobjetivo chamado *Multi-Objective Genetic Algorithm* (MOGA), que foi o primeiro a considerar a dominância entre as soluções. Desde então, diversos algoritmos, tais como, NSGA-II [6], SPEA2 [31], VEPSO [21], VEABC [19], VEDE [20], VEPBIL [3], dentre outros, têm sido propostos. Este trabalho tem um particular interesse nos algoritmos baseados em avaliação vetorial como o VEPSO, VEABC e VEDE. Como o VEPBIL foi proposto para problemas discretos o mesmo não é adequado para problemas contínuos. O interesse nos algoritmos VE se dá pelas seguintes razões:

i. Os algoritmos VE são adequados para problemas cuja quantidade de objetivos é par, sendo que muitos problemas do mundo real são baseados em dois objetivos.

ii. São fáceis de implementar se comparados a algoritmos MO tradicionais, pois cada função é tratada de forma simples por cada subpopulação.

iii. Por ser baseada em subpopulações, o algoritmo é facilmente paralelizável em diferentes modelos de programação paralela

Com base nas razões apresentadas, este trabalho propõe uma meta-heurística adaptativa baseada em VE para problemas multiobjetivos. Particularmente, três algoritmos foram implementados para formar a meta-heurística: evolução diferencial (DE) [25], otimização por nuvem de partículas (PSO) [15] e colônia de abelhas artificias (ABC) [13]. O objetivo é que cada subpopulação utilize uma dessas meta-heurísticas em um dos objetivos, sendo que a escolha de qual delas utilizar é feita de forma adaptativa estocástica, ou seja, as probabilidades de escolha de cada meta-heurística vão sendo alteradas à medida que melhores soluções vão sendo geradas. A ideia é que a melhor meta-heurística seja escolhida em tempo de execução gerando soluções de melhor qualidade sem a necessidade de ter que executar cada combinação de meta-heurística previamente para determinar qual a melhor delas. Além disso, a escolha em tempo de execução pode permitir solucionar uma gama maior de problemas do que as meta-heurísticas sem adaptação.

A adaptação estocástica tem sido usada com sucesso, como por exemplo, no trabalho de Ribeiro Jr. [2], no qual escolhe-se em tempo de execução quais operadores de cruzamento e quais de mutação são usados em um Algoritmo Genético (GA). Já em Borges et. al [1] recomenda-se estocasticamente qual meta-heurística utilizar dentre GA, PSO e DE em problemas com objetivo único. Já em aplicações multiobjetivo, Costa [4] escolhe estocasticamente entre GA, PSO e DE para fazer a atualização da população no SPEA2 em busca de novas soluções não dominadas. Nesse âmbito, este trabalho faz a adaptação escolhendo qual meta-heurística utilizar em cada população independente e formando a fronteira de Pareto de forma cooperativa. Os detalhes de seu funcionamento são dados nas próximas seções.

Nesse contexto, este artigo está dividido da seguinte maneira: a Seção 2 apresenta os conceitos básico de otimização em problemas multiobjetivos; a Seção 3 introduz os conceitos das meta-heurísticas usadas em cada subpopulação; a Seção 4 detalha o algoritmo proposto neste trabalho; a Seção 5 trata dos experimentos realizados e resultados encontrados; finalmente, a Seção 6 apresenta as conclusões deste trabalho e direcionamentos para futuros trabalhos.

2 Problemas Multiobjetivos

Um problema de otimização multiobjetivo (MOP) possui dois ou mais objetivos que devem ser otimizados, sendo que esses objetivos devem ser conflitantes, isto é, a melhora de um objetivo significa necessariamente a piora em outro. Dessa forma, um MOP com restrições de domínio tem a forma apresentada na Equação 1, na qual m é a quantidade de funções, i representa a quantidade de variáveis do vetor x, e L_i, L_s são respectivamente o limite inferior e o limite superior de cada variável.

$$\begin{aligned} Min \quad ou \quad Max\ f_m(x), m = 1,2,\cdots,M \\ L_i \leq x_i \leq L_s, 1,2,\cdots,D \end{aligned} \tag{1}$$

Diferentemente dos problemas de otimização com objetivo único, onde apenas o espaço de estado (ou espaço de busca) interessa, nos MOPs é necessário mapear o espaço de estados (representado por $\overrightarrow{x}$) para um espaço de objetivos representado por $f_m(x)$ como mostra a Figura 1, que apresenta o mapeamento de 7 (sete) soluções no espaço de estados para o espaço de funções, considerando uma dimensão tridimensional para $\overrightarrow{x}$ e duas funções objetivo.

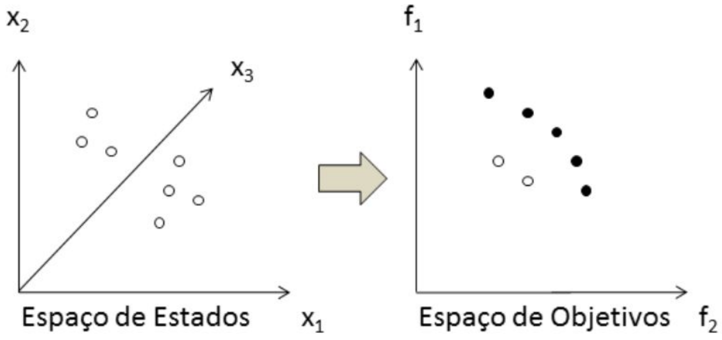


Figura 1: Mapeamento entre espaço de estados e espaço de objetivos. Adaptado de [5].

Além do mapeamento, a Figura 1 mostra um conceito importante na resolução de MOPs usado em diversos algoritmos evolutivos e de enxame, que é a dominância. No espaço de objetivos as 5 soluções em preto representam as soluções não dominadas, enquanto que as mais claras são soluções dominadas. O conjunto de soluções não dominadas é um grupo de soluções na qual nenhuma solução é melhor do que a outra, formando o que se chama de *fronteira de Pareto*. Assim, uma solução x_k domina uma solução x_i se: *(i)* x_k não é pior do que x_i em todas as funções objetivo; e, *(ii)* a solução x_k é estritamente melhor do que x_i em pelo menos um objetivo. Matematicamente diz-se que:

$$x^k \preceq x^i$$

. As próximas seções apresentam os problemas usados neste trabalho: benchmarks e o problema de otimização de portfólios de investimentos.

2.1 Benchmarks ZDT

As funções de *benchmark*, conhecidas por ZDTs, estão formalmente apresentadas na Tabela 1. As ZDTs são funções de minimização específicas utilizadas para avaliar a qualidade das soluções obtidas por algoritmos multiobjetivos [9]. Na Tabela 1 estão também as configurações para as funções de avaliação f_1 e f_2, o domínio do problema e a quantidade de genes que devem ser utilizadas para execução de cada teste. Para avaliar o resultado utiliza-se o conjunto de soluções obtidas nos testes e compara-se com a fronteira de Pareto. Assim, quanto mais próximo o conjunto de soluções estiver da fronteira de Pareto real, melhor é o conjunto de soluções.

Tabela 1: Funções ZDT's

Nome	Funções	Domínio
ZDT1	$\vec{f}: \Re^{30} \longrightarrow \Re^2$, onde $f_1 = X_1$ $g(x) = 1+(9/n-1)*(\sum_{i=2}^{k} x_i)$ $f_2 = g(x)*[1-\sqrt{x_1/g(x)}]$	$x_i \in [0,1]$ $i = 1,\ldots,30$
ZDT2	$\vec{f}: \Re^{30} \longrightarrow \Re^2$, onde $f_1 = X_1$ $g(x) = 1+(9/n-1)*(\sum_{i=2}^{k} x_i)$ $f_2 = g(x)*[1-(x_1/g(x))^2]$	$x_i \in [0,1]$ $i = 1,\ldots,30$
ZDT3	$\vec{f}: \Re^{30} \longrightarrow \Re^2$, onde $f_1 = X_1$ $g(x) = 1+(9/n-1)*(\sum_{i=2}^{k} x_i)$ $f_2 = g(x)*[1-\sqrt{x_1/g(x)}-x_1/g(x)*\sin(10\pi x_1)]$	$x_i \in [0,1]$ $i = 1,\ldots,30$
ZDT4	$\vec{f}: \Re^{10} \longrightarrow \Re^2$, onde $f_1 = X_1$ $g(x) = 1+10*(n-1)+(\sum_{i=2}^{k}(x_i{}^2-10*\cos(4\pi x_i))$ $f_2 = g(x)*[1-\sqrt{x_1/g(x)}]$	$x_1 \in [0,1]$, $x_i \in [-5,5]$,$\forall i \neq 1$ $i = 1,\ldots,10$
ZDT6	$\vec{f}: \Re^{10} \longrightarrow \Re^2$, onde $f_1 = 1-e^{-4x_1}*\sin^6(6\pi x_1)$ $g(x) = 1+9*[\sum_{i=2}^{n} x_i/(n-1)]^{0.25}$ $f_2 = g(x)*[1-(f_1/g(x))^2]$	$x_i \in [0,1]$ $i = 1,\ldots,10$

2.2 O Problema da Otimização de Portfólios

O modelo clássico de Markowitz [16] [1] para o problema de seleção de portfólio assume que os investidores preferem maximizar o retorno dentro de um certo nível de risco ou minimizar o risco dentro de um certo nível de retorno. Este modelo é conhecido como modelo média-variância por utilizar estas estatísticas sobre o histórico dos preços normalizados para calcular, respectivamente, o retorno e o risco esperados para o portfólio [26].

O risco é avaliado como a variância do retorno do portfólio e depende não só em como muitos ativos individuais variam o retorno, mas também como eles variam em relação aos outros. Sendo assim, a matriz de covariância da junção da distribuição dos retornos é um adicional para o valor esperado [27].

Portanto, um portfólio financeiro otimizará dois objetivos conflitantes, ou seja, a maximização do retorno do portfólio esperado e a minimização da variância do retorno do portfólio [27] (minimização do risco). Formalmente é descrito como mostrado na Equação 2, na qual N é o número de ativos no portfólio, isto é, a dimensionalidade do problema de otimização; w_i é o peso do i-ésimo ativo a ser otimizado; σ^2 é o desvio do risco do portfólio, enquanto σ_{ij} é a covariância entre o ativo i e o ativo j. Se i for igual a j, σ_{ij} representa a variância de um particular ativo, r_p é a média de retorno do portfólio e r_i é a média de determinado ativo i [26].

$$\begin{aligned} \text{Min } \sigma^2 &= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \\ \text{Max } r_p &= \sum_{i=1}^{N} w_i r_i \\ \text{Sujeito a } \sum_{i=1}^{N} w_i = 1,\ w_i \geq 0,\ i = 1,\ldots,N \end{aligned} \tag{2}$$

[1] Markowitz foi reconhecido em 1990, com o Prêmio Nobel de Ciências Econômicas pela sua contribuição na teoria de seleção de portfólios. [17]

A fronteira eficiente de Markowitz correspondem aos portfólios que apresentam o maior retorno esperado para determinado nível de risco. A Figura 2 apresenta a fronteira eficiente. Nesta figura os portfólios ‘A’, ‘B’ e ‘C’ são considerados eficientes. Já o portfólio ’D’ é considerado ineficiente, pois o portfólio B apresenta o mesmo risco, mas com maior retorno. Para este exemplo, um investidor não deverá escolher a carteira ‘D’, pois existe uma segunda alternativa com uma relação risco-retorno mais atrativa, ou seja, para um mesmo nível de risco da carteira ‘D’ existe a carteira ‘B’ que apresenta o melhor retorno esperado. Resumindo, os investidores devem considerar que um dos princípios para avaliar um investimento é o tipo de risco associado e o retorno que pode oferecer. Nota-se também que a fronteira de Pareto é um subconjunto da fronteira de Markowitz, pois na fronteira de Pareto D é considerado como um ponto dominado pelos demais portfólios.

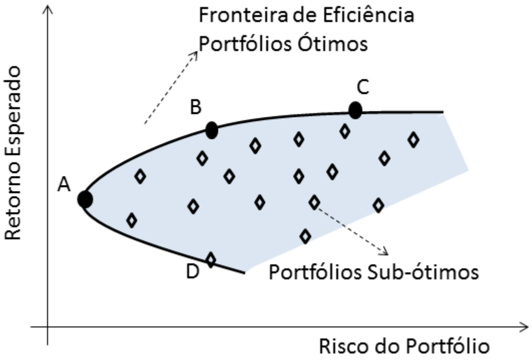


Figura 2: Fronteira Eficiente de Markowitz.

3 Meta-heurísticas

Meta-heurísticas são algoritmos que coordenam procedimentos de busca global e local com estratégias para escapar de mínimos locais em espaços de busca com soluções complexas [11], devendo ser utilizadas em problemas sobre os quais se tem pouca informação e/ou é muito custoso computacionalmente utilizar algoritmos enumerativos, ou seja, não é possível testar todas as possibilidades. A seguir são detalhadas as meta-heurísticas utilizadas neste trabalho.

3.1 Otimização por Nuvem de Partículas

Os pesquisadores Kennedy e Eberhart [15] desenvolveram a otimização por nuvem de partículas, em inglês chamada de *Particle Swarm Optimization* (PSO) [15]. O algoritmo PSO foi criado segundo a metodologia de sistemas sociais baseando-se no comportamento coletivo de indivíduos que interagem entre si e com o ambiente em torno de um ponto contendo comida ou local para descanso. Mais precisamente, é um algoritmo de enxame baseado no comportamento de como os pássaros buscam alimentos. Este tipo de sistema é conhecido também como inteligência de enxame. O Algoritmo 1 apresenta o PSO em pseudocódigo.

A inicialização das partículas (X_i) se dá de forma aleatória dentro do domínio das dimensões do problema. A velocidade das partículas (V_i) pode ser tanto inicializada por um valor aleatório quanto por um valor fixo. Após calcular o *fitness* de todo o enxame, atualiza-se a posição global (g), que é a posição da melhor solução encontrada (o *fitness* de g é chamado de *gbest*), e o histórico de cada partícula (P), que na inicialização vai conter os mesmos valores de inicialização e de avaliação do enxame. Ao entrar no laço a velocidade das partículas é atualizada a partir da equação 3, na qual w é uma constante de inércia, c_1 e c_2 são constantes aceleradoras, r_1 e r_2 são números aleatórios no intervalo $[0,1]$, g é a posição da melhor partícula (melhor solução) e p_k é o melhor valor encontrado pela partícula k.

$$v_{k+1} = wv_k + c_1 * r_1(p_k - x_k) + c_2 * r_2(g - x_k) \tag{3}$$

Em seguida, a posição da partícula é atualizada pela equação 4, ou seja, a nova posição é computada pela posição atual mais a velocidade recém calculada.

Algorithm 1 - Particle Swarm Otimization

```
Require: Inicializar partículas e velocidades;
Avaliar o enxame;
Atualizar global e histórico;
repeat
    Calcular a velocidade das partículas;
    Atualizar as posições das partículas;
    if posição atual melhor que histórico then
        Atualizar histórico;
    end if
    if posição atual melhor que global then
        Atualizar global;
    end if
until o critério de parada ser atingido.
```

$$x_{k+1} = x_k + v_{k+1} \tag{4}$$

Se x_{k+1} ultrapassar o domínio, o mesmo pode ser redefinido como o limite máximo ou mínimo que foi ultrapassado. Deve-se observar também que a velocidade pode também ser limitada no intervalo $[v_{min}, v_{max}]$. Essa restrição evita que haja uma explosão de velocidade nas partículas, o que geraria problemas tanto de exploração quanto na busca local do algoritmo.

3.2 Colônia Artificial de Abelhas

O algoritmo *Artifical Bee Colony* (ABC) foi proposto por Karaboga e Basturk [13], sendo baseado no comportamento das abelhas na busca por alimentos. No seu comportamento real, o principal objetivo das abelhas é aumentar o estoque de alimentos na colmeia, por isso as abelhas estão sempre em busca de fontes de alimentos para serem exploradas. Nesse contexto, existem três possibilidades de ações para as abelhas: *(i)* buscar novas fontes de alimento; *(ii)* explorar fontes já conhecidas; e, *(iii)* abandonar fontes de alimentação esgotadas. Como consequência das atividades, existem três tipos de abelhas: operárias, observadoras e exploradoras.

No algoritmo as abelhas operárias têm as coordenadas de uma fonte de alimento, ou seja, cada operária é responsável por uma solução. As operárias por sua vez compartilham suas informações com as abelhas observadoras, que tem por objetivo buscar novas fontes de alimento em regiões próximas às fontes atuais. As abelhas cuja fonte de alimento já se esgotou são transformadas em exploradoras e iniciam sua busca por novas fontes de alimento de forma aleatória [14]. Em termos computacionais, uma fonte de alimento representa uma solução possível para o problema, sendo que a qualidade do néctar que ali se encontra corresponde à qualidade da solução, ou seja, corresponde ao seu *fitness* ou aptidão. O algoritmo ABC executa os passos apresentados no Algoritmo 2:

Algorithm 2 - Artificial Bee Colony.

```
Require: Inicializar parâmetros da população;
repeat
    Posicione a operária em uma fonte de alimento;
    Posicione as observadoras em uma fonte de alimento;
    Envie as exploradoras na busca por novas fontes;
    Atualizar a melhor fonte de comida encontrada até o momento;
until alcançar o critério de parada.
```

Em cada ciclo ou iteração, a busca consiste em três passos: enviar a operária a uma fonte de alimento e medir a qualidade do néctar; selecionar quais fontes serão utilizadas pelas observadoras e determinar a quantidade de néctar; determinar quais abelhas serão transformadas em exploradoras e enviá-las para novas possíveis fontes. Quando o algoritmo inicia, determina-se para as operárias um conjunto aleatório de fontes de alimento e quanto néctar cada um possui. Esse processo equivale a enviar as operárias para a fonte de alimento, que quando retornam compartilham suas informações com as observadoras. As operárias retornam para a fonte de alimento conhecida e visitam novas fontes de alimento tomando como base sua informação visual, ou seja, na vizinhança da solução. As observadoras irão escolher a sua região com base na qualidade do néctar compartilhado pelas operárias, isto é, quanto maior a quantidade de néctar, maior será a possibilidade da abelha observadora escolher essa fonte. Chegando a fonte estas também escolhem novas fontes de alimentação com base em sua vizinhança. Quando o néctar de uma fonte de alimento é abandonado, uma abelha exploradora escolhe uma nova fonte de alimento de forma aleatória. No modelo proposto [13] e [14], no máximo uma abelha exploradora é enviada por ciclo. Além disso, a quantidade de abelhas operárias e de observadores é o mesmo.

Com relação ao alcance visual das abelhas, uma solução é originalmente alterada utilizando-se a Equação 5, na qual k é um número aleatório entre 1 e a quantidade de operárias, j é um número aleatório entre 1 e a respectiva dimensão do problema, sendo j necessariamente diferente de i, e ϕ_{ij} é um número aleatório no intervalo $[-1,1]$.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{5}$$

Zhu e Kwong [30], inspirados pelo PSO, proporam uma melhoria como mostrado na equação 6, na qual o novo termo é chamado de *gbest*; y_i é a melhor solução do enxame e φ é um número aleatório no domínio $[0,C]$, sendo C uma constante não negativa. Independentemente da equação sendo utilizada, se a qualidade do novo néctar é melhor que o anterior, então a fonte anterior é abandonada.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(y_i - x_{ij}) \tag{6}$$

As abelhas observadoras escolhem quais fontes visitar por meio da probabilidade fornecida pela Equação 7, na qual f representa o *fitness* de cada abelha e S é a quantidade de fontes de alimento (que é igual a quantidade de abelhas operárias). Após escolhida a fonte de alimento, a abelha observadora produz uma modificação na solução, e se a qualidade do néctar da nova solução for melhor, a observadora abandona a fonte anterior.

$$p_i = \frac{f_i}{\sum_{n=1}^{S} f_n} \tag{7}$$

Quando uma modificação é feita, seja por uma operária ou por uma observadora, se v_{ij} ultrapassar o limite superior ou inferior da respectiva dimensão, este pode ser redefinido com o valor mínimo ou máximo, dependendo do limite que foi ultrapassado.

Outro ponto a ser observado no algoritmo ABC é que qualquer fonte de alimentação que não for melhorada, isto é, que não receber melhorias em uma determinada quantidade de ciclos, deve ser imediatamente abandonada. A melhoria na qualidade do néctar é considerada positiva se o *fitness* da nova solução for melhor ou igual ao anterior, assim permite-se que novas soluções sejam exploradas mesmo tendo a mesma qualidade de néctar da anterior.

3.3 Evolução Diferencial

A Evolução Diferencial, do inglês *Differential Evolution* (DE), é um algoritmo evolutivo criado por Storn e Price [25]. Seu funcionamento é muito semelhante ao de um algoritmo genético (GA) [18], porém enquanto os GA's executam cruzamento e mutação, na DE executam-se os operadores na ordem inversa, isto é, primeiro a mutação e depois o cruzamento. No Algoritmo 3 apresenta-se os passos executados pelo DE.

Assim como no PSO, no DE a população é inicializada de forma aleatória dentro do domínio de cada gene e seu *fitness* é avaliado. Ao entrar no laço cria-se o vetor de diferenças (v) de acordo com a Equação 8, na qual x_α, x_β e x_γ são três indivíduos escolhidos aleatoriamente dentro da população em que $\alpha \neq \beta \neq \gamma$, isto é, α, β e γ são diferentes, e (F) é um fator de mutação. Como os três indivíduos são escolhidos aleatoriamente, essa estratégia é chamada de *DE/Rand/Bin*. Quando x_α é substituído pelo melhor indivíduo da população, a estratégia passa a ser chamada de *DE/Best/Bin*.

$$v = x_\alpha + F(x_\beta - x_\gamma) \tag{8}$$

Algorithm 3 - Evolução Diferencial

```
Require: Inicializar parâmetros da população;
Avaliar população;
for de i = 1 até o tamanho da população do
    Criar vetor de diferenças;
    Formar novo indivíduo
    if $fit(novo) < fit(individuo_i)$ then
        Substituir novo no alvo
    else
        Manter individuo alvo
    end if
end for
```

Após gerar o vetor de diferenças, o novo indivíduo será criado da seguinte forma: para cada gene do indivíduo sorteia-se um valor r_i no intervalo $[0,1]$, sendo que i corresponde a um gene que deve obedecer a restrição de seu domínio. Se $r_i < CR$, ou seja, se o número sorteado for menor que a taxa de cruzamento *(Crossover Rate - CR)* então o gene do novo indivíduo será originário de v, caso contrário, o gene virá do indivíduo alvo i. Essa operação é semelhante ao cruzamento discreto dos AGs e pode ser representado como mostrado na Equação 9.

$$\text{novo}_i = \begin{cases} v_i, & \text{se} \quad r \leq CR \\ pop_i & \text{se} \quad r > CR \end{cases} \tag{9}$$

Após o cruzamento deve-se verificar se o novo indivíduo apresenta um *fitness* melhor que indivíduo alvo, em caso afirmativo o novo indivíduo substitui o indivíduo alvo, caso contrário o alvo permanece na população.

4 Meta-Heurística Adaptativa - AVEMH

Como mencionado anteriormente, as abordagens vetoriais são fáceis de implementar para problemas cuja quantidade de objetivos é par; e como evoluem a partir de subpopulações independentes, essas abordagens são fortes candidatas à paralelização. A Figura 3 mostra a ideia de como os algoritmos VE funcionam para dois objetivos, no qual cada subpopulação trabalha com um único objetivo. A chave que os faz funcionar é que a atualização de cada subpopulação é feita com base no melhor indivíduo da outra população.

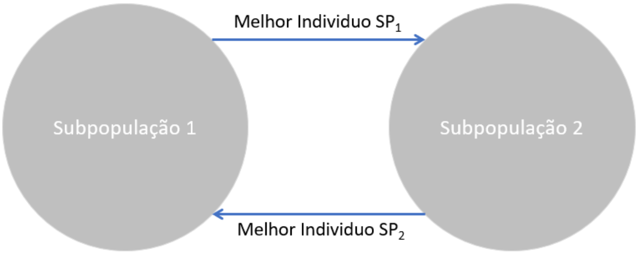


Figura 3: Mapeamento entre espaço de estados e espaço de objetivos. Adaptado de [5].

Matematicamente, considerando duas populações independentes P_1 e P_2 e sendo seus respectivos melhores indivíduos x_1 e x_2, então x_1 guia a atualização de SP_2 e x_2 guia a atualização de SP_1. É importante notar que não é necessário determinar a dominância neste ponto, pois as atualizações na posições de x são feitas pelos indivíduos das subpopulações opostas. Dessa forma, direciona-se SP_1 na direção de x_2 e SP_2 na direção de x_1. Esse comportamento tende a distribuir as soluções pela fronteira de Pareto.

Assim, quando se utiliza o *Vector Evaluated PSO* (VEPSO), por exemplo, a atualização da velocidade (v) na SP_1 usará o melhor indivíduo (g) da SP_2 e vice-versa. Com relação à DE, por estar baseado no melhor indivíduo, deve-se utilizar a estratégia $DE/Best/1$, caso contrário, uma forma de estabelecer dominância entre os indivíduos trocados deve ser estabelecida, o que acaba saindo um pouco da filosofia da avaliação vetorial. Já com relação ao algoritmo ABC, a troca entre as subpopulações é feita conforme o melhor néctar. Essa estratégia faz com que as soluções não se acumulem nos extremos da fronteira de Pareto. O fluxograma do AVEMH é apresentado na Figura 4.

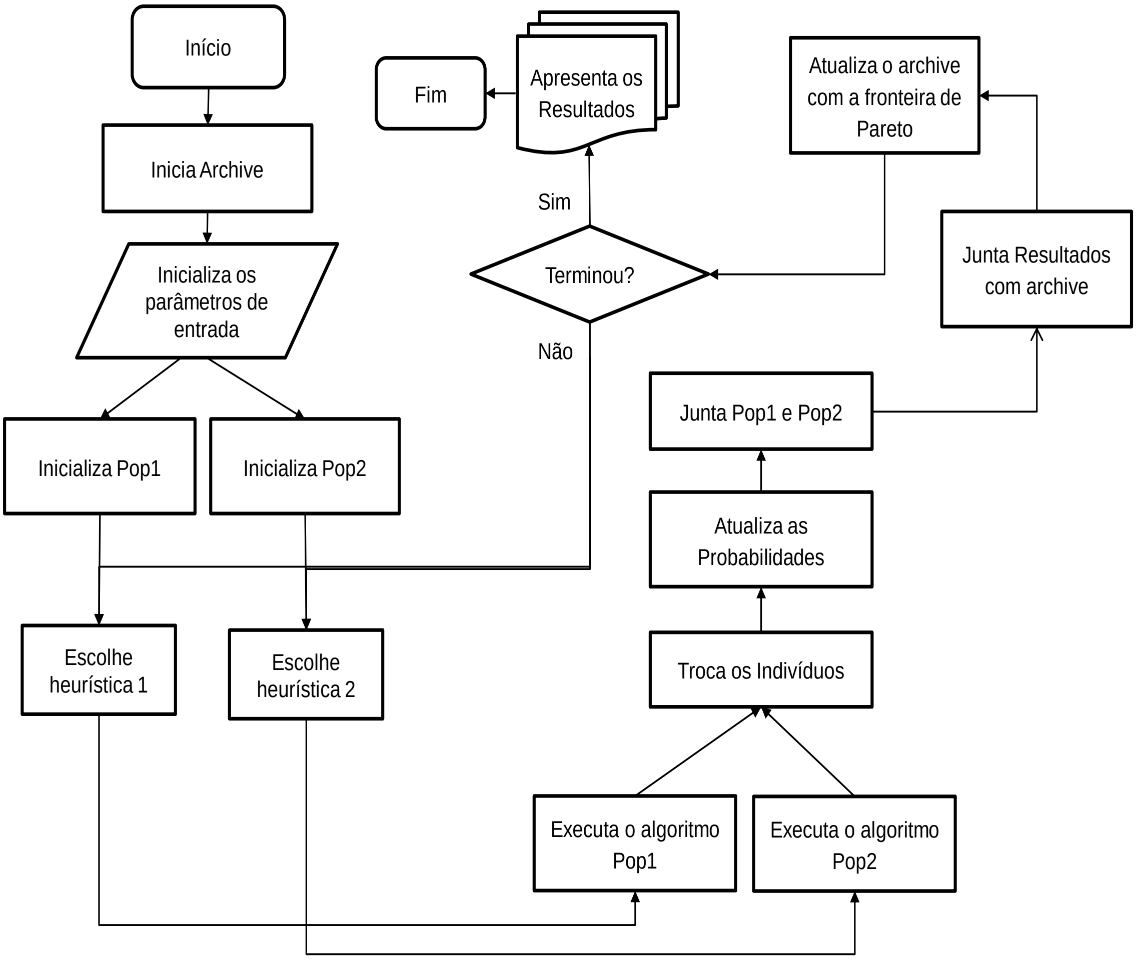


Figura 4: Fluxograma AVEMH

No começo, o AVEMH inicializa as probabilidades de cada heurística a ser escolhida de forma uniforme. Como são três possibilidades (PSO, DE e ABC) cada uma é inicializada com 0.33. Após serem definidas as probabilidades, as subpopulações são inicializadas de forma aleatória dentro do domínio do problema. Dentro do laço de iterações ocorre a escolha de qual meta-heurística utilizar, ocorre a troca de indivíduos entre subpopulações para que ocorra a atualização das soluções no espaço de busca. Em seguida, em cada subpopulação verifica-se se houve melhora no *fitness* da função.

A avaliação da população pode ser feita com duas abordagens. A primeira considera que se o melhor indivíduo veio da outra população então assume-se que houve uma melhora no ótimo da função. A outra abordagem utiliza a média da qualidade dos indivíduos da população, caso esta tenha melhorado com relação à média anterior, entende-se também que houve uma melhora na população.

Se houve melhora, então a heurística que enviou seu melhor indivíduo tem sua probabilidade aumentada, caso contrário, é diminuída. Nesse momento podem acontecer dois casos: *(i)* se houve uma melhora, a probabilidade da heurística que enviou o indivíduo é incrementada em 0.01, enquanto que as demais são diminuídas em 0.005; e *(ii)* se não houve melhora, a heurística que enviou o indivíduo tem sua probabilidade decrementada em 0.01 e as demais tem sua probabilidade aumentada em 0.005. Finalmente, as populações são unidas e as funções adjacentes são calculadas, ou seja, calcula-se f_2 em Pop_1 e f_1 em Pop_2. Em seguida os resultados são unidos ao *archive* de modo a determinar a nova fronteira de Pareto, isto é, as soluções dominadas são removidas do *archive*. Dado que as populações independentes não são alteradas e o laço re-inicia, não é necessário que haja uma etapa de separação.

Este modelo generalizado deve ser robusto o suficiente para conseguir resolver diversos problemas multiobjetivo. Para tanto, a meta-heurística AVEMH pode ser aplicada tanto em funções de *benchmark* quanto na resolução de problemas do mundo real, tais como, *Despacho Econômico e Ambiental de Energia* e a *Seleção de Portfólios*, dentre outros. O algoritmo 4 mostra a sequencia de operações em pseudocódigo de modo a facilitar seu entendimento.

Algorithm 4 - Algoritmo Multiobjetivo Vetorial - AVEMH

```
Require: Iniciar archive vazio
Iniciar Probabilidades de $Pop_1$
Iniciar Probabilidades da $Pop_2$
Escolher heurística da $Pop_1$
Escolher heurística da $Pop_2$
Inicializar parâmetros da população
repeat
    Trocar indivíduos
    Atualizar a subpopulação usando a meta-heurística escolhida
    if Melhorar solução é verdadeiro then
        Atualizar Probabilidades de quem enviou
    end if
    Calcular o fitnesss da função adjacente
    Juntar as subpopulações no archive
    Remover soluções dominadas do archive
until Alcançar o critério de parada
```

5 Experimentos

5.1 Termos, Configurações e Métricas

Aqui são apresentados os termos utilizados para nomenclatura, as configurações relacionadas ao ambiente de teste, as configurações de parâmetros utilizados para as execuções do AVEMH e a métrica adotada para avaliação dos resultados. Para uniformizar a nomenclatura, o melhor indivíduo será sempre chamado de *pBest*. No VEPSO o indivíduo migrado é o g. No VEDE o indivíduo migrado é o que apresenta melhor *fitness* na população. Já no VEABC será migrada a localização do melhor néctar. Todos os testes foram realizados em um computador com processador Intel(R) Core(TM) i5 com 2.50GHz, 8,00 GB de RAM e com sistema operacional Windows 8.1. de 64 bits. A implementação foi feita em linguagem Java versão 8 e NetBeans 8.2. As configurações das meta-heurísticas estão apresentadas na Tabela 2. Cada experimento foi executado 31 vezes para que sua distribuição seja considerada normal de acordo com o teorema do limite central, sendo possível assim a utilização de testes paramétricos na comparação dos resultados. Os parâmetros foram definidos como em [28], [3].

Tabela 2: Configurações usadas por cada meta-heurística

Meta-heurística	Parâmetro	Valor
VEPSO	inercia	1/(1/2 + ln(2))
	r_1, r_2	1/2 + ln(2)
VEABC	sobrevivem	90%
VEDE	taxa de crossover	0.9
	fator de mutação	0.7

A métrica utilizada nos testes é o hiper volume apresentado na Equação 10, na qual v_i representa um hipercubo construído a partir das soluções da fronteira de Pareto (conjunto Q). Assim, o hiper volume total é calculado pela união de todos os hipercubos v_i. Em outras palavras, o hiper volume é a área que esta sob a fronteira de Pareto. Assim, um maior o hiper volume pode tender a apresentar melhores fronteira de Pareto.

$$h_v = volume(\bigcup_{i=1}^{|q|} v_i) \tag{10}$$

5.2 Resultados nas Funções de Benchmark

Na Tabela 3 estão os valores calculados referentes ao maior, a média, a variância e o desvio padrão dos hiper volumes para as ZDTs, na qual pode-se observar que o valor real e o encontrado pela AVEMH são próximos. Na Tabela 4 é apresentada a quantidade de soluções encontradas pelo AVEMH em relação aos melhores hiper volumes encontrados para as ZDT's. Já a Figura 5 apresenta como as soluções encontradas se posicionam em relação à fronteira de Pareto Real.

Tabela 3: Resultados Calculados pelo AVEMH para as ZDTs

Função	Real Hv	Melhor Hv	Média Hv	Variância	D. Padrão
ZDT1	120.6621	120.6616	120.5367	0,000238	0.015440
ZDT2	120.3288	120.3245	120.2548	0,000138	0.011745
ZDT3	128.7781	128.7714	128.7714	3E-26	1E-13
ZDT4	120.6661	120.6621	120.6504	2E-26	1E-13
ZDT6	117.5149	117.4579	114.4111	0.27E+3	16.4851

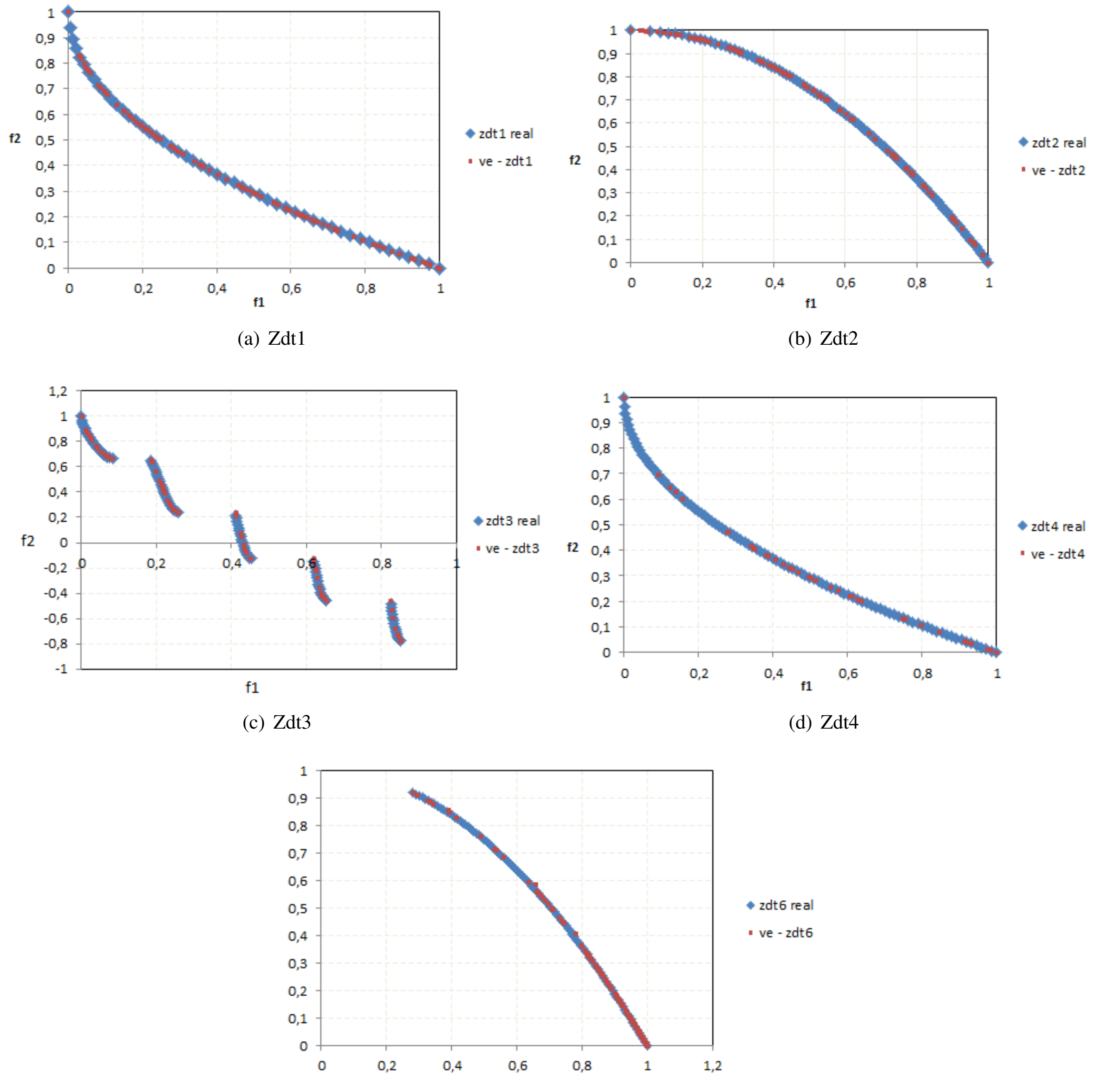
Tabela 4: Hiper volumes atingidos pelo AVEMH para as ZDT's.

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
N^o de gerações			100		
Dimensão	30	30	30	10	10
N^o de soluções na fronteira	200	73	33	59	10
Hiper volume - Melhor	120,6616	120,3245	128,7714	120,6621	117,4579
Hiper volume - Médio	120,6615	120,3230	128,7258	120,5771	117,1978

Ainda nas ZDTs, a Tabela 5 faz um comparativo dos hiper volumes entre a AVEMH e os algoritmos VE sem adaptação. O AVEMH alcança melhores resultados nas ZDT1, ZDT2 e ZDT3. O VEABC encontra o melhor hiper volume na ZDT4 e o VEDE o melhor resultado na ZDT5, muito embora todos os valores sejam muito parecidos. Com intuito de verificar se realmente existem diferenças significativas entre os resultados apresentados pelo AVEMH, VEPSO, VEABC e VEDE foi realizada uma análise de variância (ANOVA) sobre os dados obtidos nos hiper volumes com significância de 95% (α = 0,05) e um teste de Tukey apresentados nas Tabelas 6 e 7. Como se pode observar, o teste ANOVA indica que há diferenças significativas em todas as funções ZDTs.

Tabela 5: Comparativos dos melhores resultados atingidos pelo AVEMH em relação aos melhores resultados atingidos com VEPSO, VEABC e VEDE, nas funções ZDTs.

Função	Real Hv	AVEMH Hv	VEPSO Hv	VEDE Hv	VEABC Hv
ZDT1	120.6621	**120.6616**	120.6547	120.6161	120.6562
ZDT2	120.3288	**120.3245**	120.3174	120.2431	120.3244
ZDT3	128.7781	**128.7714**	128.5893	128.6462	128.6250
ZDT4	120.6661	120.6621	120.6633	120.6607	**120.6645**
ZDT6	117.5149	117.4579	117.2347	**117.5116**	117.1973



(a) Zdt1

(b) Zdt2

(c) Zdt3

(d) Zdt4

(e) Zdt6

Figura 5: Comparativo da fronteira real das funções de *benchmark* com as fronteiras encontradas pelo AVEMH para as ZDTs.

O teste de Tukey na Tabela 7 indica que o AVEMH e VEABC obtiveram o melhor desempenho na ZDT1 já que a diferença não é significativa. Nas ZDT2 e ZDT3, o AVEMH obteve realmente os melhores resultados. O algoritmo VEABC juntamente com o VEPSO obtiveram o melhor resultado na ZDT4. E na ZDT6, o algoritmo VEDE realmente obteve os melhores resultados.

5.3 Resultados na Otimização de portfólios

Para a resolução da Seleção de Portfólios fez-se um comparativo dos resultados encontrados pelo AVEMH com a fronteira eficiente dos arquivos disponíveis para testes em `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html`. Neste trabalho, os conjuntos de dados são os 'Hong Kong Hang Seng', 'German Dax 100','British FTSE 100', 'US S&P 100' e 'Japonese Nikkei' que são compostos por 63, 85, 89, 98 e 225 ativos, respectivamente.

Tabela 6: Teste ANOVA comparando o AVEMH com as demais meta-heurísticas nas funções ZDT

F crítico = 2,73									
Função	F	Função	F	Função	F	Função	F	Função	F
ZDT1	11,78	ZDT2	653,6	ZDT3	3888	ZDT4	10,4	ZDT6	100,1

Tabela 7: Teste de Tukey comparando o AVEMH com as demais meta-heurísticas nas funções ZDTs

ZDT1					
	p		p		p
AVEMH vs VEABC	0,998	AVEMH vs VEDE	0,0007	AVEMH vs VEPSO	0,00006
	p		p		p
ZDT2					
	p		p		p
AVEMH vs VEABC	0,158	AVEMH vs VEDE	0,0000	AVEMH vs VEPSO	0,0000
ZDT3					
	p		p		p
AVEMH vs VEABC	0,0000	AVEMH vs VEDE	0,0000	AVEMH vs VEPSO	0,0000
ZDT4					
	p		p		p
AVEMH vs VEABC	0,0525	AVEMH vs VEDE	0,1738	AVEMH vs VEPSO	0,039
ZDT6					
	p		p		p
AVEMH vs VEABC	0,9986	AVEMH vs VEDE	0,0000	AVEMH vs VEPSO	0,0026

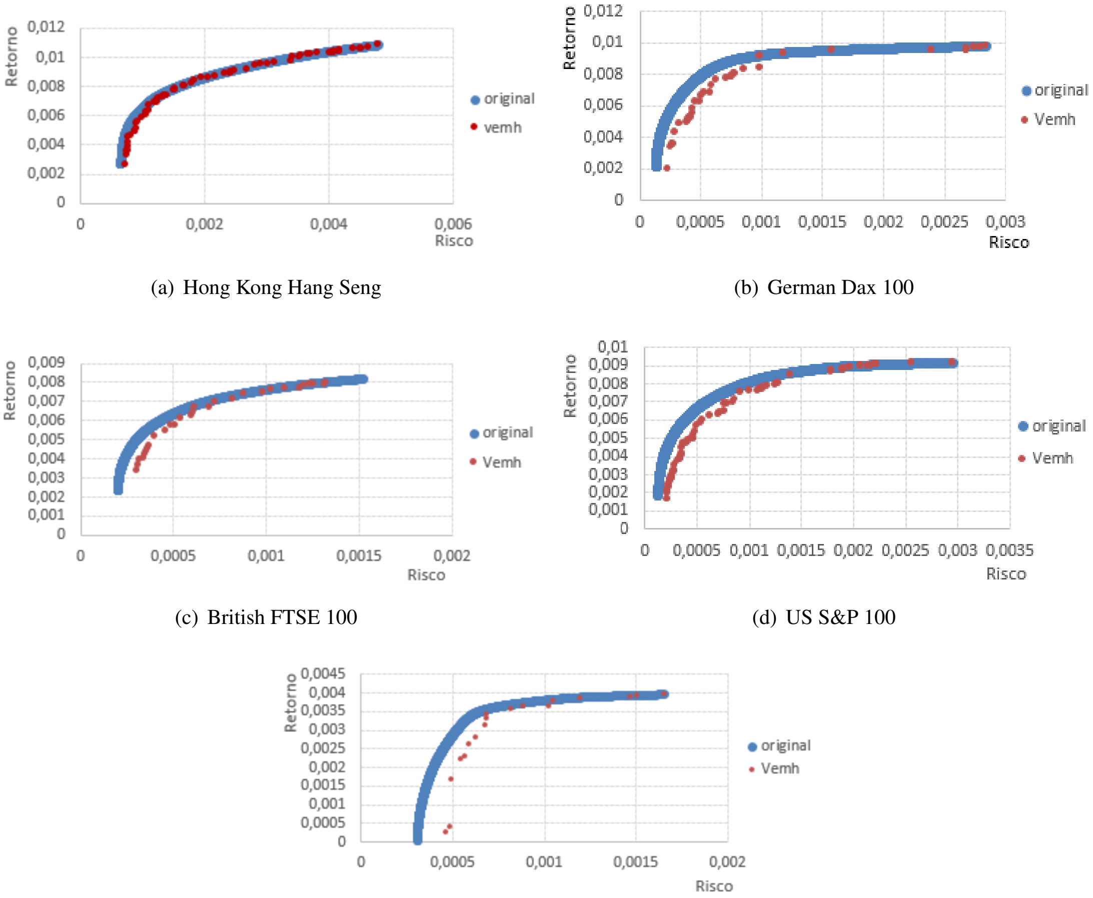
Nestes conjuntos de dados são disponibilizados a média dos retornos dos ativos, o desvio padrão dos retornos e a correlação entre esses ativos. O Hong Kong Hang Seng, conhecido oficialmente como 'Hang Seng Index' (HSI), é o principal índice da bolsa de valores de Hong Kong [22]. Já o índice German Dax, conhecido por 'DAX Índex', é formado pelas maiores empresa da bolsa de valores de Frankfurt na Alemanha [7]. O British FTSE é composto por 30 maiores empresas de capital aberto do Reino Unido, conhecido por 'FTSE' este índice afeta a economia britânica e mundial representando boa parte da capitalização na bolsa de valores de Londres [8]. O US S&P 100 é um índice do mercado de ações norte-americano mantido pela Standard & Poor's sendo formado por 101 empresas que possuem ações nas bolsas de valores NYSE e NASDAQ [12]. E por fim, o Japonese Nikkei, conhecido por 'Nikkei 225' é o principal índice econômico da bolsa de valores de Tóquio [24].

Na Tabela 8 são apresentados o número de gerações utilizadas, o número de ativos em cada problema, a quantidade de soluções encontradas e os hiper volumes calculados pelo AVEMH em cada uma das funções de testes quando da resolução da Seleção de Portfólios. A Figura 6 apresenta a comparação entre a fronteira de Pareto ótima e os testes realizados com o AVEMH e os conjuntos de dados do problema da Seleção de Portfólios.

Tabela 8: Hiper volumes atingidos no problema de seleção de portfólios

	HST	DAX	FTSE	S&P 100	Nikkei 225
N^{o} de gerações			2000		
N^{o} de ativos	31	85	89	98	225
N^{o} de soluções encontradas	65	58	36	180	30
Hiper volume - Melhor	120,9459	120,9724	120,9649	120,9650	120,9824
Hiper volume - Médio	120,8725	120,8901	120,9018	120,8815	120,8999

Na Tabela 9 estão disponíveis a média dos melhores resultados atingidos pelo AVEMH em relação aos resultados atingidos com o VEPSO, VEABC e VEDE, para os problemas da Seleção de Portfólios. Como se pode observar, o AVEMH apresenta uma média melhor nos portfólios HST, DAX e FTSE, enquanto que o VEDE apresenta uma melhor média no S&P 100 e Nikkei.



(a) Hong Kong Hang Seng

(b) German Dax 100

(c) British FTSE 100

(d) US S&P 100

(e) Japonese Nikkei

Figura 6: Comparativo da fronteira real das funções de *benchmark* com as fronteiras encontradas pelo AVEMH para a Seleção de Portfólios.

Com intuito de verificar se realmente existem diferenças significativas entre os resultados apresentados pelo AVEMH e as demais meta-heurísticas foi realizada uma análise de variância (ANOVA) com grau de significância de 95% (α = 0.05) e F crítico igual a 2,73 conforme apresentado na Tabela 10. Como pode ser observado há diferença significativas nos portfólios HST e DAX. Para identificar as diferenças fez-se um teste de Tukey conforme apresentado na Tabela 11, na qual pode-se notar que as maiores diferenças estão a favor do AVEMH e em contra ao VEDE e a VEABC, ou seja, a AVEMH apresenta melhores resultados que o VEDE e o VEABC. Além disso, não há diferença significativa entre o AVEMH e o VEPSO.

A Tabela 12 apresenta o tempo médio de cada execução para cada portfólio em minutos. Em média o AVEMH é mais rápido que o VEABC na otimização do HST, S&Pe e Nikkei 225. E mais rápido que o VEDE em todos os portfólios. Embora o VEPSO seja o mais rápido, não é o algoritmo que consegue alcançar as melhores configurações de portfólios como previamente apresentado.

6 Conclusões e Trabalhos Futuros

Este trabalho apresentou uma meta-heurística adaptativa baseada em avaliação vetorial para resolução de problemas multiobjetivo, em especial aplicado na otimização de portfólios de investimentos. A principal vantagem da avaliação vetorial é ser de fácil implementação e não consumir tempo do algoritmo com operações baseadas em dominância. Além disso, a adaptação em tempo de execução, em teoria, permite solucionar uma gama maior de problemas sem a

Tabela 9: Resultados atingidos pelo AVEMH em relação aos resultados atingidos com VEPSO, VEABC e VEDE, para o problema de Seleção de Portfólio considerando a média do hiper volume.

Função	Gerações	AVEMH Hv	VEPSO Hv	VEABC	VEDE Hv
HST	2000	**120,8725**	120,8592	120,7398	120,8191
DAX	2000	**120,8901**	120,7752	120,8895	120,8898
FTSE	2000	**120,9018**	120,8869	120,8926	120,8979
S&P 100	2000	120,8815	120,8552	120,8620	**120,9067**
Nikkei 225	2000	120,8999	120,8666	120,8772	**120,9010**

Tabela 10: Teste ANOVA comparando o AVEMH com as demais metaheurísticas

F crítico = 2,73									
Portfolio	F	Portfolio	F	Portfolio	F	Portfolio	F	Portfolio	F
HST	5,596	DAX	5,603	FTSE	0,056	S&P 100	0,055	Nikkei	0,664

Tabela 11: Teste de Tukey comparando o AVEMH com as demais meta-heurísticas nos portfólios HST e DAX

HST					
	p		p		p
AVEMH vs VEABC	**0,0017**	AVEMH vs VEDE	**0.4432**	AVEMH vs VEPSO	0.9821
	p		p		p
DAX					
AVEMH vs VEABC	**0,0017**	AVEMH vs VEDE	**0.4424**	AVEMH vs VEPSO	0.9822

Tabela 12: Tempo de execução para a Seleção de Portfólios - 2000 iterações

	AVEMH		VEPSO		VABC		VEDE	
Dados	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
HST	0,30 m	0,0086 m	0,18 m	0,0075 m	0,34 m	0,0098 m	0,38 min	0,0416 m
DAX	1,03 m	0,0088 m	0,54 m	0,0200 m	0,90 m	0,0118 m	1,05 m	0,027 m
FTSE	1,00 m	0,0066 m	0,57 m	0,0183 m	0,98 m	0,0128 m	1,12 m	0,0338 m
S&P	1,04 m	0,009 m	0,71 m	0,0085 m	1,14 m	0,0056 m	1,26 m	0,0295 m
Nikkei 225	3,25 m	0,0268 m	2,47 m	0,0115 m	3,48 m	0,0278 m	3,72 m	0,1415 m

necessidade de explorar a combinação entre as meta-heurísticas utilizadas. O resultados mostraram que a AVEMH apresentou melhores hiper volumes nas funções de benchmark ZDT1, ZDT2 e ZDT3; e na otimização de portfólio de investimentos dos portfólios HST, DAX e FTSE. Por outro lado, algumas diferenças não foram estatisticamente significativas possivelmente pela quantidade de iterações utilizadas. Nesse contexto, espera-se em trabalhos futuros testar a meta-heurísticas em problemas mais complexos, como por exemplo o S&P 500, para verificar se a adaptação estocástica produz efetivamente melhores fronteiras de Pareto.

Outros trabalhos futuros incluem: (i) paralelização dos algoritmos tanto em arquiteturas multi-núcleo (*multicore*) quanto em arquiteturas com muitos núcleos (GPU - *many cores*); (ii) incluir a auto-adaptação no algoritmo na qual os parâmetros como F, w, c_1, c_2, dentre outros sejam incluídos nos genes dos indivíduos e decididos em tempo de execução; (iii) utilizar a lógica nebulosa para determinar qual meta-heurística utilizar; e (iv) aplicar o algoritmo AVEMH na otimização de parâmetros em algoritmos de aprendizagem de máquina.

É interessante ressaltar que futuramente esta meta-heurística pode ser paralelizável com o objetivo de melhorar o tempo de execução na resolução de problemas do mundo real mais custosos, tal como a seleção de portfólios usando o S&P 500. Também é viável o desenvolvimento de um modelo fuzzy para realização da análise de melhora da população. E ainda, tornar o algoritmo suficientemente robusto para a resolução de mais problemas do mundo real.

Referências

[1] H. P. Borges and D. Cortes, O. A. C.and Vieira. An adaptive metaheuristic for unconstrained multimodal numerical optimization. In P. Korošec, N. Melab, and El-G. Talbi, editors, *Bioinspired Optimization Methods and Their Applications*, pages 26–37, Cham, 2018. Springer International Publishing.

[2] E. Carvalho Jr., O. A. C. Cortes, J. P. Costa, and A. Rau-Chaplin. A stochastic adaptive genetic algorithm for solving unconstrained multimodal numerical problems. In *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pages 130–137, May 2016.

[3] O. A. C. Cortes, A. Rau-Chaplin, and P. F. do Prado. A new vector evaluated PBIL algorithm for reinsurance analytics. In *2015 Latin America Congress on Computational Intelligence (LA-CCI)*, pages 1–6, Oct 2015.

[4] J. P. A. Costa, E. Carvalho Jr., and O. A. C. Cortes. An adaptative algorithm for updating populations on SPEA2. In *XII Simpósio Brasileiro de Automação Inteligente - Porto Alegre - RS*, pages 78–83, 2017.

[5] K. Deb. *Multi-Objetive Optimization using Evolucionay Algorithm.* Jhon Wiley & Sons, LTD, 2001.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Transaction on Evolutionary Computing*, 6(2):182–197, April 2002.

[7] Disford. Top 30 companies of germany in the dax index 2019. https://disfold.com/top-companies-germany-dax/, 2019.

[8] Disford. Top 30 companies of the uk in the ftse index 2019. https://disfold.com/top-companies-uk-ftse/, 2019.

[9] ETH. Systems optimization. https://sop.tik.ee.ethz.ch/download/supplementary/testpro-blems/, 2019.

[10] C. M. Fonseca and P. Fleming. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. *the fifth Intl conference on Genetic Algorithms*, 93, 02 1999.

[11] F. Glover and A. G. Kochenberger. *Handbook of MetaHeuristics.* Kluwer Academic Publishers Boston, New York, 2005.

[12] S&P Down Jones Indices. S&p 100. https://us.spindices.com/indices/equity/sp-100, 2019.

[13] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39(3):459–471, Nov 2007.

[14] D. Karaboga and B. Basturk. On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing*, 8(1):687–697, 2008.

[15] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Nov 1995.

[16] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7:77–91, March 1952.

[17] H. Markowitz, W. F. Sharpe, and M. H. Miller. The founders of modern finance: Their prize winning concepts and 1990 nobel lectures. *CFA Intitute*, 1990.

[18] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.)*. Springer-Verlag, Berlin, Heidelberg, 1996.

[19] S. N. Omkar, J. Senthilnath, R. Khandelwal, G. Narayana Naik, and S. Gopalakrishnan. Artificial bee colony (abc) for multi-objective design optimization of composite structures. *Applied Soft Computing*, 11(1):489 – 499, 2011.

[20] K. E. Parsopoulos, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis. Vector evaluated differential evolution for multiobjective optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 1, pages 204–211, June 2004.

[21] K. E. Parsopoulos and M. N. Vrahatis. Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM Symposium on Applied Computing*, SAC '02, pages 603–607, New York, NY, USA, 2002. ACM.

[22] T. Reis. Hang seng: saiba mais sobre o principal índice da bolsa de Hong Kong. https://www.sunoresearch.com.br/artigos/hang-seng/, 2019.

[23] J Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the First Int. Conference on Genetic Algortihms, Ed. G.J.E Grefensette, J.J. Lawrence Erlbraum*, pages 93–100, 01 1985.

[24] Nikkei 225 Official Site. Nikkei stock average (nikkei 225). https://indexes.nikkei.co.jp/en/nkave/index/profile?idx=nk225, 2019.

[25] R. Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, Dec 1997.

[26] B Y. Qu, Q Zhou, J M. Xiao, Jing Liang, and Ponnuthurai Suganthan. Large-scale portfolio optimization using multiobjective evolutionary algorithms and preselection methods. *Mathematical Problems in Engineering*, 2017:1–14, 02 2017.

[27] I. Yevseyeva, A. P. Guerreiro, M. T. M. Emmerich, and C. M. Fonseca. A portfolio optimization approach to selection in multiobjective evolutionary algorithms. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, pages 672–681, Cham, 2014. Springer International Publishing.

[28] M. Zambrano-Bigiarini, Maurice C., and R. Rojas. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In *IEEE Congress on Evolutionary Computation, CEC*, pages 2337–2344, 06 2013.

[29] W. Zhang and S. Fujimura. Improved vector evaluated genetic algorithm with archive for solving multiobjective pps problem. In *2010 International Conference on E-Product E-Service and E-Entertainment*, pages 1–4, Nov 2010.

[30] G. Zhu and S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7):3166–3173, 2010.

[31] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Proceedings of the EUROGEN*, volume 3242, 01 2001.

Inteligencia Artificial 22(64), 102-122
doi: 10.4114/intartif.vol22iss63pp102-122

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

A robust multi-agent Negotiation for advanced Image Segmentation: Design and Implementation

Hanane Allioui[1], Mohamed Sadgal[2], Aziz Elfazziki[3]
Computer science Department, Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, Morocco

Abstract It is generally accepted that segmentation is a critical problem that influences subsequent tasks during image processing. Often, the proposed approaches provide effectiveness for a limited type of images with a significant lack of a global solution. The difficulty of segmentation lies in the complexity of providing a global solution with acceptable accuracy within a reasonable time. To overcome this problem, some solutions combined several methods. This paper presents a method for segmenting 2D/3D images by merging regions and solving problems encountered during the process using a multi-agent system (MAS). We are using the strengths of MAS by opting for a compromise that satisfies segmentation by agents' acts. Regions with high similarity are merged immediately, while others with low similarity are ignored. The remaining ones, with ambiguous similarity, are solved in a coalition by negotiation. In our system, the agents make decisions according to the utility functions adopting the Pareto optimal in Game theory. Unlike hierarchical merging methods, MAS performs a hypothetical merger planning then negotiates the agreements' subsets to merge all regions at once.

Keywords: Image segmentation, Region merging, Multi-Agent System, Game Theory, Coalition, Negotiation.

1 Introduction

Image segmentation indicates the partitioning process of an image into regions of interest by following a set of given criteria. This is concerning the different characteristics of the image such as colour or texture [1]. Image segmentation is applied in many applications such as object recognition [2, 3], target tracking [4], content-based image processing [5] and medical image processing [6, 7, 8]. Generally, the segmentation is about producing numerous partitions(segments) having homogeneous characteristics, then group the significant parts. Many studies and works related to image segmentation, utilized region concept as a significant pre-processing stage. The region is an image concept that is simple and more resistant to noise in segmentation algorithms [9].

The main purpose of region techniques is to join regions having the same homogeneity criteria. However, these approaches don't have a strong determination of the natural limits of the objects due to their less satisfactory results concerning the preservation of the global properties. For that, cooperative segmentation provides an additional advantage, combining several methods or algorithms, to achieve reliable results in less time. Otherwise, cooperation remains an important property of multi-agent systems, it involves the collaboration assumptions, negotiation plans, and resolution conflicts strategies. MAS [10, 11], have proved better results, as they take into account the characteristics of the entities in the image.

The objective of this paper is to implement a cooperative image segmentation algorithm, based on the concept of region merging, where similar neighboring regions are merged according to a novel process. After being placed on regions, the agents can decide the merger under some constraints that satisfy similarity predicates. Principally, we are interested in the use of MAS in the 2D/3D region segmentation. This proposal aims to apply several principles among the model agents (communication, coordination, negotiation, and cooperation), whose objective is building efficient segmentation. Adopting cooperation between segmentation techniques improves the quality and reliability of analyzes and possible decisions [12].

Our adopted mechanism to solve conflicts uses agent negotiation which is based on game theory techniques [13, 14, 15]. In our system, the most important stage is the region merging decision. Agents are organized in a

ISSN: 1137-3601 (print), 1988-3064 (on-line)

neighboring structure according to the region adjacency (RAG) [16, 17]. Each agent proposes a merging plan (set of merging regions) which is transmitted to all its neighbors. Next, the closer neighbor evaluates the received plans and decides to accept or reject the hole plans or a specific plans' subset. The retained subset of all proposed plans will be the compromise between all neighbors. Our experimental results indicate the effectiveness of the proposed algorithm.

This paper will be organized in the following sections: In section II, we specify the related works about cooperative segmentation. In section III, we explain the adopted segmentation modalities. In section IV, we present the general resolution approach. In section V, we clarify the experimental results. Finally, in section VI, we provide a conclusion of our paper.

2 Related works

2.1 About region segmentation techniques

In the field of image segmentation, region-based approaches normally apply various protocols. For instance, region growing [18] is based on the seed points to establish similar neighboring grouping pixels [19]. Splitting and merging techniques are wildly used in the image segmentation [20]. The principle of “region segmentation” is based on the combination of the regions Ri in a way to form an image I=U R_i with i = {1,2,3, ..., n}. If there is no match between the two regions, then no merging action is possible. In image segmentation, many techniques have been used such as superpixel, region growing, region splitting and merging [21]. Whereas, the region growing technique is based on the growing of seed points in order to acquire several homogeneous regions [22]. However, its major drawback is the difficulty to select the appropriate initial seed automatically, and, to deal with the noises and regions with holes form [23]. Region merging aims to join sub-regions together to produce a meaningful merge.

Most methods use over-segmentation results to determine the initial regions. After that, according to a certain criterion, the segmentation is carried out by merging the similar neighboring regions. Whereas segmentation is achieved by making local decisions, some techniques have been shown effectiveness [24, 25]. In region merging techniques, the objective is to merge regions that meet a homogeneity criterion. In previous works, there are region-based fusion algorithms based on statistical properties [24, 26, 27], graph properties [28, 29, 30] or spatiotemporal similarity [25].

Through decades, several leading works of cooperative segmentation have been made. The cooperation of edge\ region-based [31, 32], self-organizing maps (SOMs) \ fuzzy C-means (FCM) [33], watershed \ region merging [34], regions\contours [35] …The main advantage of these cooperative imaging methods is to provide results with a high-quality level. In fact, each method represents a set of limits. For that, combining methods, techniques, or even tasks have been subject to different scientific works to reinforce the systems, and to ensure a good quality of the proposed solutions. For instance, Authors in [36] proposed an active contour model using hybrid region information. That allowed them to ensure better detection of small structures than the traditional models based on the length of a feature's boundaries. In [37] authors were able to extract significant regions using two steps (K-means clustering and region growing).

Later, authors in [38] presented an approach to improve the segmentation of brain MRI images based on convolutional neural networks (CNN) [39] by optimizing the loss function during training. In the same sense, Lei Bi et al., a stacked of Fully Convolutional Networks (FCN) [40] architecture with multi-channel learning to separate the characteristics of the region of interests, belonging to the background of those who do not, to be able to integrate these channels in final segmentation results. Definitely, rather than sequentially exploiting multiple approaches to improve segmentation, it can be more interesting to run multiple segmentation methods or tasks simultaneously. The whole issue of cooperative segmentation lies in the definition of modes of a combination of different sources of information and their use.

2.2 About Game theory

The Game theory is a challenging formalism that aims to study the planned, real or subsequently justified agents’ behavior in antagonism situations (opposition), and to highlight optimal strategies [41]. This theory is based on the concept of a game defined by a set of players (agents), regarding all possible strategies for each player, also the specifications of players’ gain for each combination of strategies [42]. In order to organize these concepts, several types of games were set:

- Cooperative games.
- Non-cooperative games.

- Finished and infinite games.
- Synchronous and asynchronous games.
- Zero-sum games and non-zero-sum games.
- Comprehensive information games and perfect information games.

As a base of a game, an agent can make decisions and take action. For an agent, the strategy defines a complete plan of possible actions during the game. One of the most useful strategies is Nash equilibrium which is a strategy defined by John Nash [43]. It ensures a steady situation, during the players' interactions. If none of them has the interest to change its strategy, the game becomes stable. That means that no player can change its strategy without weakening his own position [44]. Theoretically, it is a combination of strategies to satisfy each player i.

In a cooperative environment, each agent acts to respond to system needs. However, there is no alternative strategy that improves all the earnings of the agents simultaneously. This can only take place by a Pareto optimal [45]. The optimality of Pareto is considered as an efficiency concept. Thus, no state will be optimized, at least one player can get more gains without the other player. There are many examples of Nash equilibria that are not optimal. A result is optimal if no result allows at least one player to be better without the players losing more.

2.3 Image segmentation and game theory

Historically, few works were interested in the correspondence between game theory and image segmentation. One of the possible reasons is game theory grounding as a principle of economic needs satisfaction. Whereas, the first published work is that of A. Chakraborty et al. [46, 47]. In this section, we will mention several works in this area. Initially, Chakraborty presented an original and remarkable work [46]. It is based on a solid mathematical model integrating game theory with image segmentation by cooperation between the edge detector (active contour) and the Region detector using Markov Random Fields. We consider this work as a reference for our method. Years later, E. Cassel et al. [48] proposed a modified and simplified implementation of Chakraborty et al [46]. This simplification involves the removal of the "Prior Information on Shape to Segment" in the edge detector equation. The authors opted for the "growing region" as a region detector and for the morphological operation "for closure" of the edge detector.

Even in [49] authors proposed an approach to iris and pupil segmentation based on the work of Chakraborty et al [46]. However, this approach is particularly suited to this particular area of application. For this, they integrated the pre-treatment and post-treatment phases in their procedure. In this work, the methods "Growing Region" and "Levels" were used. Later, the works of [50], consist of two individual segmentation approaches (edge-based segmentation only). The authors proposed a supervised algorithm based on game theory and dynamic programming for pulmonary field segmentation, while Kallel et al. presented a game-based approach to simultaneously restore and segment noisy images [51]. Lastly, authors in [52] presented a Polarimetric synthetic aperture radar image classification which is an important and continually developing issue in the automatic analysis of remote sensing data. The regions or over-segments are merged into clusters using a game theory-based approach. In the repetitive game, the region merging problem is conducted by an iterative figure/ground separation.

3 The proposed approach

3.1 Merging region using Superpixels/Supervoxels

Superpixel has become ubiquitous in image processing to group pixels into meaningful regions (Superpixels) [53]. In fact, fast pixel grouping allows fast region merging in order to offer trustworthy segmentation results. Superpixels, introduced in [54], have become an essential part of vision researches. In general, the methods producing Superpixels can be collected into a group based on graphs [55, 56] of pixels (Fig 1), such as Slic [57] and Mean Shift, and evolution of curves, such as Turbopixels [58] and seeds.

In fact, fast pixel grouping allows fast region merging in order to offer trustworthy segmentation results. Thus, in 3D analysis, the Supervoxel is agreed to connect similar voxels in 3D regions. Typically, each superpixel is considered as a non-overlapping region with an adaptive shape [59]. Typically, region merging algorithms have some specifications especially the merging criterion which defines the merging threshold. The choice is to study carefully some global properties for the segmentation results. In fact, we use the Region Adjacency Graph (RAG) [60] to have a global representation of the image. In RAG, the adjacency of two nodes specifies that while the segmentation their corresponding pixels are in the immediate neighbourhood of the predefined threshold. An example of the graph structure is shown in Fig. 1, the initial image has 8 partitioned regions, hence the RAG is shown on the right side.

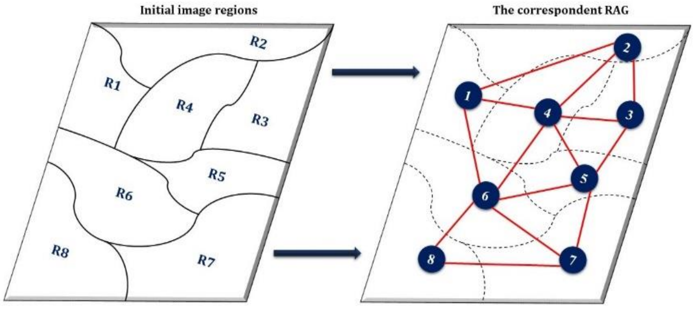


Fig. 1. Region partition and the corresponding region adjacency graph (RAG)

Merging techniques have been the subject of different researches. Previously, the authors explained in [61, 62, 63] that an image can be partitioned, so a set of unconnected regions would be merged to satisfy the similarity conditions. As well in [64], the merge predicate considers the minimum weighting edge between two regions to measure the difference between them. The team of [65] presented an iterative algorithm that considers all similar regions, as well as the gradient image and the edge map [66]. Likewise, [67, 68] proposed a fusion of neighbouring regions following a predicate. Also, the fusion predicate is obtained from the statistical concentration inequalities based on different properties contained in the region. Our initial motivation to use the merging criteria was born out of a pragmatic research methodology: overcome the regions merging deficiencies to obtain objects with more precision.

3.1.1 The merging decision

Selecting the similarity definition between adjacent regions R_i and R_j is an important step, it enables making decisions and obtaining an optimal segmentation result. This similarity can be defined based on the correlation between features or, on some distance definitions. We call the similarity function ‘S’ so S_{ij} denotes the similarity value between R_i and R_j. Since specifying the threshold value, with tidy precision, remain a difficult task, we assume that a surrounding margin exists around this threshold in case of uncertain decision. For these cases, we define an uncertainty interval, depending on the application domain, limited by two thresholds: a low threshold T_1 and a high threshold T_2. Fig. 2 illustrates this process.

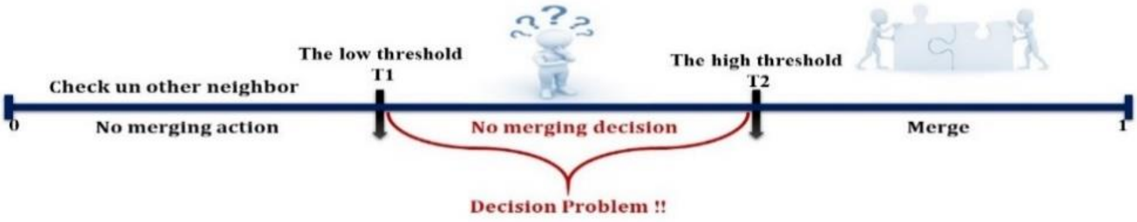


Fig. 2. The merging interval actions

We define the region merging decision strategy (Fig.2):

$$\text{The merging Decision} \begin{cases} \text{if: } S < T_1, \text{then No merging action is applied} \\ \text{if: } S > T_2, \text{then Merge imediatly} \\ \text{If : } T_1 < S < T_2, \text{then no merging decision} \end{cases}$$

In case, the Similarity value is in the interval:] T_1, T_2[all the merging suggestions are considered as uncertain. So, to handle this merging problem, it is important to adopt a cooperative technique, as it is explained in the following sections.

3.1.2 The merging problem

Two questions must be resolved with the above decision:

- How to choose the thresholds T_1 and T_2?
- How to continue the merging process for the remaining regions?

The thresholds (T_1 and T_2) depends on the nature of images and the features used in the Similarity definition. Statistical analyses can be made on some datasets of segmented images (PASCAL VOC [69], BSD500 [70], ...) allow the thresholds' estimation. The implication of Experts of the application domain and images' nature is required to validate the estimation. To continue the merging process, it is necessary to associate all neighbours of a Region and a complement to the defined Utility to refine the decision. For this propose, we consider regions as players (agents) in game theory that can evaluate the merging decision with its adjacency regions but the decisions of all its neighbours too.

3.2 Formulation of the problem using game theory

3.2.1 Game formulation

In Game Theory, the game is a formal model that usually includes a set of players (agents) and different actions or strategies available for each of them. There is a reward for each agent according to the combination of strategies. This approach has been used in computer vision, including clustering [71, 72, 73]. In a cooperative game, agents can cooperate in a predetermined way. A simple form of cooperation allows agents to form coalitions. By acting together in a coalition, agents can enhance their rewards against what they could have achieved by playing according to their personal interests. The problem is to implement a practical coalition structure in order to optimize the overall gain.

a. Basic concepts

In our context, the nodes in the graph (RAG) are replaced by autonomous and dynamic agents. As well, links of the adjacency graph are replaced by acquaintance relations between agents. An acquaintance between two agents means that they are neighbours in the image. This organization represents the region information which is in the image at a particular resolution.

Moreover, regions are characterized by a set of features $f_1, ..., f_p$. These features are based on:

- statistic features: Means, Variances, Histograms, etc.
- geometric features: Shapes, Boundaries, etc.
- Topological features: Betting numbers (adjacency properties), etc.

Each player (agent) A_i is situated in a Region Ri with information as feature values: $v_{i1}, \ldots, v_{ip}$. Respectively, an agent A_i has two possible actions: merge (a_1) or not-merge (a_2). To calculate the payoffs or rewards, we consider a utility Function U which depends on features. This utility can be formulated as a combination of mutual motivations like similarity and individual motivation. For instance, for two agents, the payoffs could be illustrated as follows:

		A_j	
		a_1	a_2
A_i	a_1	(u^i_{11}, uj_{11})	(u^i_{12}, uj_{12})
	a_2	(u^i_{21}, uj_{21})	(u^i_{22}, uj_{22})

Where: $\mathbf{u^i_{kk'}}$ is the U value (payoff) for the agent A_i using action a_k and the other Agent using the action $a_{k'}$

- A situation $(a^i_{k*}, a^j_{k'*})$ is Nash equilibrium if: $\mathbf{u^i_{kk*'}} \leq \mathbf{u^i_{k*k*'}}$ $\forall$ k $\in\{1,2\}$ and $\mathbf{u^j_{k*k'}} \leq \mathbf{u^j_{k*k*'}}$ $\forall k' \in \{1,2\}$.
- A situation $(a^i_{kp}, a^j_{k'p})$ is Pareto optimal if any other situation can only increase an agent pay-off but reduce other agents' pay-offs.

b. Example of two situations: Pareto and Nash equilibrium

		A_j	
		a_1	a_2
A_i	a_1	(2, 2)	(4, 1)
	a_2	(1, 4)	(3, 3)

- (**a**$_2$, **a**$_2$) is a Pareto optimal but not a Nash equilibrium
- (**a**$_1$, **a**$_1$) is a Nash equilibrium but with no optimal payoffs
- In general, the Nash equilibrium situation doesn't guarantee the optimal payoffs for agents.
- In cooperative games, the Pareto situation is preferred.

For merging decisions, we use Thresholds C_1 and C_2 on utility (based on similarity thresholds T_1 and T_2 with the same term).

3.2.2 The particularity of the utility function

The utility is the policy that prescribes the fact of acting (or not) to maximize the gain of the system agents [74, 75]. Its principle is that the contribution to the general utility determines the value of an action. Thus, the utility evaluates an action (or a rule) only according to its consequences. The utility is the value that an agent gets from the performed actions within the system. Otherwise, a utility function is a representation to define agent preferences for suitable actions beyond the value of the explicit consequences of those actions. In other words, the utility is the measure an agent gets from some chosen actions. To model the utility in a cooperative game, it is necessary to use certain indices:

- The satisfaction rate of constraints relating to the agent,
- Personal satisfaction rate,
- Satisfaction neighborhood rate,
- Total satisfaction rate

3.3 Resolution by coalition

3.3.1 Overview of the multi-agent proposed system

An appropriate way to practice a cooperative game is to use the architecture of a multi-agent system. This architecture offers the implementation of shared knowledge, individual skills, and timely action choices. In our proposed agent-based approach, agents are cooperatively organized to work directly on individual pixels or directly on voxels. In the global approach (Fig. 3), we assume that a homogeneous segment can be specified using a cooperative principle based on MAS strengths.

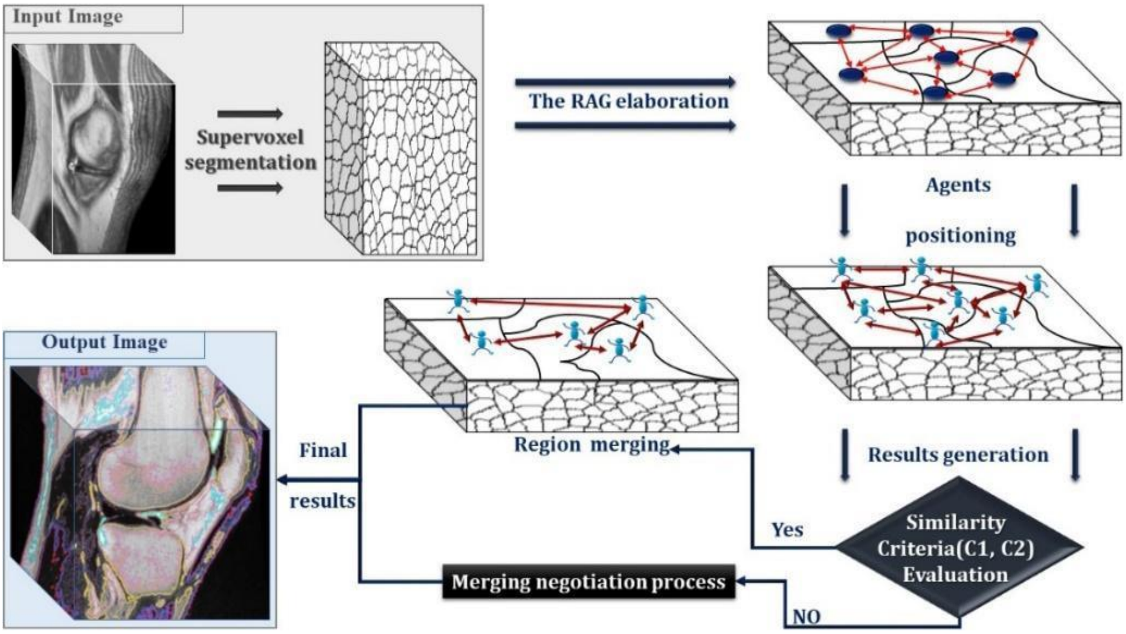


Fig. 3. The global approach

We present a structured vision system architecture as a society of cooperative agents located in the image. In fact, such an architecture is appropriate for ensuring:

- Knowledge and uncertainties integration: the agents provide an attractive abstraction to encapsulate local and intelligent processing. MAS (Multi-Agent System) architectures can be used to organize complex system information which involves several operational and/or descriptive knowledge.
- Local cooperation can be carried out by agents who propagate new constraints locally and information to their neighbors.
- Local adaptations: agents can adapt their behaviors locally according to the context of the image, prior knowledge or any extracted information.

Once placed in a region, the agent launches an investigation to find the adjacent neighbours in order to determine its merging decisions. The decision may vary depending on the following situations: 1) allowed merging, 2) unapproved merging, 3) the merging can occur only with cooperation and negotiation.

3.3.2 General Resolution approach

The connection of the adjacent regions of the RAG leads to the connection of adjacent agents. This indicates that each node of the RAG is examined during the merging process. In fact, in each region, an agent, who perceives his neighbours, is located. The agent's actions can vary during the evolution of his tasks:

- keep his area (R), no merge is possible
- carry out a merge (M) simply or by negotiation (Mn)

The utility function, considered here, quantifies the desire for a merger with a neighbour for an agent. Thus, u_{ij} expresses the desire of the agent A_i to merge with the agent A_j. This utility is based on the similarity of the features of the two regions. Let N_i the set of neighbours of the agent $A_{i.}$ Each agent A_i forms a coalition with its neighbours in N_i to maximize the number of regions to merge. Then, A_i calculates its utilities with its neighbours, orders them in a list L_i $\{(A_j, u_{ij}) / A_j \in N_i\}$ according to decreasing utility and then asks the list L_j of each A_j in N_i. Finally, A_i constitutes 3 sets:

1- Set of solutions where the merger is possible: $L_{ia}= \{(A_j, u_{ij})/ A_j \in N_i, u_{ij} > C_2$ and $u_{ji} > C_2\}$

2- Set of solutions where the merger is probable: $L_{ip}= \{(A_j, u_{ij})/ A_j \in N_i, u_{ij}$ or $u_{ji} < C_2$ and u_{ii} or $u_{ji} > C_1\}$

3- Set of solutions where the merger is impossible: $L_{ir}= \{(A_j, u_{ij})/ A_j \in N_i, u_{ij} < C_1$ and $u_{ji} < C_1\}$

Taking a decision concerning the merger could be accomplished in two steps:

1) Step 1:

- A_i accepts to merge the regions R_i and R_j for A_j / $(A_j, u_{ij}) \in L_{ia}$ after validation of agents A_i and A_j in a Pareto situation.
- A_i rejects the merger of the regions R_i and R_j for A_j / $(A_j, u_{ij}) \in L_{ir}$.

2) Step 2:

A_i negotiates the merger of Regions R_i and R_j A_j / $(A_j, u_{ij}) \in L_{ip}$.

In this case, the agent A_i must revise its utilities at the level of the list L_{ip} in the sense of strengthening his desires or not. To confirm his decision, the agent A_i first forms a group G_i (a L_{ip} subset) as a set of possible merger solutions according to its new utilities, then checks its neighbors for validation. According to the optimal Pareto situations, the agent and the neighbours seek a compromise to validate (positive reassessment) the group G_i or a part of the group. The negotiation procedure is that the agent A_i sends the group to one of his neighbours (choice to be discussed), the neighbour re-evaluates the solutions and constitutes a new solutions' subgroup that he decides to keep, then it chooses, in turn, new neighbour (among the remaining ones in N_i) and sends him the subgroup and so on. Solutions that remain after the last visited neighbour will be taken into account.

To carry out an image segmentation, distinct regions have to be merged. Generally, a "good qualified" merger must satisfy the following criteria:

- All pixels have to be assigned to regions,
- Every pixel has to belong to only one region,
- Each region represents an associated set of pixels,
- Every region must be homogeneous according to a given predicate,

To ensure good control of the system, the region agent has to send an information message to others at the end of the executed task. The treatment of the messages between agents is according to the following diagram (Fig.4):

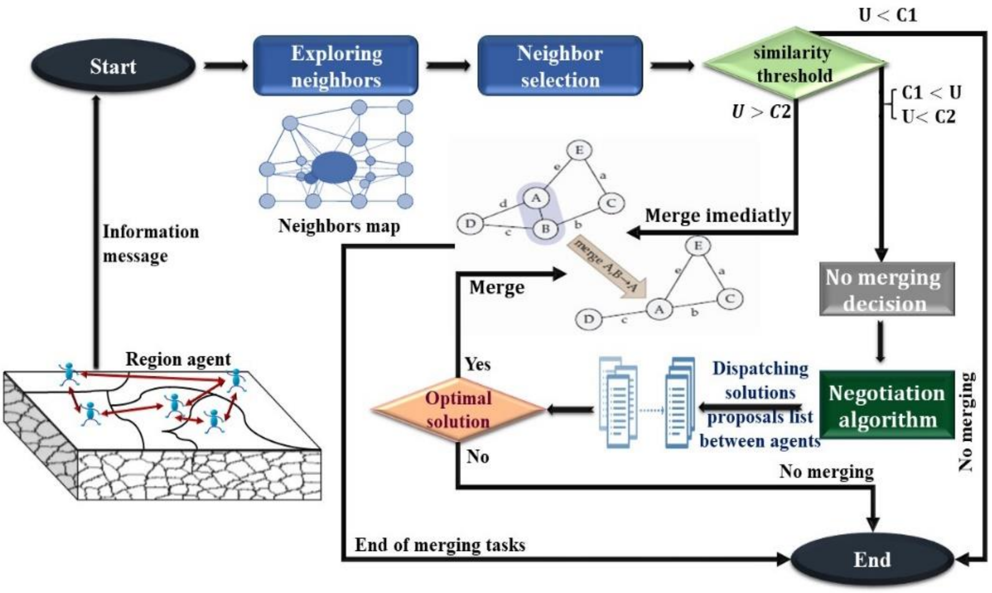


Fig. 4. The proposed approach for region merging process

a. Coalition

The coalition represents a grouping of cooperative agents for the common goal achievement [76]. For this reason, a function called “utility” is associated with different agents forming the coalition. Thus, we seek to model the relations between the different entities. With the utility function, the cooperation problem could be solved in a centralized way according to mathematical models and using the multi-criterion techniques [77]. Our proposal aims to form a coalition of agents that can be included in a segmentation task.

b. Negotiation importance

For an agent, the definition of negotiation must include all the acts to carry on, communication rules, and the resolution of conflicts. Negotiation remains an essential arrangement of interactions managed by a group of agents to reach a mutual agreement according to specific beliefs, goals or plans. Each agent maintains a preference list of the collaborating neighbours. Some elements must be defined to build a model of negotiation by a multi-agent system, for instance, in [78] three fundamental components were identified:

1. Negotiation Protocol collects the rules for organizing negotiation, communication, conversation and decision-making between agents [79]. Since we have adopted a restrictive number of negotiation protocols in our MAS, it is suitable to describe the answers that agents can use.
2. Negotiation Object represents, in our case, all the interactions on which a compromise must be found. The object can contain only one problem (such as merging), while it can cover different problems (conflicts, merging problem, deadlines, penalties, communication rules, etc.).
3. Decision Strategy defines the process approved by the negotiating agents. During the negotiation, each agent develops and exchanges arguments to defend its position or to change it to reach another region. For those who are dissatisfied, or who have not received an agreement simply through the utility test with merger criteria C_1 and C_2, they will form a coalition with agents with the same constraints.

c. Resolution Procedure

The power to negotiate is a fundamental property of an agents’ status. Negotiating allow getting optimal solution who is beneficial for the agent and the wholesale system. For that the resolution procedure contains four essential stages:

1. The negotiation Initialization: Agents have negotiation and reasoning skills. However, the negotiation process includes different stages, the start is done by an initiating agent which is chosen based on some considerations or randomly. A task launching puts on hold other agents who are aiming to start a new negotiation.

2. Sorting the received solutions: The adopted principle is the Pareto optimum. So, each suggestion merging solution received by an agent is then reevaluated (in sense of pay-off) and could be accepted or rejected. Consequently, the agent updates the list and sort it according to his preference.
3. Transmission of the solution: The group of solutions is transmitted between the negotiating agents. Once a group of solutions is identified as being a compromise between the courant agent and the earliest agents. The transmission stops if the courant group is empty.
4. End of negotiation: Agents can check all the received solutions, filter them, approve or reject them. If the last group is not empty the agents adopt the proposed merge suggestions.

d. Practical example

In Figure (Fig.5) an example of adjacent agents is shown to explain the transitions to validate a group of solutions and obtain a final compromise. Clearly, it is also a process of exchanges. Studying the case of a neighbourhood composed of 4 agents negotiating the possibilities of merger (green colour). Let Coa the Coalition containing the agents A_1, A_2, A_3, and A_4. we denote u_{ij} the calculated utilities considering its neighbours. The list for A_1:
L_{1p}= {(A_2, u_{12}), (A_4, u_{14})}. The proposed merging solutions are in G_1= {(A_2, u'_{12}), (A_4, u'_{14})} with new utility values. Let's study the negotiation between first agents:

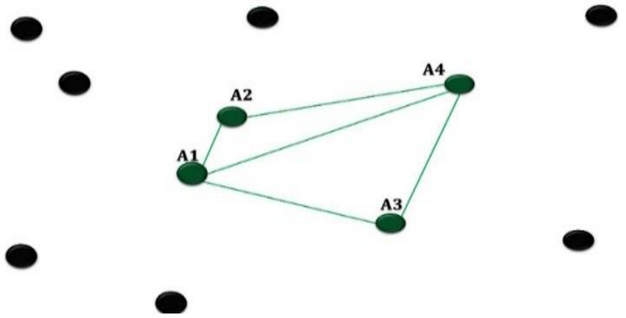


Fig. 5. Example of adjacent agents

If the agent A_1 starts the negotiation process with G_1 the relative proposed solutions to merge with agents A_2 and A_4 after being filtered by new merging criteria.

Table. 1. Example of the possible solutions groups exchange by agents

A_1	-The computed solutions: L_{1p}= {(A_2, u_{12}), (A_4, u_{14})}
	-Order and filtering results with local utility: L_{1p}={(A_2,4), (A_4,9)} -Group of solutions by the Agent A_1: **G_1** = {(A_2,4), (A_4,9)}
A_2	-The computed solutions: **G_1** = {(A_2,4), (A_4,9)}
	-Order and filtering results with local utility: **G_1** = {(A_2,6), (A_4,10)} -Group of solutions by the Agent A_2: **G_2** ={(A_2,6), (A_4,10)}
A_4	-The computed solutions: **G_2** ={(A_2,6), (A_4,10)}
	-Order and filtering results with local utility: **G_2** ={(A_2,1), (A_4,11)} -Group of solutions by the Agent A_4: **G_4** ={(A_4,11)}
A_3	-The computed solutions with local utility: **G_4** ={(A_4,11)}
	-Order and filtering results with local utility: **G_4** ={(A_4,8)} -Group of solutions by the Agent A_3: **G_3** {(A_4,8)}

Table. 1 represents the group transfer between these agents. The possible proposed solutions are exchanged through the agent communication to obtain a possible common compromise which is the Pareto optimum. The agent A_1 initiates the negotiation. It sorts all the acceptable solutions in the group **G. A_1** then sends it to the next agent A_2. The group **G_1** contains all the acceptable merging solutions for the agent A_1 because they correspond to a situation as or more satisfactory than its initial reference situation. In the same way, the A_2 agent also looks for its first acceptable solutions, he has its own solutions that can evaluate, and agent A_1's ones so he has to make an order for these solutions in order to form his own group **G_2** that can send to the following agents which would process on the same way. At the end of this negotiation, all agents (A_1, A_2, A_3, A_4) have a compromise for (A_4,8) so they agree with the merge of R_1 and R_4 regions. The figure Fig.6 explains the group transfer between agents A_1 and A_2, A_2 and A_4 and finally A_4 and A_3.

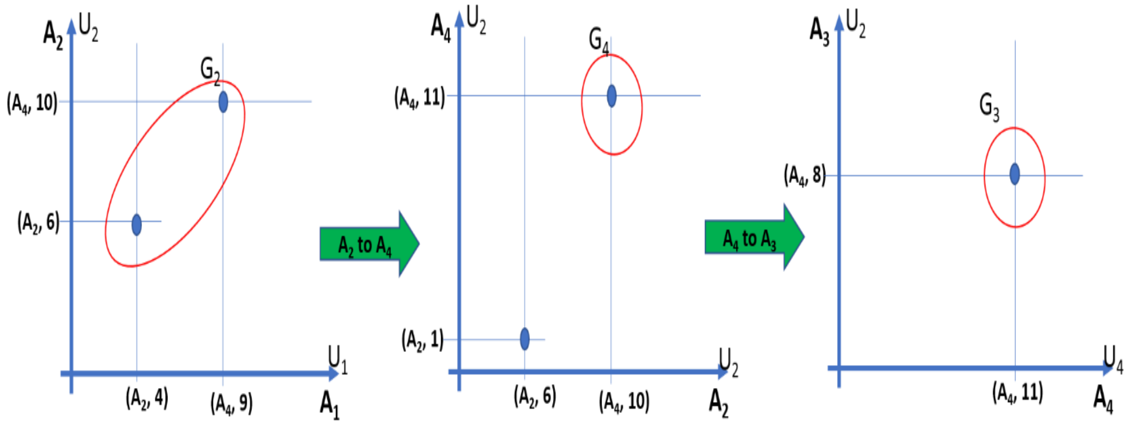


Fig. 6. Graph representing the possible lists of utilities sent by agents

The common objective of cooperating agents is to find a satisfactory solution for all. This example illustrates a small part of the negotiation exchanges between agents in our system.

4 Experimental Results

In this section, we evaluate the proposed architecture by applying the previous steps in python programming and testing different images. To evaluate the quality of a compromise, we introduce here different metrics. Furthermore, our proposed approach for region merging is based on two stages:

- Stage 1: the mechanism of merging is based on similarities according to the threshold criteria C_1 and C_2, then the region map update.
- Stage 2: Since we have a new graph after the region map update, we can use a cooperative approach to negotiate the optimal solution.

4.1 Stage 1: the basic mechanism of the merging process

4.1.1 Determination of similarity criteria

The measurement of the similarity between the pixels along the regions defines the proposed criteria. The choice of these threshold criteria C_1 and C_2 requires specific knowledge and must be done with caution. In this stage, the merging Criteria are the quantitative variables: C_1= 0.06 and C_2=0.824. For similarities less than C_1, the merge is rejected, for these greater than C_2 merge accepted, and for utilities between C_1 and C_2, there is a problem to take the accurate decision. In this case, the region map is updated, a second RAG is obtained after the completed region merging, and the second stage is needed to find the appropriate fusion solution.

4.1.2 RAG construction

One of the strong points of (RAGs) is the spatial view of the image [80]. In fact, the RAG allows us to have a simple view of the connectivity figuring on the image. In order to test the robustness of our method, we carried out experimental data from the Patient Contributed Image Repository [81] We opted for these variant medical MRI images to have the opportunity to compare the results obtained by segmenting images of different cases:

Table 2. The tested datasets specifications

Medical datasets	Number of slices	Type
Knee	124	MR
Shoulder	69	MR
Heart	237	CT

Where: MR refers to Magnetic resonance images and CT refers to the computed tomography scan.

After testing these datasets, different quantitative and qualitative results were obtained. For that, to simplify the presentation of visual results of each of the three datasets, we have chosen randomly a slice then we compared the progression of merging during the execution of our method.

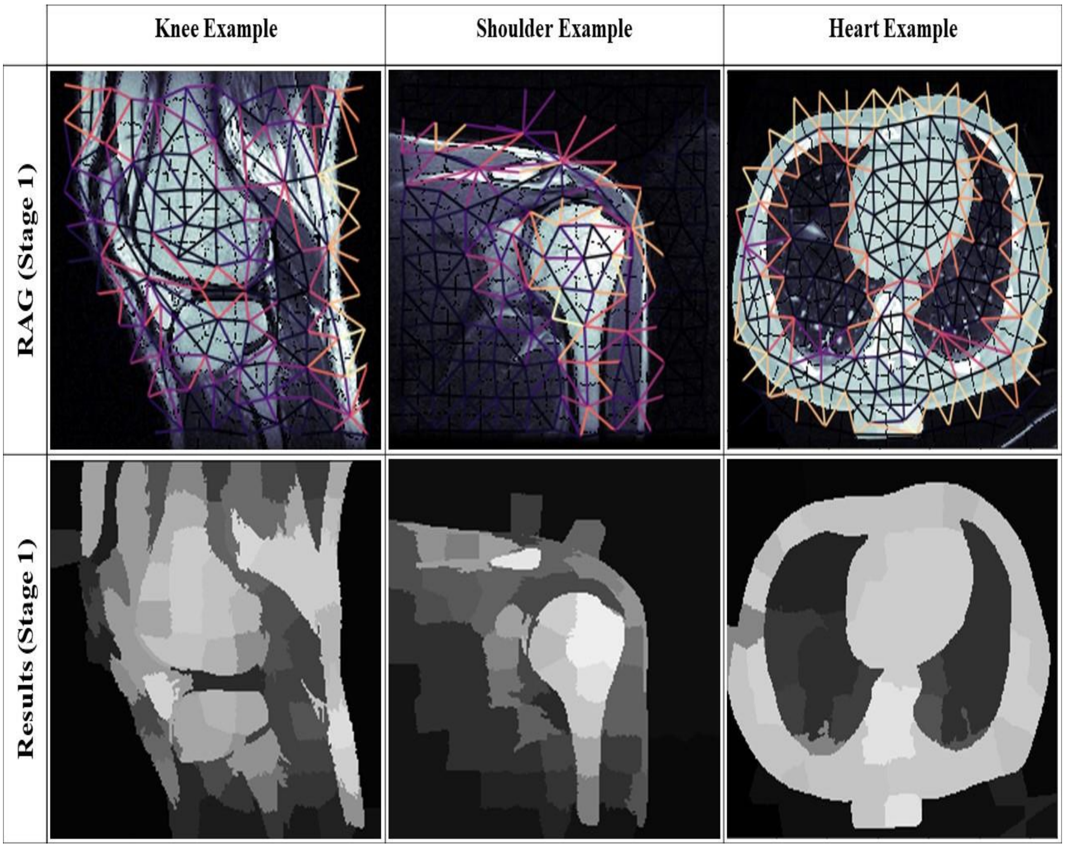


Fig. 7. Stage 1 results crossing the three different datasets

To understand the results of this step, the graph of the chosen examples is displayed in Fig. 7(RAGs-stage1). We can notice easily that the RAGs in this stage contain a huge number of start regions to be segmented. The main idea of our approach is to obtain leading results by merging the numerous regions we have at the beginning. It is quite remarkable, in Fig. 7(results-stage1), there are many resulting regions even more than the observed objects on the studied slices. At this point, we notice the necessity to move to the second stage which is based on a multi-agent system rather than the work of a single agent.

4.1.3 The expression of the used Similarity

The similarity between adjacent regions R_i and R_j can be calculated using weights on RAG arcs and variance measures. Thus, the weights are defined as follow:

$$W_{ij} = \exp\left(-\beta \left(\frac{|h_i - h_j|}{dist(i,j)}\right)^2\right) \qquad (1)$$

Where i and j are two adjacent nodes (regions) of the graph, h_i and h_j are the histograms of R_i and R_j, dist (i, j) is the distance function between the two neighbouring regions, and β is the free integer parameter referring the Euclidean distance.

We can form the similarity $\mathbf{S_{ij}}$ by using a linear combination as follows:

$$S_{ij} = V_{ij} + W_{ij} \qquad (2)$$

Where:

- V_{ij} is the variance measure between the regions i and j.

4.1.4 The Merging Task:

Since the two criteria, C_1 and C_2 need to be checked in order to decide the merge possibility. All the nodes' similarities of the RAG are calculated and stored for each graph layer. Accordingly, an iterative task can be managed regarding the following merging rules:

- If $S_{ij} > C_2$: the merge is accepted,
- If $S_{ij} < C_1$: the merge is rejected,
- Otherwise: no-decision.

The last situation (no decision) can lead to a loss of data from the regions and very possible to the confusion of detection of the good segments. Consequently, for the remaining regions without decision, we adopt MAS decisions by negotiation in stage 2.

4.2 Stage 2: the optimal solution negotiation

4.2.1 MAS adopted architecture

We suggest a merging negotiation framework based on the Multi-Agent System (MAS). This MAS (Fig. 8) consists of heterogeneous types of agents implementing different functionalities in the environment (image). Our system is based on agent applications in compliance with the specifications [83] for an inter-operable intelligent region merging system. FIPA offers, in addition to an agent, a communication language, a specification of the essential agents for the system management, the used ontologies, the behaviour of the agents in the different situations, as well as the compatibility with the platforms of MAS implementation including [84] in Java and [85] in python. The system ontology includes knowledge about the image segmentation, region merging, and game theory concepts. Functional agents can belong to two types: The environment agent and the region agents.

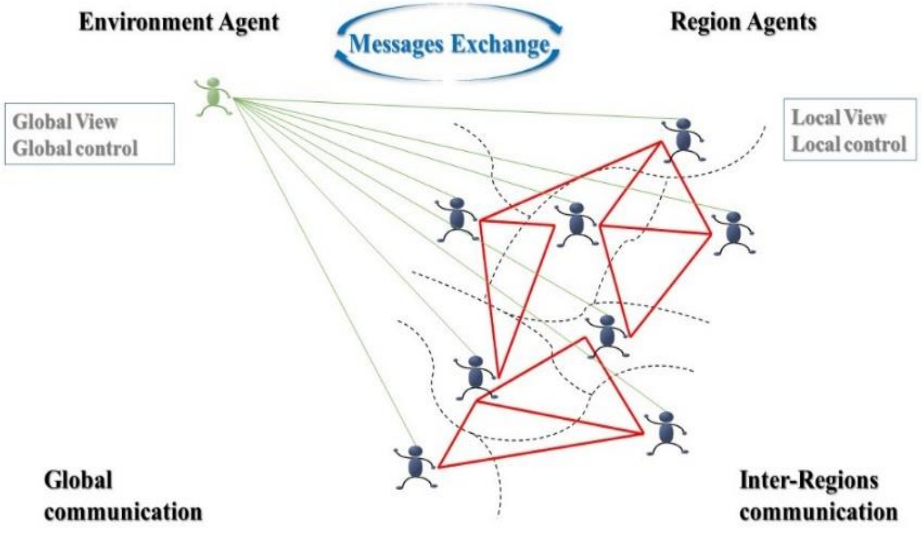


Fig. 8. The cooperative MAS

During this stage, agents' society relies on the Environment Agent and the Region Agents who can exchange and develop the shared information.

a. Environment agent

The environment agent acts globally, with the full ability to reach global system resources, especially the possibility of the exchanged information organization, the results collection, the task distribution, and negotiation management. The Environment Agent has a general vision of the executed tasks. The information it has on regions and agents allows it to properly manage the negotiation process by initializing region agents, according to the results of the functions called at the heart of the process. Its role as the superior in the hierarchy gives him several opportunities using special functions:

- Request update of the list (): allows to rank a list of agents' proposals, which can be updated after each negotiation.
- Request a negotiation (): Triggers a negotiation, so that agents can cooperate and negotiate messages to reach a compromise to resolve the faced problem.
- Approve a merger (): Allows the approval of merger proposals after validation of the results of the negotiation process.

Like all other agents, the environment agent exchanges the messages either to inform others about system changes, the most remaining ones are:

- The initialization of Region agent: The environment agent is in charge of positioning the region agents in the image to be processed and, their initialization to work so agents are listening for new messages.
- The end of negotiation lists construction to a region agent: A negotiation list is formed by the positioned agents.
- The end of negotiation: After agent negotiations, the obtained results are transferred to the environment agent to be approved.

b. Region agent behavior

The region agent is a local actor that performs tasks serving the merge process such as communicating with other region agents, satisfying similarity criteria, and negotiating tasks. These agents are cooperating to improve the merging situation in the system by resolving potential conflicts through communication processes, agent knowledge, and mutual negotiation decisions. In this context each one of the agents has functions allowing it to manage its tasks:

- Do update (): update the list of negotiations.
- Do negotiation (): contribute in a negotiation, by exchanging messages to reach an optimal solution
- Provide merging results (): Send the obtained results to the environment agent so he can approve them or not according to other agents′ results.

Having its own skills, and having access to necessary ontologies, the region agent must communicate through messages to inform others about his progression in the system:

- Behaviors ordering: Some functions need a special behavior from the agent or its neighbors.
- Behaviors execution: after defining the action to proceed the agent informs others about the reaction that he will execute.
- Messages handling: the agent must check his messages to be informed about any changes.

4.2.2 The adopted agent utility

Utility [82, 75] is the policy that prescribes the fact of acting (or not) to maximize the gain of the system agents. Its principle is that the contribution to the general utility determines the value of an action. Thus, the utility evaluates an action (or a rule) only according to its consequences. The utility is the value that an agent gets from the performed actions within the system. As mentioned in section 3.3.2, The utility U_{ij} expresses the desire of the Region agent A_i to merge with the Region agent A_j. We calculate this utility using the similarity defined above (S_{ij} in (2)) and agent preferences depending on local information.

$$U_{ij} = S_{ij} * Nbv_i (Lp_i - Lp_j)/E_{ij} \qquad (3)$$

Where:

- S_{ij}: the similarity computed using the equation (2)
- The Nbv_i denotes the number of neighbouring regions for the region R_i
- E_{ij} is the distance average of the two regions R_i and R_j
- Lp_i and Lp_j are respectively the iso-segments describing the distance ranges between the two regions R_i and R_j

4.2.3 The obtained merging results

The merging results of the first stage are used as the input of the second. The negotiation is based on the tests of criteria for decision making. The merge strategy involves an order for the execution of the task. The merging proposals are transferred between the agents, evaluated, and updated until an optimal solution satisfying all the system (Final results in Fig. 9). In our approach, agents interact to accomplish the common objectives and resolve the merger decision issues.

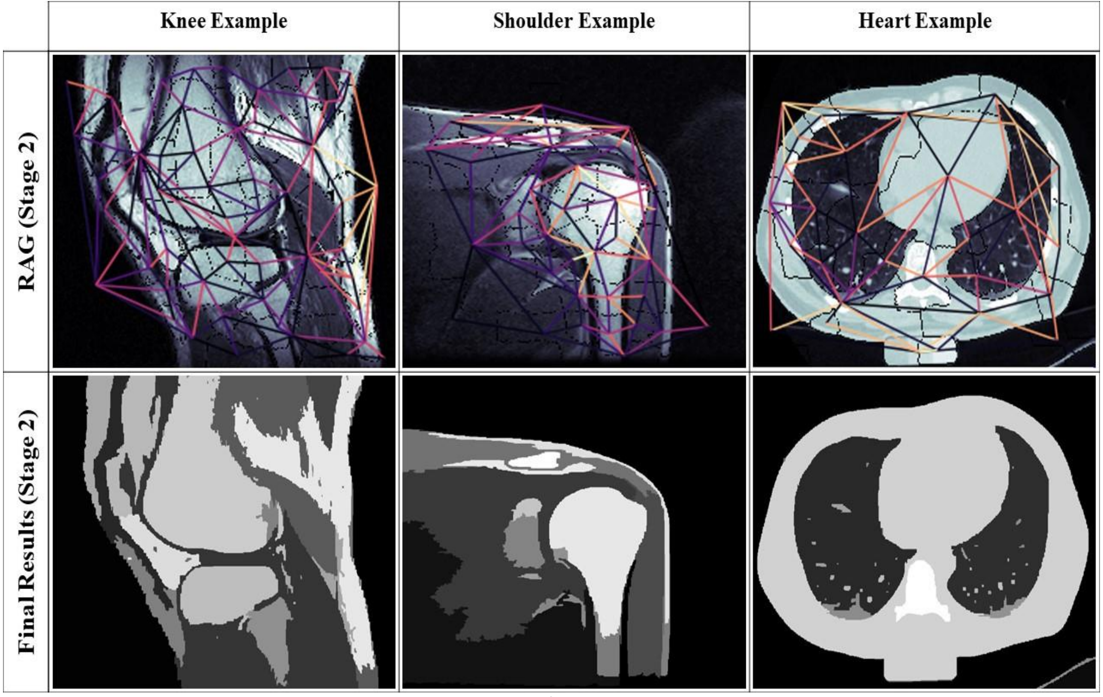


Fig. 9. Results of the second stage

5 Discussion

In the previous sections, we presented our proposed method to resolve a region merging problem. In this section, we discuss the obtained results. Thanks to the multi-agent architecture better fusion quality results were found. Our approach is based on both concepts: agent's negotiation and similarity criteria. Indeed, to evaluate the obtained results we performed several tests. First, the RAGs of the final results as shown in (Fig.10) ensure that we had a tiny number of regions comparing with the RAGs of the first stage.

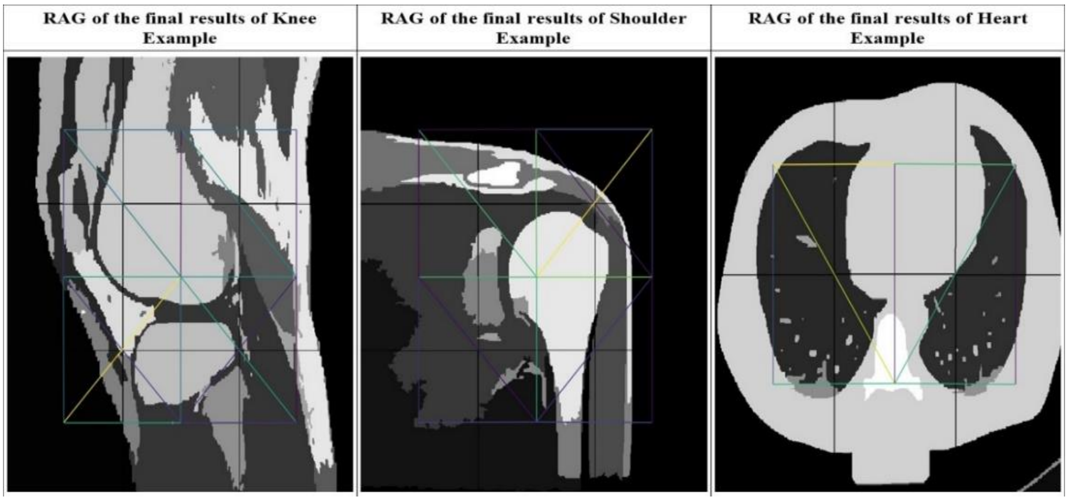

RAG of the final results of Knee Example
RAG of the final results of Shoulder Example
RAG of the final results of Heart Example

Fig. 10. RAG of the final result indicating fewer regions compared to the starting RAG

The introduced approach in this paper can be implemented to segment different sets of images. The image can be divided into regions and the characteristics of the images can be extracted automatically. In fact, our tests have shown promising results.

In order to show the effectiveness of the proposed method, we evaluated different medical datasets containing Knee, shoulder and heart images. In this part, we quantitatively compare the results with those proven by other methods. First, we tested the results of various methods (including ours) to evaluate their pertinence. Secondly, we classed and compared the methods' outcomes. Finally, we infer the method that offers better results. For that, we used the SSIM [86, 87], F-Measure [88, 89], Dice [90, 91, 92], sensitivity [93], specificity [94] and Jaccard [95, 96, 97] indices:

- SSIM: Structural SIMilarity was developed to measure the visual quality of an image compared to a reference. The idea of SSIM is to measure the similarity of structure between the two images.
- F-measure: Quality index for the results of an image processing that combines the two indices of recall and precision.
- Dice: measures the similarity of the found objects by calculating the size of the overlap of the two segments according to the total size of the two objects.
- Sensitivity: measure the proportion of real positive pixels correctly identified.
- Specificity: refers to the ability of the test to correctly detect pixels that meet the same similarity criteria.
- Jaccard: Measure adopted to evaluate the similarity of a given set. More precisely, it is the ratio between the size of their intersection and the size of their union.

Considering the theoretical properties of possible segmentation approaches into a practical system, our objective in this paper is to present an optimal region-merging approach. However, the efficiency of the proposed method is compared with other methods of image segmentation: Region growing [18], MAS for s Markov random fields (MRF) segmentation [98], and Nash-game approach [51]. To accomplish these tests, the different datasets were tested with the same measures. We chose to test different paradigms to compare the effectiveness of the obtained results by our approach.

Table. 3. The results from methods comparison in segmentation of Knee images dataset

Methods	*SSIM%*	*F-measure%*	*Dice %*	*Sensitivity %*	*Specificity%*	*Jaccard %*
Region growing	74,03	78,56	70,89	67,85	71,46	76,25
MRF segmentation	69,47	77,86	76,10	67;11	74,32	80,76
Nash-game approach	76,46	80,99	73,32	70,28	73,89	78,68
The proposed approach	**88,78**	**90,78**	**86,25**	**84,86**	**87,34**	**92,29**

Table. 4. The results from methods comparison in segmentation of shoulder images dataset

Methods	*SSIM%*	*F-measure%*	*Dice %*	*Sensitivity %*	*Specificity%*	*Jaccard %*
Region growing	***78,67***	***75,57***	***75,67***	***75,59***	***67,12***	***76,10***
MRF segmentation	***67,69***	***75,73***	***69,23***	***66,85***	***71,83***	***75,06***
Nash-game approach	***81,10***	***78,01***	***78,23***	***78,02***	***69,55***	***78,53***
The proposed approach	**89,90**	**91,36**	**90,29**	**88,69**	**85,47**	**90,11**

Table. 5. The results from methods comparison in segmentation of heart images dataset

Methods	*SSIM%*	*F-measure%*	*Dice %*	*Sensitivity %*	*Specificity%*	*Jaccard %*
Region growing	67,23	68,05	70,48	69,83	67,77	71,12
MRF segmentation	65,60	67,73	63,46	65,21	62,67	75,06
Nash-game approach	74,55	75,90	77,30	74,17	71,00	78,23
The proposed approach	**88,69**	**89,73**	**91,07**	**87,91**	**81,66**	**89,00**

To evaluate the effectiveness of the tested methods, we used the same datasets which are open to the scientific researchers on (PCIR) [99], then we carried out a numerical analysis of the qualitative and quantitative results. In particular, we found that segmentation methods based on region growing, and Markov random fields (MRF) provide modest results comparing to other approaches.

Relevant conclusions can be drawn from Tables (3,4 and 5) which compare the results of our proposed approach with the other methods. The table shows a high overall rating and the results from our approach show comparable performances. The other methods obviously have varied performances depending on the tested cases and the features of the images. The results show the validity of our method based on the cooperative MAS for region merging. Indeed, we realized that the performance is always proportional to the calculated utility.

By examining the segmented images, we could notice variable efficiency values from one slice to another, because the high or low slices can have intensities or can contain objects very different from those existing in the middle. Although the comparisons with the obtained segmentation results show evidence that our cooperative MAS produces more efficient segmentation. We noticed that our results became more relevant when we improved the utility in (3). To achieve this, we added another criterion α which is Levine and Nazif intra-region uniformity criterion [100]. This criterion calculates the sum of the contrasts of the regions weighted by their surface So, formula 3 would be updated as follows:

$$U_{ij} = \alpha * S_{ij} + Nbv_i * (Lp_j - Lp_i) / E_{ij} \qquad (4)$$

This allowed us to have the outcomes presented in Table 6:

Table .6. Results of the proposed approach after utility changes for the studied datasets

Methods	*SSIM%*	*F-measure%*	*Dice %*	*Sensitivity %*	*Specificity%*	*Jaccard %*
Knee dataset	91,31	93,27	88,78	87,19	89,77	94,72
Shoulder dataset	92,43	93,89	92,82	91,32	87,91	92,64
Heart dataset	91,11	92,15	93,01	90,33	84,49	91,47

The performance of our method has been increased, as shown in table 6, thanks to the cooperative work of the MAS. However, to obtain homogeneous regions, we need an imaging expert in order to fix the criteria of the merge. Yet, according to our tests, it is a promising idea to start directly through the negotiation stage without leading two successive stages. This can be done by placing the multi-agent system on the image, each agent will progress by calculating its utility function U and communicating its results to the environment agent. Then the environment agent will test each utility according to the similarity criteria C1 and C2 to make a merging decision. After this step, we found that the stronger the utility function, the higher the efficiency of the system. By improving the utility function, we also show that the method can easily be extended by integrating other particularities to improve its performance.

Different challenges have been met to ensure an optimal solution. First, the proposed image segmentation approaches were generally based on the Nash equilibrium to deal with agents' communication. However, in a non-cooperative game, the optimality can't be achieved by only using the Nash equilibrium. For that reason, the negotiation between agents remains a strong solution especially using the Pareto optimal to reach the optimal solution. Second, the size of the proposed prototype and the number of the exchanged messages during the negotiation slow down the execution. But this problem was overcome by adopting lighter agent architectures with certain tasks parallelization. Moreover, it's crystal clear that the number of the negotiated regions is often smaller than the first Superpixels/Supervoxels outcome thanks to agents' work. Finally, the non-compromise case can affect some real mergers, for that, the agent payoffs (utility) must be carefully chosen to improve the quality of the obtained results.

6 Conclusion

This article proposes a new cooperative segmentation considering the exchanged information by the system agents based on the Pareto optimal to improve the performances and to reduce the calculation cost of the MAS. Our studies address the issues of image segmentation to produce a generalized approach. We defined a relevant approach that leads to pertinent results. We believe it is an opportunity to perceive the effective functioning of leading principles of game theory and MAS cooperation. Thus, these principles are adopted in a concrete situation of image segmentation using region merging techniques. The purpose of our study was to study the different combinations of image segmentation techniques. In our case, the adopted principle was the Pareto Optimum which favours the optimum gain.

We have studied and proved that a well-defined approach based on a cooperative Multi-Agent System can improve the continuity of the researches in medical image segmentation. We have also examined that the agents' behaviour can influence the results significantly. Since conflicts can occur while agents are executing their tasks the negotiation remains an important operation to ensure the effectiveness of the segmentation system. Consequently, the main difficulty we faced was the specification of consistent merge criteria. In order to accomplish a satisfactory segmentation, we believe image experts' opinions would help in eliminating ambiguities in all the criteria.

The results have shown that the proposed method is effective and robust in the cooperative segmentation of medical datasets. Eventually, our method offers competitive results with machine learning algorithms that are subject to current trends in image processing.

References

[1] S.Kannan, Vairaprakash Gurusamy, and G.Nalini. Review on image segmentation techniques. The Regional Telecommunications Independent Review Committee (RTIRC), 2014.

[2] Andrew E. Johnson, and Martial Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. IEEE Transactions on Pattern Analysis & Machine Intelligence, 21(05) : 433-449, 1999.

[3] Yan Yan, Elisa Ricci, Gaowen Liu, and Nicu Sebe. Egocentric Daily Activity Recognition via Multitask Clustering. IEEE Transactions on Image Processing, 24(10): 2984 - 2995, 2015.

[4] Anton Milan, Konrad Schindler, and Stefan Roth. Multi-Target Tracking by Discrete-Continuous Energy Minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(10): 2054 - 2068, 2016.

[5] Zhihua Xia, Neal N.Xiong, Athanasios V.Vasilakos, and XingmingSun. EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing. Information Sciences, 387: 195-204, 2017.

[6] Thomas M. Lehmann, Claudia Gonner, and Klaus Spitzer. Survey: Interpolation methods in medical image processing. IEEE Transactions on Medical Imaging, 18(11), 1999.

[7] Erik Smistad, Thomas L.Falch, Mohammad Mehdi Bozorgi, Anne C.Elster, and Frank Lindseth. Medical image segmentation on GPUs – A comprehensive review. Medical Image Analysis, 20(1): 1-18, 2015.

[8] Song Zhu, Lidan Wang, and Shukai Duan. Memristive pulse coupled neural network with applications in medical image processing. Neurocomputing, 227: 149-157, 2017.

[9] Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global Contrast Based Salient Region Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(3): 569 - 582, 2015.

[10] Jacques Ferber. Multi-Agent System: An Introduction to Distributed Artificial Intelligence. Harlow: Addison Wesley Longman, ISBN 0-201-36048-9, 1999.

[11] Alvaro L. Bustamante, José M. Molina, and Miguel A. Patricio. A practical approach for active camera coordination based on a fusion-driven multi-agent system. International Journal of Systems Science, 45(4), 2014.

[12] Hsairi L, Ghedira k, Alimi AM, and BenAbdellhafid A. Argumentation Based Negotiation Framework for MAIS-E2 model. Chapter VI in book : Open Information Management: Applications of Interconnectivity and Collaboration, ISBN:978-1-60566-246, 2009.

[13] Camerer and Colin F. Progress in Behavioral Game Theory. Journal of economic perspectives, 11(4): 167-188, 1997.

[14] D. Ary A.Samsura, Erwinvan der Krabben, and A.M.A.van Deemen. A game theory approach to the analysis of land and property development processes. Land Use Policy, 27(2): 564-578, 2010.

[15] Ilhan K. Geckil and Patrick L. Anderson. Applied Game Theory and Strategic Behavior. Chapman and Hall/CRC, DOI: https://doi.org/10.1201/9781584888444, 2016.

[16] Joanna Jaworek-Korjakowska and Pawel Kleczek. Region Adjacency Graph Approach for Acral Melanocytic Lesion Segmentation. Applied Sciences, 8(9), 1430; doi.org/10.3390/app8091430, 2018.

[17] Bindita Chaudhuri, Begüm Demir, Lorenzo Bruzzone, and Subhasis Chaudhuri. Region-Based Retrieval of Remote Sensing Images Using an Unsupervised Graph-Theoretic Approach. IEEE Geoscience and Remote Sensing Letters, 13(7): 987 - 991, 2016.

[18] Xiaoqi Lu, Jianshuai Wu, Xiaoying Ren, Baohua Zhang, and Yinhui Li. The study and application of the improved region growing algorithm for liver segmentation. Optik, 125(9): 2142-2147, 2014.

[19] CH Wei, SY Chen, and X Liu. Mammogram retrieval on similar mass lesions. Computer methods and programs in biomedicine, 106(3): 234-248, 2012.

[20] Szénási and Sándor. Distributed Region Growing Algorithm for Medical Image Segmentation. International Journal of Circuits, Systems, and Signal Processing, 8: 173-181, 2014.

[21] Dilpreet Kaur and Yadwinder Kaur. Various Image Segmentation Techniques: A Review. International Journal of Computer Science and Mobile Computing, 3(5): 809-814. 2014.

[22] Adams R and Bischof L. Seeded region growing[J]. Pattern Analysis and Machine Intelligence. IEEE Transactions on, 16(6): 641-647, 1994.

[23] R Rouhi, M Jafari, S Kasaei, and P Keshavarzian. Benign and malignant breast tumors classification based on region growing and CNN. Expert Systems with Applications, 42: 990–1002, 2015.

[24] R Nock and F Nielsen. Statistic region merging. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26: 1452-1458, 2004.

[25] Ning J, L Zhang, D Zhang, and C Wu. Interactive Image Segmentation by Maximal Similarity-based Region Merging. Pattern Recognition, 43: 445-456, 2010.

[26] F Calderero and F Marques. Region merging techniques using information theory statistical measures. IEEE Transactions on Image Processing, 19(6): 1567-1586, 2010.

[27] F. Calderero and F. Marques. General region merging approaches based on information theory statistical measures. The 15th IEEE International Conference on Image Processing (ICIP), 3016-3019, 2008.

[28] Y Shu, GA Bilodeau, and F Cheriet. Segmentation of laparoscopic images: integrating graph-based segmentation and multistage region merging. The 2nd Canadian Conference on Computer and Robot Vision, 2005.

[29] H Liu, Q Guo, M Xu, and I Shen. Fast image segmentation using region merging with a k-Nearest Neighbor graph. In IEEE Conference on Cybernetics and Intelligent Systems, 179-184, 2008.

[30] B Peng, L Zhang, and J Yang. Iterated Graph Cuts for Image Segmentation. In Asian Conference on Computer Vision, 2009.

[31] Youssef El Merabet, Cyril Meurie, Yassine Ruichek, Abderrahmane Sbihi, and Raja Touahni. Building Roof Segmentation from Aerial Images Using a Line-and Region-Based Watershed Segmentation Technique. Sensors, 15(2): 3172-3203, 2015.

[32] Sebari, I. and He, D. Les approches de segmentation d'image par coopération régions-contours. Revue Télédétec, 7: 499–506. 2007.

[33] Awad M and Nasri A. Satellite image segmentation using Self- Organizing Maps and Fuzzy C-Means. Proceedings of the 2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, UAE, 398–402. 2009.

[34] Chen Q, Zhou C, Luo J, and Ming D. Fast Segmentation of High-Resolution Satellite Images Using Watershed Transform Combined with an Efficient Region Merging Approach. Proceedings of the International Workshop on Combinatorial Image Analysis, 621–630, December 2004.

[35] Y Cherfa, Assia Jaillard, A Cherfa , and Y Kabir. Segmentation coopérative d'images RMN cérébrales : Application à la caractérisation des accidents vasculaires cérébraux. Journées d'études sur l'imagerie médicale JETIM Blida, 2004.

[36] Yitian Zhao, Lavdie Rada, Ke Chen, Simon P. Harding, and Yalin Zheng. Automated Vessel Segmentation Using Infinite Perimeter Active Contour Model with Hybrid Region Information with Application to Retinal Images. IEEE Transactions on Medical Imaging, 34(9): 1797 - 1807, 2015.

[37] Omid Sarrafzadeh and Alireza Mehri Dehnavi. Nucleus and cytoplasm segmentation in microscopic images using K-means clustering and region growing. Adv Biomed Res, 4(174), 2016.

[38] Pim Moeskops, Mitko Veta, Maxime W. Lafarge, Koen A. J. Eppenhof, and Josien P. W. Pluim. Adversarial Training and Dilated Convolutions for Brain MRI Segmentation. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA 2017, ML-CDS 2017. Lecture Notes in Computer Science, vol 10553. Springer, 2017.

[39] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In ECCV 2014: Computer Vision: 818-833, 2014.

[40] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In The Fourth International Conference on 3D Vision (3DV), DOI: 10.1109/3DV.2016.79, 2016.

[41] Thisse and Jacques-François. Théorie des jeux: une introduction. Techno-Science.net. http://www.autzones.com/din6000/textes/semaine04/Thisse%20(2002)%20-%20Jeux.pdf, 2003.

[42] Chaib-draa, B. Chapitre 1: Introduction à la Théorie des Jeux. Laval University, Computer Science & Software Engineering (CSSE) Department, Canada. http://www.damas.ift.ulaval.ca/~coursMAS/ComplementsH10/Intro-TJ.pdf. 2008.

[43] Nash, John F. Equilibrium Points in n-Person Games. Proceedings of the National Academy of Sciences of the United States of America, 36(1): 48-49, 1950.

[44] Sankardas Roy, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Vivek Shandilya, and Qishi Wu. A Survey of Game Theory as Applied to Network Security. In the 43rd Hawaii International Conference on System Sciences, 2010.

[45] E Semsar-Kazerooni and K Khorasani. Multi-agent team cooperation: A game theory approach. Automatica, 45(10): 2205-2213, 2009.

[46] Chakraborty Amit and James S Duncan. Integration of boundary finding and region-based segmentation using game theory. XIVth International Conference on Information Processing in Medical Imaging, 189-200, 1995.

[47] Chakraborty A and Duncan JS. Game-theoretic integration for image segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence, 12-30, 1999.

[48] Elizabeth Cassell, Sumanth Kolar, and Alex Yakushev. Using Game Theory for Image Segmentation. http://www.angelfire.com/electronic2/cacho/machine-vision/ImSeg.pdf, 2007.

[49] Roy K, Suen Ching Y, and Bhattacharya P. Segmentation of Unideal Iris Images Using Game Theory. ICPR: 2844-2847, 2010.

[50] Ibragimov B, Vrtovec T, Likar B, and Pernus F. Segmentation of lung fields by game theory and dynamic programming. In the 4th International Workshop on Pulmonary Image Analysis, PIA 2011: 101-111, 2011.

[51] Kallel, Aboulaich R, Habbal A, and Moakher M. A Nash-game approach to joint image restoration and segmentation. Applied Mathematical Modelling, 38, (11–12): 3038-3053, 2014.

[52] Hossein Aghababaee and Mahmod Reza Sahebi. Game-Theoretic Classification of Polarimetric SAR images. European Journal of Remote Sensing, 48: 33-48, 2015.

[53] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Analysis and Machine Intelligence, 34(11): 2274–2282. 2012.

[54] X Ren and J Malik. Learning a classification model for segmentation. Proceedings Ninth IEEE International Conference on Computer Vision, DOI: 10.1109/ICCV.2003.1238308, 2003.

[55] Felzenszwalb P and Huttenlocher D. Efficient graph-based image segmentation, IJCV, 59(2), 2004.

[56] Liu MY, Tuzel O, Ramalingam S, and Chellappa R. Entropy rate superpixel segmentation. In CVPR, 2011.

[57] R Achanta, A Shaji, K Smith, A Lucchi, P Fua, and S S¨usstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE TPAMI, 2012.

[58] A Levinshtein, A Stere, K Kutulakos, D Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. IEEE TPAMI, 31(12), 2009.

[59] Fan Fana, Yong Ma, Chang Li, Xiaoguang Mei, Jun Huang, and Jiayi Ma. Hyperspectral image denoising with superpixel segmentation and low-rank representation. Information Sciences: 397–398, 2017.

[60] A. Trémeau and P Colantoni. Regions adjacency graph applied to color image segmentation. IEEE Transactions on Image Processing. 9: 735-744, 2000.

[61] WX Kang, QQ Yang, and RR Liang. The Comparative Research on Image Segmentation Algorithms. IEEE Conference on ETCS: 703-707, 2009.

[62] J Maeda, A Kawano, S Yamauchi, Y Suzuki, ARS Marcal, and T Mendonca. Perceptual image segmentation using fuzzy-based hierarchical algorithm and its application to dermoscopy images. Proceedings of the Conference on Soft Computing in Industrial Applications: 66-71, 2008.

[63] M Silveira, JC Nascimento, JS Marques, ARS Marcal, T Mendonca, S Yamauchi, J Maeda, and JRozeira. Comparison of segmentation methods for melanoma diagnosis in dermoscopy images. IEEE Journal of Selected Topics in Signal Processing, 3: 35-45, 2009.

[64] P Felzenszwalb and D Huttenlocher. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision 59(2): 167-181, 2004.

[65] Costas Panagiotakis, Ilias Grinias, and Georgios Tziritas. Natural Image Segmentation based on Tree Equipartition, Bayesian Flooding and Region Merging. IEEE Transactions on Image Processing, 20(8):2276 - 2287, 2011.

[66] Peng Bo, Zhang Lei, and Zhang David. Automatic Image Segmentation by Dynamic Region Merging. IEEE Transactions on Image Processing, 20(12): 3592 - 3605, 2011.

[67] Nock R and F Nielsen. Statistical region merging. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11): 1452–1458, 2004.

[68] Nock R and Nielsen F. Fast Graph Segmentation Based on Statistical Aggregation Phenomena. MVA: 150-153, 2007.

[69] Pascal VOC Dataset. 2010. https://pjreddie.com/projects/pascal-voc-dataset-mirror/.

[70] The Berkeley Segmentation Dataset and Benchmark. BSDS500. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.2007.

[71] Torsello A, Bulo SR, and Pelillo M. Grouping with asymmetric affinities: A game-theoretic perspective. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, DOI: 10.1109/CVPR.2006.130, 2006.

[72] Albarelli A, Bulo SR, Torsello A, and Pelillo M. Matching as a non-cooperative game. In The IEEE 12th International Conference Computer Vision, DOI: 10.1109/ICCV.2009.5459312. 2009.

[73] Albarelli A, Rodolà E, and Torsello A. Imposing Semi-Local geometric constraints for accurate correspondences selection in structure from motion: A Game-Theoretic Perspective. International Journal of Computer Vision, 97 (1): 36-53, 2012.

[74] Edward Hagen and Peter Hammerstein. Game theory and human evolution: A critique of some recent interpretations of experimental games. Theoretical Population Biology, 69: 339–348, 2006.

[75] Beibei Wang, Yongle Wu, and Ray Liu. Game theory for cognitive radio networks: An overview. Computer Networks, 54(14-6): 2537-2561, 2010.

[76] Onn Shehory and Sarit Kraus. Methods for Task Allocation via Agent Coalition Formation. Artificial Intelligence Journal, 101(1–2): 165-200, 1998.

[77] Roy B and Bouyssou D. Multicriterion Assistance with the Decision: Methods and Case. Edition Economica, 1993.

[78] Jennings NR, Faratin P, Lomuscio AR, Parsons S, Wooldridge M, and Sierra C. Automated negotiation: Prospective customers methods and challenges. Intl Newspaper of Group Decision and Negotiation, 10 (2): 199-215, 2001.

[79] Sarit Kraus. Automated Negotiation and Decision Making in Multiagent Environments, ACAI 2001: Multi-Agent Systems and Applications: 150-172, 2001.

[80] Raimondo Schettini. A segmentation algorithm for color images. Pattern Recognition Letters, 14(6): 499-506, 1993.

[81] PCIR. 2016. http://www.pcir.org/index.html.

[82] Edward H Hagen and Peter Hammerstein. Game theory and human evolution: A critique of some recent interpretations of experimental games. Theoretical Population Biology, 69(3); 339-348, 2006.

[83] FIPA. Foundation for Intelligent Physical Agents. 1999. http://www.fipa.org.

[84] JADE. Java Agent DEvelopment Framework. 2000. http://sharon.cselt.it/projects/jade.

[85] Spade. Smart Python Agent Development Environment. 2012. https://pypi.org/project/SPADE/.

[86] Mi Zengzhen. Image quality assessment in multiband DCT domain based on SSIM. Optik, 125(21): 6470-6473, 2014.

[87] Pinghua Zhao, Yanwei Liu, Jinxia Liu, Antonios Argyriou, and Song Ci. SSIM-based error-resilient cross-layer optimization for wireless video streaming. Signal Processing: Image Communication, 40: 36-51, 2016.

[88] Daniel McDonald, Morgan N Price, Julia Goodrich, Eric P Nawrocki, Todd Z DeSantis, Alexander Probst, Gary L Andersen, Rob Knight, and Philip Hugenholtz. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. The ISME Journal, 6: 610–618, 2012.

[89] N. Lazarevic-McManus, J.R. Renno, D. Makris, and G.A. Jones. An object-based comparative methodology for motion detection based on the F-Measure. Computer Vision and Image Understanding, 111(1): 74-85, 2008.

[90] Zou KH, Warfield SK, Bharatha A, Tempany CM, Kaus MR, Haker SJ, Wells WM 3rd, Jolesz FA, and Kikinis R. Statistical validation of image segmentation quality based on a spatial overlap index. Acad Radiol, 11(2): 178-89, 2004.

[91] Klein S, van der Heide UA, Raaymakers BW, Kotte ANTJ, Staring M, and Pluim JPW. Segmentation of the prostate in MR images by atlas matching. ISBI, Arlington, VA: 1300–3, 2007.

[92] Al-Faris AQ, Ngah UK, Isa NAM, and Shuaib IL. MRI breast skin-line segmentation and removal using integration method of level set active contour and morphological thinning algorithms. J Med Sci, 2013.

[93] Loong TW. Understanding sensitivity and specificity with the right side of the brain. BMJ, 327 (7417): 716–9, 2003.

[94] Altman DG and Bland JM. Diagnostic tests 1: Sensitivity and specificity. BMJ. 308 (6943): 1552, 1994.

[95] Keyvan K, Mohammad Javad D, Kamran K, Mohammad Sadegh H, and Kafshgari S. Comparison evaluation of three brain MRI segmentation methods in software tools. Biomedical Engineering (ICBME): 1–4, 2010.

[96] Reddy AR, Prasad EV, and Reddy LSS. Abnormality detection of brain MR image segmentation using iterative conditional mode algorithm. Int J Appl Inform Syst, 5(2): a56–66, 2013.

[97] Reddy AR, Prasad EV, and Reddy LSS. Comparative analysis of brain tumor detection using different segmentation techniques. Int J Comput Appl, 82(14):14–28, 2013.

[98] Kamal E.Melkemi, Mohamed Batouche, and Sebti Foufou. A multiagent system approach for image segmentation using genetic algorithms and extremal optimization heuristics. Pattern Recognition Letters, 27(11): 1230-1238, 2006.

[99] 2016. http://www.pcir.org/index.html.

[100] Levine MD, Nazif AM. Dynamic measurement of computer-generated image segmentation. IEEE Trans. on PAMI, 7(25): 155-164, 1985.

Inteligencia Artificial 22(64) (2019), 123-134
doi: 10.4114/intartf.vol22iss64pp123-134

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

An Enhanced Discrete Bees Algorithms for Resource Constrained Optimization Problems

Mohamed Amine Nemmich[1, A], Fatima Debbat[1, B], Mohamed Slimane[2, C]
[1] Department of Computer Science, University Mustapha Stambouli of Mascara, Algeria
[2] Université de Tours, Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LIFAT), Tours, France
[A] amine.nemmich@gmail.com, [B] debbat_fati@yahoo.fr, [C] mohamed.slimane@univ-tours.fr

Abstract In this paper, we propose two slightly different models based on Bees Algorithm (BA) for the Resource-Constrained Project Scheduling Problem (RCPSP). The studied RCPSP is a NP-hard combinatorial optimization problem which involves resource, precedence, and temporal constraints. It has been applied to many applications. The main objective is to minimize the expected makespan of the project. The proposed Enhanced Discrete Bees Algorithms iteratively solve the RCPSP by utilizing intelligent foraging behaviours of honey bees. The potential solution is represented by the multidimensional bee, where the Activity List representation (AL) is considered. This projection involves using the Serial Schedule Generation Scheme (SSGS) as decoding procedure to construct the active schedules. In addition, the conventional local search of the basic BA is replaced by a neighbouring technique, based on the Swap operator, which takes into account the specificity of the solution space of project scheduling problems and reduces the number of parameters to be tuned. As second variant, the Negative Selection is embedded into our approach to overcome its drawback. The proposed models are tested on well-known benchmark problem instance sets from Project Scheduling Problem Library (PSPLIB) and compared with other approaches from the literature. The promising computational results reveal the effectiveness of the proposed approaches for solving the RCPSP problems of various scales.

Keywords: Optimization, Project scheduling, Resource-constraints, Bees Algorithm, Serial Schedule Generation Scheme, Activity list representation

1 Introduction

Constraint optimization forms an important part of many problems in engineering and industry. Most of the real world optimization problems have constraints of different types which modify the shape of search space. Engineering design, VLSI design, structural optimization, economics, allocation and location problems are just a few examples of fields in which constrained optimization problems are encountered. As constrained optimization problems are difficult to solve due to the presence of various types of constraints (in the form of equalities or inequalities) and their interrelationship between the objective functions [1].

The Resource constrained project scheduling problem (RCPSP) is a well-known computationally intractable combinatorial problem with considerable practical relevance. A vast range of applications in logistics management, manufacturing operations and project management can be modeled as RCPSP such as production planning, scheduling of port operations, school and university timetabling, scheduling of port operations, sports league scheduling, medical research scheduling, assembly shop scheduling, software project, network packet switching and grid computing [2]. The objective of the RCPSP is to schedule the activities in order to minimize the project completion time (makespan) against resource and precedence constraints [3].

Until now, many investigations on RCPSPs have made remarkable progress. Several exact (or deterministic) approaches have been explored firstly such as branch-and-bound procedures, lower bounding, linear programming, mathematical formulations and constraint programming [4]. These procedures are typically used to solve small-sized instances since the execution time needed is impractical when the number of activities increases

ISSN: 1137-3601 (print), 1988-3064 (on-line)

[5]. In addition, constructive heuristics and meta-heuristic methods have been proposed to obtain near-optimum solutions in reasonable amount of time. The constructive heuristics consider two main components, namely the Priority Rule and Schedule Generation Scheme (SGS) [6,7]. Many studies solve the RCPSP with the use of meta-heuristics methods such as Simulated Annealing (SA) [8], Tabu Search (TS) [9], Genetic Algorithm (GA) [10,11,12], Ant Colony Optimization (ACO) [13], Particle Swarm Optimization (PSO) [3,7,14,15], Artificial Bee Colony (ABC) [5,16, 17], Bee Swarm Optimization (BSO) [17], Artificial Immune Systems (AIS) [18], Scatter Search (SS) [19], Multi-Agent Optimization (MAO) [20], etc.

One of the most recent and effective optimization meta-heuristic is the Bees Algorithm (BA). It imitates the foraging behavior of honeybee in nature to solve optimization problems [21]. In this investigation, two modified versions of BA are used and adapted in order to solve the RCPSP.

The rest of this paper is organized as follows. Section 2 provides a description of the RCPSP problem. Section 3 presents the new adaptation of the Bees Algorithm for solving the considered problem. Computational experience and comparative analysis are given in Section 4. Finally, section 5 concludes this work.

2 Description of RCPSP Problem

In this section, the classic single-mode RCPSP is discussed. It is strongly NP-hard [22]. It can be defined as follows. A project involves a set of activities $V = \{0, 1,..., n, n+1\}$ to be scheduled subject to both temporal (precedence) and resource constraints, where each activity has to be processed without preemption in order to complete the project. Each activity j requires a known processing duration d_j ($j = 0,...,n+1$) and consumes $r_{j,k}$ units of resource k for all resource types $k = 1, . . . , K$ during each period of its duration [6,16]. Each type of resource $k \in K$ has a limited capacity R_k that cannot be exceeded in any time period. A resource can be re-allocated to other activities when that resource is released from finished activities. Let $A(t)$ is the activities set being processed at time instance t. Each activity has precedence constraints. Let P_j be the set of immediate predecessors of activity j. Precedence constraints force each activity j to be scheduled after all predecessor activities P_j are completed [15]. By convention the dummy (pseudo) activities 0 and $n + 1$ refer to the start and the end of schedule (i.e. project), respectively. Activity 0 is the source that has no predecessor while activity ($n + 1$) has no successor. In addition, these activities have no processing duration and have no resource requirements for any type of resources. The goal of RCPSP is to find optimal schedule with minimal total project duration time [15].

An example of resource constrained project scheduling problem is given in Figure 1. In the diagram, the activities are represented as nodes and the precedences by directed arcs. The project consists of $n = 6$ non-dummy activities which have to be scheduled, consuming $K = 2$ type of renewable resources with a capacity of 4 and 2 units respectively.

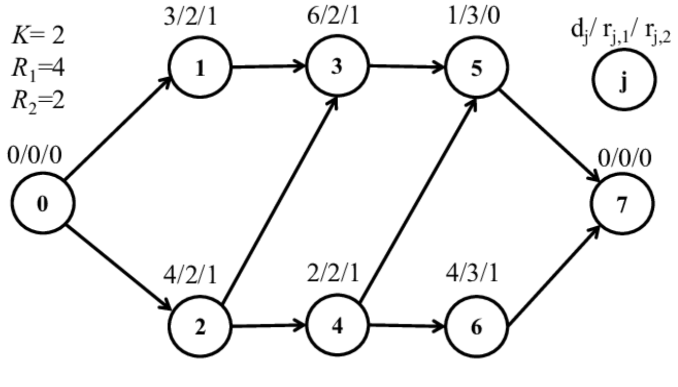


Figure 1: A project example for RCPSP

Let F_j is the finish time of activity j, then a schedule can be represented by a vector of finish times (F_1,...,F_m,...,F_{n+1}). Subsequently, the conceptual model of the RCPSP can be defined as follows [6,16]:

$$\min F_{n+1} \tag{1}$$

$$F_h \leq F_j - d_j \qquad j = 1,\dots,n+1; h \in P_j \tag{2}$$

$$\sum\nolimits_{j \in A(t)} r_{j,k} \leq R_k \qquad k \in K; t \geq 0, \tag{3}$$

$$F_j \geq 0 \qquad j = 1,\dots,n+1. \tag{4}$$

In the RCPSP, the main objective is to identify a feasible schedule that would minimize a project's makespan (Equation1), i.e. the finish time of the last activity processed F_{n+1}, taking into consideration the precedence (Equation 2) and resource limitation (Equation 3) constraints. Equation (4) illustrates the constraint of the decision variables [6,16].

A feasible schedule, of the above problem is represented in Figure 2 with an optimal makespan of 15.

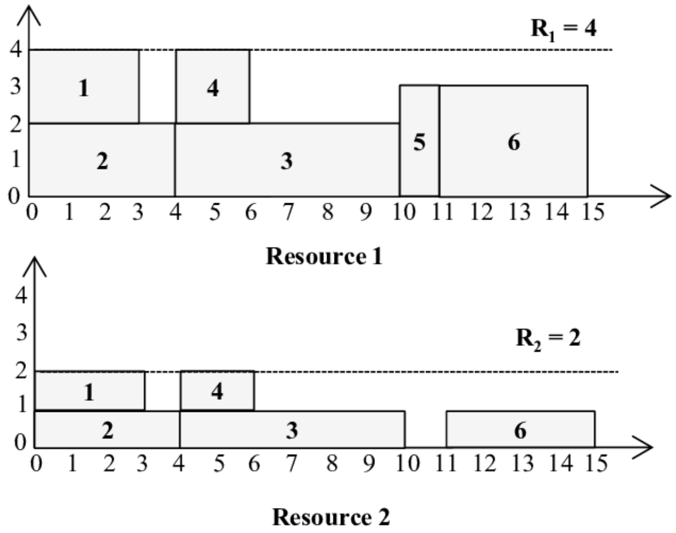


Figure 2: A feasible schedule of the activities

3 Proposed Enhanced Discrete Bees Algorithms for RCPSP

The Bees Algorithm (BA) is a population-based optimization meta-heuristic that mimics the natural foraging behavior of bees. BA was initially designed to solve continuous optimization problems [21]. However, the original BA can be modified and adapted for solving combinatorial problems, such as RCPSP.

Two slightly different algorithms are proposed to solve resource constrained project scheduling problem (RCPSP).

3.1 The First Enhanced Discrete Bees Algorithm (EDBA1)

In the proposed Enhanced Discrete Bees Algorithm (EDBA1), each bee represents a feasible solution for the considered problem, while each activity in a schedule is represented as one component of a bee in the BA process. The representation and decoding process for RCPSP plays a key role in the performance of an approach. In this work, the permutation-based (or the activity list, (AL)) representation was adopted to model solutions of the problem [23], hence the search space is a set of permutations. Each solution is a 1 × n + 2 activities vector including dummy activities. The index of a vector determines the order (i.e. relative priority) of an activity among the others. Therefore, a solution (i.e. activity list) is a permutation of activities such that every activity comes in the list before all activities that are its successors in precedence relation. Values are not repeated in the permutations. In this representation, each activity must have a higher index than each of its predecessors. As an example, consider the feasible schedule of the RCPSP problem described in Figure 1. The solution corresponding is $\lambda = [0, 2, 1, 3, 4, 5, 6, 7]$.

A given solution needs to be decoded and evaluated, but firstly, the solution must be feasible. To do so, the Schedule Generation Schemes (SGS) [24] are generally applied for generating an active schedule from the solution, based on the resource availability and precedence constraints. The SGS is the core decoding procedures for the RCPSP. The basic division distinguishes serial (SSGS) and parallel (PSGS) scheme [6]. In this work, the decoding Serial SGS (SSGS) is adopted, since it always leads to active (i.e. feasible) schedules meaning that using this procedure one is guaranteed to be able to generate an optimal schedule [6]. In addition, the SSGS can be used directly as the decoding procedure for the activity list to obtain the schedule without additional modifications. Kolisch [6] showed that the parallel scheme does not generally perform better than the serial scheme.

The SSGS for Activity List starts from zero to build a schedule by stepwise improvements. The SSGS performs $g = 1,..., n$ iterations, where n is the number of activities. There are two sets of activity in each g state. The scheduled set S_g comprises all the activities which have been already scheduled, and the eligible or decision set D_g, which includes the activities not scheduled for which all predecessor are scheduled. In each iteration one activity j is selected from the decision set D_g and scheduled at it earliest precedence and resource feasible completion time. Next, the decision set of eligible activities and the resource profiles of partial schedule are updated [23]. The scheme terminates on iteration n when all n non dummy activities are scheduled. An application of the SSGS can be recorded by a list $\lambda = [j_1, j_2, ... ,j_n]$, where j_g represents the activity which is selected and scheduled in iteration g. This list is precedence feasible. The makespan (i.e. project duration) of the solution is given by the maximum completion time of all predecessor activities of activity n+1. Due to the lack of space, SSGS pseudo code is omitted but can be found in [24].

In the proposed approach, each bee represents a solution (i.e. schedule) in a combinatorial space which can be presented as permutation; therefore the search space is a set of permutations. These permutations are placed in the space relating to the order of their elements. Movement in the search space is achieved by changing order of the elements.

The proposed EDBA1 approach starts working with the *ns* scout bees being placed randomly across the search space to generate early solutions, which in fact are lists of activities. The discrete uniform distribution technique is used for randomly generating the solutions during this initialisation in which all possible values have equal probabilities. In order to get feasible solutions in the initial population, the SSGS is called to calculate the new timings for all activities (i.e. schedules an activity in earliest possible time in which both precedence constraints and resource constraints for given activity are met) and acquire the fitness values. Hence, the SSGS eliminates all options which do not respect precedence, and all activities of each solution are successively examined and reordered according to the precedence feasibility condition from left to right. Then, the fitness of the sites visited by the scouts is set to the makespan of the project.

The sampled scouts are ranked, and the m fittest scout bees are picked for local search (i.e. neighborhood search) [21]. To avoid duplication, a neighborhood (called a flower patch or site) is created around every best solution. Amongst these m selected scouts, the top-rated $e \leq m$ (elite) scouts recruit *nre* foragers, and the remaining $m - e$ (non-elite) best scouts recruit $nrb \leq nre$ foragers for local exploration. One neighboring technique based on the swap operator is introduced, eliminating two parameters from the original algorithm which are neighbourhood size (*ngh*) and shrinking factor (*sf*). The patch idea is replaced by this operator to be able to perform a local search and the, shrinking procedure is also removed from the algorithm. However, the abandonment procedure is kept to help the algorithm to improve the global search part. The local search operator is used to generate a variety of new neighboring feasible solutions for the forager bees: performing 'one' swap mutation. In this technique, two activities from each best solution are simply selected at random and their positions are swapped. The algorithm picks the top bee (i.e. solution or schedule) from each site (i.e. flower patch) as the new representative scout of the site to participate in the 'waggle dance' in the next generation. This step is called local search or exploitation [25,26]. If the fitness in a site has stagnated for a predefined number *stlim* of iterations, the scout associated to the flower patch is abandoned (i.e. the local fitness peak has been attained) and randomly re-initialized to a new position in the solution space. If the abandoned position corresponds to the best-so-far solution, it is recorded.

The remaining $(ns - m)$ scout bees are assigned to look for any new potential solutions via random sampling of the solution space, and hopefully escaping from local minima. This step is called global search, i.e. exploration [25].

At the end of each iteration, the population of the bee colony is updated and formed out of two groups. The first group comprises the m bees related to the best solutions (schedules) of the high-fitness flower patches (the results of the local exploitative search), and the second group comprises $ns - m$ scouts assigned to conduct random searches (the results of the global explorative search) [26,27]. Sequences of local and global search are repeated until one of the two stopping criteria is met: an adequate makespan is revealed, or a given number of optimization cycles have passed [21]. Figure 3 illustrates the pseudo code of the EDBA employed for solving the RCPSP problem.

The EDBA1 explores their neighborhoods in a selective manner to discover the most promising solutions, and find the global minimum of the objective function which is the minimal makespan (total completion time) schedule of the project. Therefore, maintaining the diversity and making tradeoff between diversification (i.e. global exploratory search) and intensification (i.e. local exploitative search) is indispensable to produce high-quality solutions.

It is clear that our algorithm does not require complex improvement techniques to obtain results. Our goal is to design an approach that is easily adaptable to the RCPSP and different variety of combinatorial optimization problems.

```
Enhanced Discrete Bees Algorithm (EDBA) for RCPSP
1. Initialize population with random solutions
2. Generate and evaluate initial feasible solutions from the population by using SSGS
   //Forming new population.
3. While (stopping criterion not met)
      a. Select sites for neighborhood search.
      b. Recruit bees for selected sites using neighboring technique (more bees for best e sites): use SSGS to
         generate and evaluate feasible solutions.
      c. Select the fittest bee (i.e. a schedule with smaller makespan) from each patch.
      d. Assign remaining bees to search randomly: use SSGS to generate and evaluate feasible solutions.
4. End While.
5. Return the best feasible schedule found so far;
```

Figure 3: Pseudo code of the proposed EDBA1 for RCPSP problem

3.2 The Second Enhanced Discrete Bees Algorithm (EDBA2)

In the first Enhanced Discrete Bees Algorithm (denoted by EDBA1), it is possible that the algorithm retains duplicate solutions, which means that each selected site may sometimes unintentionally produce the same sequences as other sites during the local search. Furthermore, there is a probability that the second best solution, and sometimes even the third best solution on a site, may have better fitness values than the best solution of other sites. Keeping deceitful or duplicitous solutions for the next generation could cause high computational time as well as difficulties at the local optimum.

To overcome this disadvantage, our idea is to introduce the Negative Selection in Artificial Immune Systems. This process is based on the principles of T cell maturation and self / non-self-discrimination in biological immune systems [28]. The objective is to improve the algorithm in choosing the best solutions from selected sites after the local neighborhood research for the next generation. This new version is denoted by EDBA2.

In the EDBA1, after the neighborhood search, the best bee of each site will be recorded for the next iteration. For the EDBA2, all the solutions derived from the local search will be sorted (according to their fitness values) and transferred to the repertoire (R0) and the best bees are selected by the Negative Selection process. Figure 4 presents the local search procedure of the second Enhanced Discrete Bees Algorithm (EDBA2). Figure 5 shows the adopted Negative Selection model.

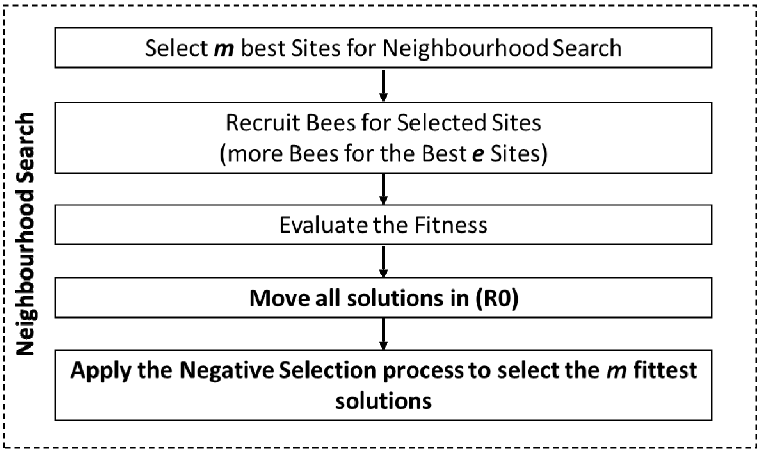

Neighbourhood Search
Select *m* best Sites for Neighbourhood Search
Recruit Bees for Selected Sites (more Bees for the Best *e* Sites)
Evaluate the Fitness
Move all solutions in (R0)
Apply the Negative Selection process to select the *m* fittest solutions

Figure 4: Local search of EDBA 2

Two major phases of Negative Selection process, detector generation and anomaly monitoring, are therefore considered. The first solution in repertoire (R0) will be copied into the Self-strings (S) and into the repertoire (R). Then, the next solution in the repertoire (R0) will be considered by associating it with strings in (S). If it is recognized, it will be eliminated. Otherwise, it will be introduced in the repertoire (R). In the opposite direction, if all the strings of (S) do not correspond to the solution from the repertoire (R0), then it is eliminated and replaced by the solution of the repertoire (R0).

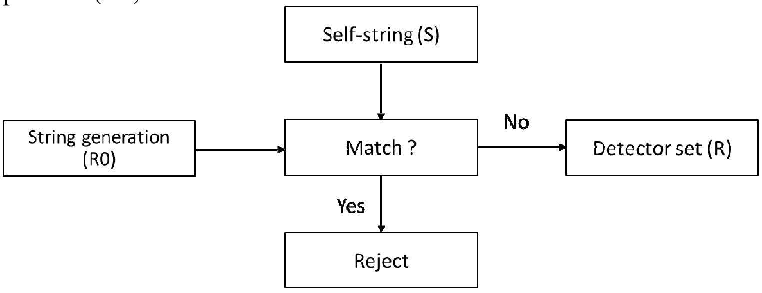


Figure 5: Negative Selection model

4 Computational Experiments

The proposed EDBA algorithms were implemented using Octave programming language and all experiments were performed on a 3.40 GHz Intel Core i3 3240 CPU and 4 GB RAM running a 64-bit operating system. The standard PSPLib (Project Scheduling Problem Library) test problems (i.e. J30, J60, J90 and J120) [29] were used for evaluating and comparing the EDBA approaches with existing algorithms from the literature. J30, J60 and J90 consist of 480 instances with 30, 60 and 90 non-dummy activities, respectively. J120 consists of 600 instances with 120 non-dummy activities. Hence, there are total 2040 RCPSP instances are simulated in this investigation. All projects involved 4 renewable resources, and each activity with a maximum of 3 successors. Table 1 illustrates the 30 activities example instance j301_6 in PSPLIB. In this instance case, there are 4 types of resource, 30 activities and sufficient processors available are assumed.

The numbers of generated solutions (schedules) are set as 1000, 5000 and 50000 for each instance. The parameters' settings of the EDBA are listed in Table 2. Values of parameters used in numerical experiments were selected, based on those recommended by our previous investigations [30,31].

The success rate was used to measure the performance of the algorithms. It shows the number of instances in a case study which are successfully solved by the algorithms. For an instance, an algorithm is called successful if it can find the optimal schedule which has the minimum makespan.

A set of simulations has been carried out to verify how many cases of PSPLIB library can be solved by the proposed EDBA. The objective is to find optimal solution (for J30) or lower bound solution (for J60, J90 and J120).

Table 1: 30 activities case (j301_6) with precedence and resource requirement constraints [29]

Activity#	**Successors**			**Activity#**	**Duration**	**Required resources**			
						R1	**R2**	**R3**	**R4**
1	2	3	4	1	0	0	0	0	0
2	5	7	8	2	10	0	0	0	4
3	11			3	1	0	0	0	10
4	6	16		4	9	4	0	0	0
5	15	23		5	3	6	0	0	0
6	10	12		6	1	3	0	0	0
7	9	14	25	7	7	0	4	0	0
8	13			8	1	0	0	0	2
9	24			9	4	10	0	0	0
10	22			10	10	0	0	0	2
11	14	16	24	11	6	0	0	10	0
12	13	21		12	2	0	0	0	6
13	17	24	30	13	3	0	7	0	0
14	18			14	1	0	0	3	0
15	16	29		15	3	0	0	0	6
16	19			16	1	0	0	10	0
17	18			17	3	0	0	0	7
18	20	31		18	10	0	0	0	9
19	28			19	1	0	6	0	0
20	26			20	3	5	0	0	0
21	28			21	4	0	3	0	0
22	28			22	2	8	0	0	0
23	27			23	4	1	0	0	0
24	26	31		24	2	3	0	0	0
25	30			25	4	0	9	0	0
26	29			26	6	0	0	0	7
27	30			27	9	0	0	0	7
28	31			28	2	0	0	0	5
29	32			29	1	0	0	9	0
30	32			30	1	0	0	9	0
31	32			31	9	0	0	4	0
32				32	0	0	0	0	0
Available resources						12	10	10	12

Table 2: Parameters Setting [30,31]

Algorithm	**Parameters**	**Value**
EDBA	Scout bees (ns)	12
	Elite sites (e)	2
	Best sites (m)	6
	Recruited bees for elite sites (nre)	29
	Recruited bees for remaining best sites (nrb)	9
	Limit of stagnation cycles for site abandonment (stlim)	10
	Number of generation schedules (genmax)	1000/5000/50000

Tables 3-6 present the computational results, comparing the proposed EDBA1 and EDBA2 algorithms with 11 other approaches for the four instance sets of RCPSP. Each algorithm is briefly defined by a few keywords, the SGS employed, and the reference. The best results for each instance set are shown in boldface; the second best is in italics.

From the results, it can be seen that the success rate values decrease (i.e. the performance of the proposed models decrease) as the problem size increase. This involves that the RCPSP with larger scale is more difficult to solve since the problem is NP-hard problem. Furthermore, for each set of instances, the success rate values increase as the schedules number increase.

The percentage of problem instances successfully solved by EDBA1 varies in range of [76.88%, 93.96%], [67.08%, 77.71%], [65.63%, 74.58%] and [17.17%, 31.67%] for J30, J60, J90 and J120, respectively. Whereas, the success rates of the second proposed method EDBA2 are between 89.21% and 90.62% for J30, 67.29% and 76.04% for J60, 67.08% and 74.38% for J90, and 18.67% and 31.50% for J120.

Table 3 presents the success rates for the instance set J30 in which all problem instances have been solved to optimality. The results show that the proposed EDBA1 obtains the fifth rank among the 13 existing heuristics after EDBA2, BA, ACROSS and BSO for 1000 schedules and obtains the first rank after 5000 and 50000 schedules, respectively. The second improved algorithm EDBA2 obtained the first after 1000 and the second after 5000 and the sixth after 50000 schedule generations, respectively.

Table 3: Average success rate (%) from optimal makespan for J30

Algorithm / SGS	**SGS**	**Reference**	**Schedules**		
			1000	**5000**	**50000**
GA	Serial	Hartmann [10]	51.87	53.75	58.96
PSO	Serial	Zhang et al. [14]	53.54	58.13	61.26
ANGEL	Serial	Tseng and Chen [32]	71.46	78.70	89.11
GAPS	Parameterized active schedule	Mendes et al. [12]	57.30	63.29	68.33
ACOSS	Serial & Parallel	Chen et al. [15]	77.41	85.04	***93.27***
Neurogenetic	Serial & Parallel	Agrawal et al. [33]	74.13	81.33	91.05
OOP-GA	Serial	Montoya-Torres et al. [11]	53.19	57.48	65.82
PSO+	Novel SGS	Chen et al. [15]	67.80	75.05	87.15
ABC	Serial / Priority list	Akbari et al. [5]	72.71	80.84	90.42
BSO	Serial / Priority list	Ziarati et al. [17]	77.30	85.63	92.09
BA	Serial / Priority list	Ziarati et al. [17]	***78.54***	86.25	92.50
EDBA 1	Serial / Activity list	Proposed approach 1	76.88	**89.58**	**93.96**
EDBA 2	Serial / Activity list	Proposed approach 2	**80.21**	***87.71***	90.62

Table 4 shows the lower bound results for instances with 60 activities (J60). Due to the larger number of activities, the proposed approach and the other ones have more difficulty to solve. The success rates decrease compared to J30 case study. In an experiment of 1000 and 5000 schedules, EDBA2 scheme has been proven to be better than other algorithms followed by EDBA1 as second best. After 50000 schedules, EBDA1 is the most efficient method and EBDA2 became the second for this case study.

Table 4: Average success rate (%) from lower bound for J60

Algorithm / SGS	**SGS**	**Reference**	**Schedules**		
			1000	**5000**	**50000**
GA	Serial	Hartmann [10]	43.33	52.30	55.84
PSO	Serial	Zhang et al. [14]	45.21	53.96	58.55
ANGEL	Serial	Tseng and Chen [32]	53.95	61.33	64.91
GAPS	Parameterized active schedule	Mendes et al. [12]	46.39	52.40	56.07
ACOSS	Serial & Parallel	Chen et al. [15]	55.12	60.94	65.21
Neurogenetic	Serial & Parallel	Agrawal et al. [33]	56.37	63.86	68.35
OOP-GA	Serial	Montoya-Torres et al. [11]	48.52	54.04	59.26
PSO+	Novel SGS	Chen et al. [15]	58.66	62.05	66.76
ABC	Serial / Priority list	Akbari et al. [5]	61.88	67.09	71.88
BSO	Serial / Priority list	Ziarati et al. [17]	64.38	70.63	71.88
BA	Serial / Priority list	Ziarati et al. [17]	66.25	68.34	71.67
EDBA 1	Serial / Activity list	Proposed approach 1	***67.08***	***71.25***	**77.71**
EDBA 2	Serial / Activity list	Proposed approach 2	**67.29**	**71.67**	***76.04***

The J90 simulation results are shown in Table 5. The proposed EDBA2 and EDBA1 in this study rank first and second, respectively, on the comparison table for 1000 and 5000 schedules. When 50000 schedules are examined, our EDBA1 outperforms all the approaches used in this comparative study, followed by the EDBA2 algorithm.

Table 5: Average success rate (%) from lower bound for J90

Algorithm / SGS	**SGS**	**Reference**	**Schedules**		
			1000	**5000**	**50000**
GA	Serial	Hartmann [10]	39.95	44.30	53.96
PSO	Serial	Zhang et al. [14]	42.63	49.34	55.27
ANGEL	Serial	Tseng and Chen [32]	53.08	58.23	61.49
GAPS	Parameterized active schedule	Mendes et al. [12]	45.85	48.72	54.21
ACOSS	Serial & Parallel	Chen et al. [15]	55.72	58.10	62.03
Neurogenetic	Serial & Parallel	Agrawal et al. [33]	57.64	60.53	64.82
OOP-GA	Serial	Montoya-Torres et al. [11]	47.44	50.58	56.84
PSO+	Novel SGS	Chen et al. [15]	61.65	65.24	67.17
ABC	Serial / Priority list	Akbari et al. [5]	60.13	65.21	68.75
BSO	Serial / Priority list	Ziarati et al. [17]	64.59	69.17	69.59
BA	Serial / Priority list	Ziarati et al. [17]	64.80	67.09	70.00
EDBA 1	Serial / Activity list	Proposed approach 1	***65.63***	***71.25***	**74.58**
EDBA 2	Serial / Activity list	Proposed approach 2	**67.08**	**71.88**	***74.38***

The last case study with 120 activities is the most challenging to solve. Table 6 displays the success rates for this case study. The investigated approaches have more difficulty and their performance decrease drastically. The results demonstrate that the EDBA algorithms had good and competitive performance compared with other algorithms. The first and the second rank are obtained by the proposed EDBA2 and BA, respectively, after 1000 schedule generations. The two proposed EDBA algorithms perform equally and achieve the first best performance when the number of schedules is 5000. The best success rates after 50000 schedules are obtained by EDBA1 approach. The second rank is obtained by the second version of the proposed algorithm.

Table 6: Average success rate (%) from lower bound for J120

Algorithm / SGS	**SGS**	**Reference**	**Schedules**		
			1000	**5000**	**50000**
GA	Serial	Hartmann [10]	7.15	9.18	16.33
PSO	Serial	Zhang et al. [14]	9.44	11.89	20.42
ANGEL	Serial	Tseng and Chen [32]	14.85	15.36	19.91
GAPS	Parameterized active schedule	Mendes et al. [12]	10.12	12.83	18.78
ACOSS	Serial & Parallel	Chen et al. [15]	14.56	17.72	20.69
Neurogenetic	Serial & Parallel	Agrawal et al. [33]	14.32	16.85	21.08
OOP-GA	Serial	Montoya-Torres et al. [11]	10.39	13.51	19.82
PSO+	Novel SGS	Chen et al. [15]	14.60	16.48	21.26
ABC	Serial / Priority list	Akbari et al. [5]	15.34	18.97	22.84
BSO	Serial / Priority list	Ziarati et al. [17]	17.00	22.50	25.17
BA	Serial / Priority list	Ziarati et al. [17]	***17.84***	20.84	24.50
EDBA 1	Serial / Activity list	Proposed approach 1	17.17	**25.00**	**31.67**
EDBA 2	Serial / Activity list	Proposed approach 2	**18.67**	**25.00**	***31.50***

Different models of bee algorithms have been proposed in literature such as Artificial Bee Colony (ABC) [5], Bees Algorithm (BA) [17] and Bee Swarm Optimization (BSO) [17] to solve the RCPSP (Tables 3-6). Each approach uses different types of bees to provide appropriate level of exploration over search space while maintaining exploitation of good solutions. The three bee algorithms use the priority-based representation [14] for their individuals and the SSGS to generate the schedule.

We can see from the results that our proposed EDBA algorithms which uses the activity-list as a default representation of a solution commonly offers better results than other bee algorithms using random-key representation schemes in majority of cases. In contrast, the priority list-based BA, followed by BSO surpasses our EDBA1 approach over j30 for 1000 schedules and EDBA2 over j120 for 50000 schedules. In addition, BA provides better performance than EDBA1 approach for j120 when we set number of schedules at 1000.

To sum up, the best results were obtained by the proposed EDBA1 and/or EDBA2 algorithms. EDBA2 ranks first, compared to the other approaches, with 6 out of 12 case studies, followed by EDBA1 in 5 out of 12 cases. The two algorithms give the same result in only one case study (J120 with 5000 schedules).

EDBA2 is a good alternative if the number of schedules generated is not too high. Otherwise, EDBA1 is the best option.

In brief, the proposed EDBA algorithms are highly ranked between state-of-the-art algorithms and reveals robust competitiveness as an effective technique in solving medium and large-scale RCPSP. Such level of improvement is associated with the BA intelligent behavior, the use of the permutation based representation formalism for the schedules, the employment of serial SGS in order to generate feasible schedules for the RCPSP, and the application of move operators for local search.

5 Conclusion

This paper presents two Enhanced Discrete Bees Algorithms for solving the classical resource constrained project scheduling (RCPSP) with the makespan minimization criterion. The proposed algorithms, which are based on foraging behaviors of honey bees, use a population of different types of bees to find the optimal solution. Its codifications are based on the activity-list representation schemes with the Serial Schedule Generation Scheme (SSGS) to yield feasible solutions. In local search, we use one structure of neighborhoods, called swap operator. In addition, the site abandonment procedure is used to maximize the diversity of the population and to push down the solution from local optimum. In second variant, the Negative Selection process is introduced in order to obtain better performance. The improved algorithms are developed with reduction of parameters to be tuned.

The performance of our approaches has been evaluated on the well know PSPLIB instances. The computational experiments confirm that the proposed models produce consistently high-quality solutions and it is more effective than other bee inspired algorithms and several well-known heuristics from the literature. Moreover, the use of negative selection rules has demonstrated their effectiveness to boost the solution quality of RCPSP if the number of schedules to be analyzed is low.

As future research, the proposed approach could be used for solving other important variants of the RCPSP, e.g. multiple modes and multiple projects, and different kinds of hard combinatorial optimization problems. Also, hybridization with the other heuristic or meta-heuristics may offer a way to improve their efficiency.

References

[1] Harish Garg. A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation,* 274: 292–305, 2016.

[2] Sönke Hartmann and Dirk Briskorn. A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1-14, 2010.

[3] Ruey-Maw Chen. Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications,*38(6):7102–7111, 2011.

[4] Olivier Liess and Philippe Michelon. A constraint programming approach for the resource-constrained project scheduling problem. *Annals of Operations Research,*157:25–36, 2007.

[5] Reza Akbari, Vahid Zeighami and Koorush Ziarati. Artificial Bee colony for resource constrained project scheduling problem. *International Journal of Industrial Engineering Computations,*2(1):45–60, 2011.

[6] Rainer Kolisch. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research,* 90(2):320–333, 1996.

[7] Amer Fahmya, Tarek M Hassanb and Hesham Bassioni. Improving RCPSP solutions quality with Stacking Justification – Application with particle swarm optimization. *Expert Systems with Applications,* 41(13):5870–5881, 2014.

[8] Kamel Bouleimen and Lecocq H. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research,*149(2):268–281, 2003.

[9] Ikonomou A, John Galletly and Daniel RC. Solving resource-constrained project scheduling problems using Tabu Search. In *International Conference on Intelligent Systems for Manufacturing IFIP Advances in Information and Communication Technology*, pages 311–322. Springer, 1998

[10] Sönke Hartmann. A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics*.;45(7):733–750, 1998

[11] Jairo R.Montoya-Torres, Edgar Gutierrez-Franco and Carolina Pirachicán-Mayorga. Project scheduling with limited resources using a genetic algorithm. *International Journal of Project Management*. 2010;28(6):619–628.

[12] Jorge J M Mendes, José F Gonçalves and Mauricio G C Resende. A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers & Operations Research*, 36(1):92–109, 2009

[13] Daniel Merkle, Martin Middendorf and Hartmut Schmeck. Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation,* 6(4):333–346, 2002

[14] Hong Zhang, Xiaodong Li, Heng Li and Fulai Huang. Particle swarm optimization-based schemes for resource-constrained project scheduling. *Automation in Construction,*14(3):393–404, 2005

[15] Ruey-Maw Chen, Chung-Lun Wu, Chuin-Mu Wang and Shih-Tang Lo. Using novel particle swarm optimization scheme to solve resource-constrained scheduling problem in PSPLIB. *Expert Systems with Applications*, 37(3):1899–1910, 2010.

[16] Yan-jun Shi, Fu-Zhen Qu, Wang Chen and Bo Li. An artificial bee colony with random key for resource-constrained project scheduling. In *International Conference on Intelligent Computing for Sustainable Energy and Environment. ICSEE 2010*, pages 148–157. Springer, 2010.

[17] Koorush Ziarati, Reza Akbari and Vahid Zeighami. On the performance of bee algorithms for resource-constrained project scheduling problem. *Applied Soft Computing*, 11(4):3720–3733, 2011.

[18] Rina Agarwal, Manoj K Tiwari and Sanat K Mukherjee. Artificial immune system based approach for solving resource constraint project scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 34(5-6)584–593, 2006.

[19] Dieter Debels, Bert De Reyck, Roel Leus and Mario Vanhoucke. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research,* 169(2):638–653, 2006.

[20] Xiao-long Zheng and Ling Wang. A multi-agent optimization algorithm for resource constrained project scheduling problem. *Expert Systems with Applications*, 42(15-16): 6039–6049, 2015.

[21] Duc T Pham, Afshin Ghanbarzadeh, Ebubekir Koç, Sameh Otri, Shafqat Rahim and Monji Zaidi. The Bees Algorithm — A novel tool for complex optimisation problems. In *Proceedings of International Conference on Intelligent Production Machines and Systems*, pages 454–459, 2006

[22] Jacek Blazewicz, Jan K Lenstra and AHG R Kan. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics,* 5(1):11–24, 1983.

[23] Rainer Kolisch and Sönke Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research,*174(1):23–37, 2006.

[24] Rainer Kolisch and Sönke Hartmann. Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis. *Project Scheduling International Series in Operations Research & Management Science*, 14:147–178, 1999.

[25] Duc T Pham and Marco Castellani. The Bees Algorithm: Modelling foraging behaviour to solve continuous optimization problems. In *Proceedings of the Institution of Mechanical Engineers, Part C*: *Journal of Mechanical Engineering Science*, 223(12):2919–2938, 2009

[26] Mohamed A Nemmich, Fatima Debbat and Mohamed Slimane. A data clustering approach using bees algorithm with a memory scheme. In *Demigha O, Djamaa B, Amamra A, editor. CSA 2018: Advances in Computing Systems and Applications*, pages 261–270. Springer, 2018.

[27] Mohamed A Nemmich and Fatima Debbat. Bees algorithm and its variants for complex optimisation problems. In: *The 2nd International Conference on Applied Automation and Industrial Diagnostics (ICAAID 2017)*. Djelfa, 2017.

[28] Stephanie Forrest, Lawrence Allen, Alan S Perelson and Rajesh Cherukuri. Self-nonself discrimination in a computer. In *IEEE Computer Society Symposium on Research in Security and Privacy*, pages 202-212. IEEE, 1994.

[29] Rainer Kolisch, Christoph Schwindt and Arno Sprecher. Benchmark instances for project scheduling problems. *Project Scheduling International Series in Operations Research & Management Science,* 14:197–212,1999

[30] Mohamed A Nemmich, Fatima Debbat and Mohamed Slimane. A novel hybrid firefly bee algorithm for optimization problems. *International Journal of Organizational and Collective Intelligence*, 8(4):21–46, 2018.

[31] Mohamed A Nemmich, Fatima Debbat and Mohamed Slimane. A Permutation-Based Bees Algorithm for Solving Resource-Constrained Project Scheduling Problem. *International Journal of Swarm Intelligence Research*, 10(4):1-24, 2019.

[32] Lin-Yu Tseng and Shih-Chieh Chen. A hybrid metaheuristic for the resource-constrained project scheduling problem. *European Journal of Operational Research,*175(2):707–721, 2006.

[33] Anurag Agarwal, Selcuk Colak and Selcuk Erenguc. A Neurogenetic approach for the resource-constrained project scheduling problem. *Computers & Operations Research.* 38(1):44–50, 2011.

Inteligencia Artificial 22(64), 135-142
doi: 10,4114/intartif.vol22iss64pp135-142

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Aplicación de la Matriz de Co-ocurrencias de niveles de gris a Imágenes de Masa de Aceituna.

Gray level Co-ocurrence matrix application to olive paste images

Antonio Jiménez Márquez[1], Gabriel Beltrán Maza
Instituto de Investigación y Formación Agraria y Pesquera (IFAPA).Junta de Andalucía. Centro 'Venta del Llano'. Ctra Bailén-Motril km18,5. 23620 Megibar (Jaén). España.
[1]antonio.jimenez.marquez@juntadeandalucia.es

Abstract This paper shows the results obtained from images processing digitized, taken with a 'smartphone', of 56 samples of crushed olives, using the methodology of the gray-level co-occurrence matrix (GLCM). The values of the appropriate direction (θ) and distance (D) that two pixel with gray tone are neighbourhood, are defined to extract the information of the parameters: Contrast, Correlation, Energy and Homogeneity. The values of these parameters are correlated with several characteristic components of the olives mass: oil content (RGH) and water content (HUM), whose values are in the usual ranges during their processing to obtain virgin olive oil in mills and they contribute to generate different mechanical textures in the mass according to their relationship HUM / RGH. The results indicate the existence of significant correlations of the parameters Contrast, Energy and Homogeneity with the RGH and the HUM, which have allowed to obtain, by means of a multiple linear regression (MLR), mathematical equations that allow to predict both components with a high degree of correlation coefficient, r = 0.861 and r = 0.872 for RGH and HUM respectively. These results suggest the feasibility of textural analysis using GLCM to extract features of interest from digital images of the olives mass, quickly and non-destructively, as an aid in the decision making to optimize the production process of virgin olive oil.

Resumen En éste trabajo se muestran los resultados obtenidos en el análisis de imágenes digitalizadas realizadas con un '*smartphone*', de 56 muestras de aceituna trituradas, mediante la metodología de la matriz de co-ocurrencia del nivel de gris (GLCM). Se definen los valores de la dirección y número de píxeles adecuados para extraer la información de los parámetros: Contraste, Correlación, Energía y Homogeneidad. Los valores de estos parámetros son correlacionados con varios componentes característicos de dicha masa de aceituna: su contenido en aceite (RGH) y su contenido en agua (HUM), cuyos valores cubren los rangos habituales durante su procesado para la obtención del aceite de oliva virgen en almazaras y contribuyen, además, a generar diferentes texturas mecánicas en la masa según su relación HUM/RGH. Los resultados obtenidos indican la existencia de correlaciones significativas de los parámetros Contraste, Energía y Homogeneidad con el RGH y la HUM, que han permitido obtener, mediante una regresión lineal múltiple (MLR), ecuaciones matemáticas que permiten predecir ambos componentes con un alto grado de coeficiente de correlación, r=0,861 y r=0,872 para RGH y HUM respectivamente. Estos resultados sugieren la viabilidad del análisis textural mediante GLCM para extraer información de interés de imágenes digitales de la masa de aceituna, de forma rápida y no destructiva, como ayuda en la toma de decisiones de optimización del proceso de elaboración del aceite de oliva virgen.

Keywords: Olive paste, Image processing, GLCM, Texture, Olive oil.
Palabras clave: Masa de aceituna, Procesado de imagenes, GLCM, Textura, Aceite de Oliva.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

1. Introducción

El análisis textural de una imagen digitalizada constituye un método ampliamente utilizado para la caracterización e identificación de zonas de interés dentro de la imagen. Uno de los métodos más comunes en el análisis de textura de imágenes es la matriz de co-ocurrencias de los niveles de grises (GLCM). Éste método, empleado por primera vez por Haralick et al. [1] para caracterizar y discriminar diferentes tipos de terrenos a partir de imágenes satelitales, es muy utilizado en sectores tan diversos como la teledetección [2], análisis de superficies [3][4], medicina [5][6][7], últimamente en la industria agroalimentaria [8][9][10][11][12] y en particular en la industria de elaboración del aceite de oliva virgen [13][14].

Por definición, la GLCM consiste en una matriz de la frecuencia con que un píxel, con un nivel de gris, se repite en una relación espacial específica [15][16] y en la que se asume que la información textural de la imagen viene dada por la distribución espacial que los diferentes tonos de gris tienen entre ellos. Así pues, a partir de una imagen digital, convertida a niveles de gris, un análisis textural mediante GLCM permite extraer información de interés relacionada con características del objeto fotografiado.

Por su parte, la textura mecánica es una propiedad característica de muchos productos agroalimentarios que influye en el aspecto de estos y que afectan a la percepción humana cuando evalúa su estructura superficial, comportándose como indicador visual de la ejecución de un proceso industrial [17].

Para el caso de éste trabajo, se pretende ver la viabilidad de la GLCM para obtener información sobre ciertas características del fruto de la aceituna a partir de imágenes tomadas de la masa triturada de ésta. La aceituna, fruto del olivo (‘Olea Europaea L.’), constituye la materia prima para la obtención del aceite de oliva virgen en las almazaras. Dicho fruto se caracteriza por estar constituido, básicamente, por aceite, agua y materia sólida en diferentes proporciones que influyen en sus características morfológicas, fisiológicas y bioquímicas y que varían a lo largo de su proceso de maduración [18]. Cuando la aceituna se tritura, para extraer el aceite en la fábrica, la masa que origina presenta una textura mecánica característica y diferenciable visualmente según el momento en que ésta es recogida y molida, estando muy influenciada por los contenidos en agua y aceite en ese momento y afectando a la capacidad extractiva de las máquinas empleadas [19]. Estas variaciones en sus características hace presumible pensar que un análisis GLCM de las imágenes digitales de dicha masa molida pueda dar cieta información sobre las características de ésta en cuanto al contenido en aceite y agua presente en el fruto.

Si bien existen métodos para realizar esta caracterización, como son la Resonancia Magnética Nuclear (RMN) o la espectroscopía en el infrarrojo cercano (NIR), estos precisan de costosos equipos en comparación con los ordenadores de bolsillo que constituyen los actuales ‘smartphones’, en los cuales se combinan la capacidad fotográfica con la capacidad de procesamiento para constituir un equipo de visión artificial facilmente asequible y portátil, con lo que se puede convertir en una interesante herramienta de ayuda para el técnico de almazara a la hora de tomar decisiones rápidas, a pié de fábrica, sobre la regulación y optimización del proceso en base a las características del fruto a procesar.

Para contribuir a éste desarrollo tecnológico, en éste trabajo, se analizan imágenes de masa de aceituna, en un amplio rango de valores de aceite y agua, tomadas con un teléfono móvil en unas condiciones determinadas y fijas. Se realiza un estudio previo de las variables de GLCM, con las cuales se vá a efectuar el análisis de las imagenes para extraer la información de los parámetros texturales y, finalmente, se busca la existencia de correlaciones entre éstas y los parámetros composicionales de la aceituna considerados.

2. Material y métodos

2.1 Material vegetal.

Se ha empleado un total de 56 muestras de aceituna procedentes de varias variedades, siendo la predominante la variedad ‘Picual’. Estas muestras de aceituna han sido analizadas entre los meses de noviembre a enero, coincidiendo con la época normal de recolección y procesamiento de éste fruto, lo que ha proporcionado un amplio rango de características asociadas a su estado de maduración. Las muestras, que proceden de diferentes agricultores y fábricas, han sido procesadas dentro de las 24 horas desde su recolección.

2.2 Métodos.

2.2.1 Preparación de las muestras.- La toma de imágenes digitales ha sido llevada a cabo sobre muestras de aceituna molida. Para ello las diferentes muestras han sido molturadas en un molino de martillos, con criba dentada, empleado habitualmente para la preparación de muestras en los análisis físico-químicos habituales de éste fruto. Se muelen unos 500 g., aproximadamente, de cada muestra, recogida en un recipiente en donde es homogeneizada. Una porción de dicha masa de aceituna molida y homogeneizada es introducida en una cápsula cilíndrica de unos 8,5 cm. con un hueco cilíndrico de unos 4,5 cm. de diámetro por unos 1,5 cm. de profundidad, nivelando su superficie con una espátula. Esta última operación se realiza por duplicado para realizar la toma de imágenes. La muestra, una vez digitalizada, es analizada inmediatamente para conocer sus características composicionales.

2.2.2 Métodos químicos.- La caracterización composicional de la masa de aceituna se lleva a cabo mediante el equipo de infrarrojo cercano (NIR) InfraAlyzer 2000 de Bran+Luebbe (Bran+Luebbe GMBH, Norderstedt /Alemania), para determinar los parámetros de: % m/m de contenido graso sobre base húmeda (RGH) y % m/m de contenido en agua (HUM).

2.2.3 Toma de imágenes.- La digitalización de las muestras de masa de aceituna se ha llevado a cabo empleando la cámara y flash de un 'smartphone' comercial de 4,5"de pantalla: cámara de 8 megapixel y flash situados en la parte central superior trasera. Para que la digitalización se realice siempre en las mismas condiciones se ha dispuesto de una caja de poliespan de 21x12,5x11 cm. en cuyo fondo se coloca la cápsula con la muestra y en la parte superior el teléfono móvil de tal manera que entre la superficie de la muestra y el objetivo de la cámara exista una distancia focal de 9 cm. que es la distancia a la cual, éste móvil, muestra en su pantalla toda la cápsula sin necesidad de hacer zoom. La digitalización se realiza con la aplicación de fotografía que trae de serie el sistema operativo android 4,4,2 de éste equipo, en modo 'autofocus' con la máxima resolución y flash activado. Las imágenes se almacenan en formato JPG. Bajo las mismas condiciones se ha realizado la digitalización de un blanco de cerámica como indicador de textura cero.

2.2.4 Pretratamiento de imágenes.- Para el posterior análisis textural las imágenes originales han sido pretratadas con las funciones 'imcrop' y 'rgb2gray' del software MATLAB (Matlab 7.0, MathWork). La primera función realiza un recorte de la imagen de la zona central de la muestra, eliminando las partes de la cápsula portamuestras que ha sido fotografiada. Esta función selecciona una zona cuadrada que origina una imagen en formato RGB-JPG de unos 1100x1100 píxeles. La segunda función convierte ésta imagen recortada en color RGB a una única matriz de escala de grises, eliminando la información de matiz y saturación pero conservando la de luminancia.

2.2.5 Análisis textural de imágenes.- La extracción de las propiedades texturales de la imagen transformada a escala de grises se lleva a cabo mediante las funciones 'graycomatrix' y 'graycoprops' de MATLAB. La primera función crea una matriz de co-ocurrencias a nivel de gris (GLCM) a partir de la imagen. Este método, propuesto por Haralick et al. [1], es uno de los más empleados en las aplicaciones de análisis texturales de imágenes y crea una matriz que describe la frecuencia con que un píxel central, con un determinado nivel de gris, aparece en una dirección y distancia específica dentro de un área cuadrada determinada de la imagen. Se utiliza la sintaxis: [glcmS,SI]=graycomatrix(IGray,'Offset',[0 D;-D D;-D 0;-D -D]), para cuatro orientaciones posibles y varias distancias de píxeles D.

La segunda función extrae, de la matriz de co-ocurrencias obtenida (glcmS), los indicadores texturales: Contraste, Correlación, Energía y Homogeneidad. El Contraste proporciona información sobre variaciones bruscas en la intensidad de grises por lo que un alto valor de éste parámetro indica un alto grado de variación de grises, el Contraste es 0 para una imagen constante (eq 1). La Correlación es un indicador de la dependencia lineal entre las diferentes intensidades presentes en el entorno definido por la matriz de co-ocurrencias, altas correlaciones están asociadas a imágenes con amplias áreas de similar intensidad y su valor de máxima correlación toma el valor de 1 o -1, según exista correlación positiva o negativa, respectivamente (eq 2). La Energía da una idea de la suavidad de la textura y mide, en cierta manera, la complejidad de la imagen, la Energía toma el valor de 1 para una imagen constante y poco desordenada (eq 3). Finalmente, la Homogeneidad es un indicador de la uniformidad de la imagen, cuanto menos variación hay en los niveles de grises la imagen es más uniforme y éste parámetro tiende a 1 (eq 4). La sintaxis empleada toma la forma de: stats=graycoprops(glcmS), donde 'stast' es una matriz que contiene los valores de los indicadores texturales.

Para éste trabajo, el análisis textural de las imágenes se ha llevado a cabo para cuatro orientaciones θ (0°, 45°, 90° y 135°) y varias distancias D entre píxeles (1, 3, 6 y 12)

$$Contraste = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i-j)^2 P_{D,\theta}(i,j) \tag{1}$$

$$Correlación = \frac{\sum_{i=o}^{N-1}\sum_{j=0}^{N-1}(i-\mu_i)(j-\mu_j)P_{D,\theta}(i,j)}{\sigma_i\sigma_j} \tag{2}$$

$$Energía = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}P_{D,\theta}(i,j)^2 \tag{3}$$

$$Homogeneidad = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\frac{P_{D,\theta}(i,j)}{1+|i-j|} \tag{4}$$

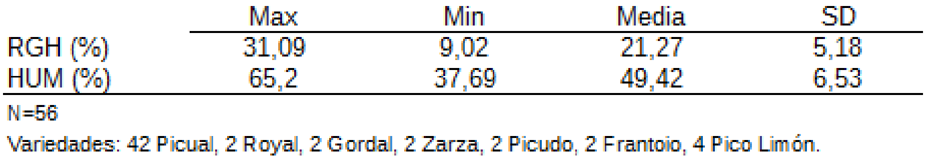
donde $P_{D,\theta}(i,j)$, es la matriz de co-ocurrencias para la pareja de píxeles con valores de gris i y j, a una distancia D y en la dirección θ.

2.2.6 Análisis estadístico.- Para analizar el efecto de la dirección y número de píxeles de cada parámetro textural se ha llevado a cabo un análisis de la varianza (ANOVA) de una vía con un test de Tukey, a un nivel de significancia de α=0,05, empleado el software Statistix 9.0 (Analytical Software). Este mismo software ha sido empleado para realizar una correlación de Pearson a fin de determinar si existe alguna correspondencia entre parámetros texturales y composicionales de la masa de aceituna. Estudios de correlación lineal simple y múltiple son llevados a cabo mediante el software Unscramber 9.7 (CAMO Software AS).

3. Resultados y discusión

3.1 Caracterización de las muestras.-Las muestras de aceituna empleadas en éste trabajo han proporcionado un amplio rango de valores para los parámetros analizados de RGH y HUM. Estos rangos son los habituales que se presentan, para éste tipo de fruto, en lo que dura la época de recolección, durante la cual tiene lugar un proceso de maduración de la aceituna que se refleja en los cambios que sufren estos parámetros y en la textura y consistencia de la masa molida a que da lugar. A principio de la recolección los frutos son más verdes, tersos y con valores de RGH y HUM bajos y altos, respectivamente. Conforme la recogida se va retrasando, el fruto suele sufrir una perdida de agua de vegetación con la consiguiente disminución del parámetro HUM. Al ser una determinación porcentual y ser la materia sólida muy constante, se produce un incremento relativo del parámetro RGH, por lo que a final de campaña de recogida los valores de RGH y HUM suelen ser altos y bajos, respectivamente, con frutos de color negro y con una textura blanda. En la Tabla 1, se muestra una descriptiva de las características composicionales de los fruto empleados en la digitalización.

Tabla 1. Descriptivas de las características composicionales de las muestras de masa de aceituna digitalizadas. Rendimientos graso en húmedo (RGH) y contenido en humedad (HUM). Valores máximos /Max), minimos (Min), valor medio y desviación estándar de la media (SD).



	Max	Min	Media	SD
RGH (%)	31,09	9,02	21,27	5,18
HUM (%)	65,2	37,69	49,42	6,53

N=56

Variedades: 42 Picual, 2 Royal, 2 Gordal, 2 Zarza, 2 Picudo, 2 Frantoio, 4 Pico Limón.

3.2 Matriz de co-ocurrencias y parámetros texturales.- La obtención de la matriz de co-ocurrencias ($P_{D,\theta}(i,j)$), de cada imagen, precisa de definir dos variables: el número de píxeles de distancia D que, con respecto al píxel de referencia, se van a evaluar, y la orientación θ en la que esa pareja de píxeles se encuentran en la imagen. En la figura 1 se detalla el proceso para generar la matriz GLCM.

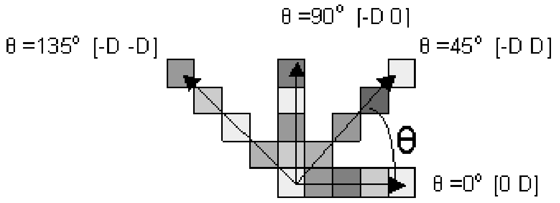


Figura 1. Construcción de la matriz de co-ocurrencia para niveles de gris (GLCM). Orientación (θ) y píxeles (D)

De acuerdo a la sintaxis [glcmS,SI]=graycomatrix(IGray,'Offset',[0 D;-D D;-D 0;-D -D]), se estudian las matrices:

orientación θ =0º [0 D], para D=1, ,3, 6 y 12.
orientación θ =45º [-D D], para D=1, ,3, 6 y 12.
orientación θ =90º [-D 0], para D=1, ,3, 6 y 12.
orientación θ =135º [-D -D], para D=1, ,3, 6 y 12.

Con éstas matrices se extraen, de la imagen, los parámetros texturales que caracterizan a cada muestra y, a partir de estos valores se determinan la mejor combinación de estas dos variables aplicando el análisis de la varianza con el test de Tukey, para cada parámetro textural. En la tabla 2 se muestran los resultados de éste ANOVA. En ella se puede observar que, de forma general, la variable D tiene un efecto significativamente importante en los cuatro parámetros texturales. Los valores del Contraste aumentan con D, mientras que Correlación, Homogeneidad y Energía disminuyen cuando aumenta D; solo en el caso de la Energía la diferencias entre 6 o 12 píxeles no se muestran significativas para determinadas orientaciones. En cuanto a la orientación se observa que para los cuatro parámetros no existen diferencias significativas cuando se consideran las orientaciones 0º y 90º o 45º y 135º, pero si entre 0º y 45º o 90º y 135º. Para el Contraste θ =45º o θ =135º son los que dan mayores valores, mientras que Correlación, Homogeneidad y Energía son los que dan los valores más bajos. Teniendo en cuenta estos resultados se toman como valores de trabajo θ =45º, para la orientación y D=6, para el número de píxeles.

Tabla 2. Valores GLCM para las variables texturales: Contraste, Correlación, Homogeneidad y Energía según la dirección (θ) y el número de píxeles (D). Análisis de la varianza con test de Tukey para α=0.05.

	Variables texturales y número de pixeles							
	Contraste				Homogeneidad			
Dirección θ	1	3	6	12	1	3	6	12
0	0,1589^{Bd}	0,5881^{Bc}	1,1272^{Ab}	1,6239^{Aa}	0,9297^{Aa}	0,8561^{Ab}	0,8036^{Ac}	0,7561^{Ad}
45	0,2431^{Ad}	0,8475^{Ac}	1,3956^{Ab}	1,8326^{Aa}	0,9051^{Ba}	0,8281^{Bb}	0,7777^{Ac}	0,7358^{Ad}
90	0,1582^{Bd}	0,5897^{Bc}	1,1391^{Ab}	1,6523^{Aa}	0,9302^{Aa}	0,8561^{Ab}	0,8028^{Ac}	0,7541^{Ad}
135	0,2444^{Ad}	0,8591^{Ac}	1,4204^{Ab}	1,8181^{Aa}	0,9045^{Ba}	0,8269^{Bb}	0,7758^{Ac}	0,7332^{Ad}
	Correlación				Energía			
Dirección θ	1	3	6	12	1	3	6	12
0	0,9384^{Aa}	0,7865^{Ab}	0,6007^{Ac}	0,4236^{Ad}	0,3127^{Aa}	0,2381^{Ab}	0,2288^{Ab}	0,1971^{Ac}
45	0,9073^{Ba}	0,6961^{Bb}	0,5057^{Bc}	0,3494^{Bd}	0,3119^{Aa}	0,2457^{Ab}	0,2117^{Ac}	0,1848^{Ad}
90	0,9386^{Aa}	0,7862^{Ab}	0,5969^{Ac}	0,4141^{Ad}	0,2956^{Aa}	0,2632^{Ab}	0,2288^{Ac}	0,1972^{Ad}
135	0,9071^{Ba}	0,6931^{Bb}	0,4995^{Bc}	0,3407^{Bd}	0,2955^{Aa}	0,2452^{Ab}	0,2104^{Ac}	0,1834^{Ad}

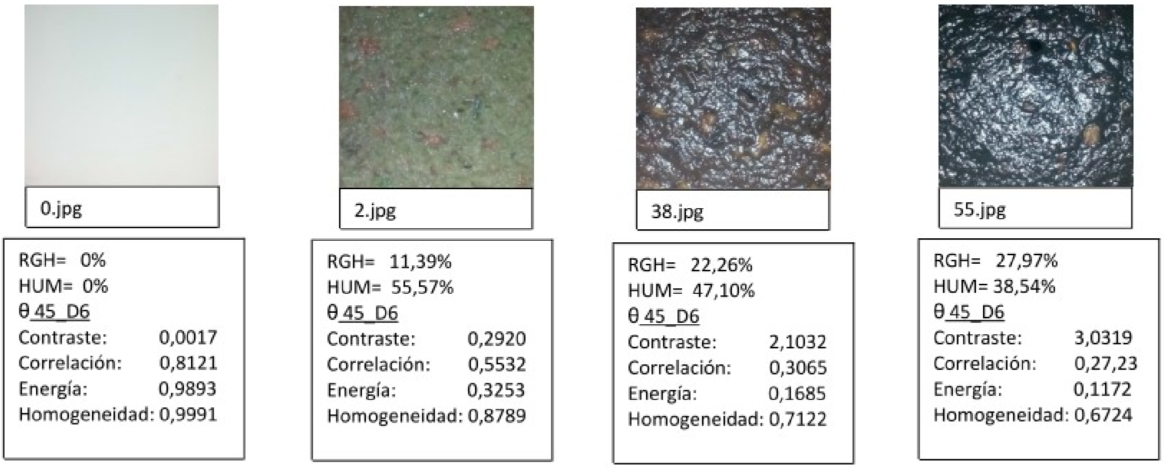
Test ANOVA: Letras diferentes indican diferencias significativas para una p=0.05.

Letras mayúsculas, diferencias entre valores de una columna (dirección θ).

Letras minúsculas, diferencias entre valores de una fila y por variable (número de pixeles)

Bajo éstas condiciones, y analizando los resultados individuales de varias muestras, como se muestra en la figura 2, se observa que el Contraste aumenta conforme la imagen se hace más compleja. La cerámica blanca (0.jpg) es una imagen muy constante, apenas hay variaciones del grises, y por tanto el valor de correlación es prácticamente cero; en las muestras, a medida que la humedad de ésta va disminuyendo el aceite va aflorando y comunicando a la imagen más tonos de color diferentes, de ahí que la imagen 55.jpg sea la que presente el valor del Contraste más elevado. Con respecto a los otros tres parámetros, se observa que todos disminuyen. La cerámica blanca, indicador de textura cero, presentan los valores altos; la imagen es totalmente homogénea, las intensidades entre los píxeles son prácticamente iguales, no hay complejidad, por lo que la Correlación y la Energía presenta valores elevados como corresponde a este tipo de imagen. Cuando se digitalizan muestras, dependiendo de los valores que toman RGH y HUM, las imágenes adquieren diferentes grados de complejidad que va aumentado a medida que disminuye la relación HUM/RGH. El contenido en agua (HUM) parece contribuir a mantener una cierta uniformidad en la superficie de la muestra de manera que a medida que va disminuyendo ésta la superficie se va haciendo más rugosa, menos homogénea, y aparecen mas sombras que contribuyen a incrementar los niveles de intensidades del color.

Se aprecia, pues, la existencia de una correspondencia entre la textura de la imagen y las características de la masa de aceituna, representadas en este caso por la RGH y la HUM. Para verificar éste hecho, se aplica una correlación de Pearson entre los parámetros texturales y los parámetros analíticos, observándose, como se aprecia en la tabla 3, la existencia de correlaciones lineales significativas del RGH y de la HUM con los parámetros: Contraste, Energía y Homogeneidad. El RGH muestra una correlación positiva con el Contraste y negativa con Energía y Homogeneidad, mientras que la HUM muestra con el Contraste una correlación negativa y positiva para Energía y Homogeneidad.



RGH= 0%
HUM= 0%
θ 45_D6
Contraste: 0,0017
Correlación: 0,8121
Energía: 0,9893
Homogeneidad: 0,9991

RGH= 11,39%
HUM= 55,57%
θ 45_D6
Contraste: 0,2920
Correlación: 0,5532
Energía: 0,3253
Homogeneidad: 0,8789

RGH= 22,26%
HUM= 47,10%
θ 45_D6
Contraste: 2,1032
Correlación: 0,3065
Energía: 0,1685
Homogeneidad: 0,7122

RGH= 27,97%
HUM= 38,54%
θ 45_D6
Contraste: 3,0319
Correlación: 0,27,23
Energía: 0,1172
Homogeneidad: 0,6724

Figura 2. Imágenes digitales del blanco de cerámica (0.jpg) y tres muestras de masa de aceituna de diferentes contenidos en grasa en húmedo (RGH) y humedad (HUM). Valores GLCM de las cuatro variables texturales para una orientación de 45º y 6 píxeles de distancia (θ 45_D6).

Tabla 3. Correlación de Pearson para los valores GLCM a 45º y 6 píxeles de distancia.

	Contraste	Correlación	Energía	Homogeneidad
Contraste	0			
Correlación	-0,5698	0		
Energía	-0,8993	0,3683	0	
Homogeneidad	-0,9722	0,6174	0,9278	0
RGH	0,8262	-0,4375	-0,7851	-0,8060
HUM	-0,8839	0,5125	0,7769	0,8383

3.3 Análisis de correlación.-En la tabla 4, se muestran los resultados de los estadísticos encontrados cuando se realiza una análisis de la correlación lineal simple, cuando solo se considera el parámetro textural de Contraste, y de la correlación lineal múltiple cuando se consideran los parámetros texturales de Contraste, Energía y Homogeneidad. Las correlaciones más altas, R^2=0,7405 y R^2=0,7611 para RGH y HUM respectivamente, se consiguen con la correlación lineal múltiple (MLR), con errores de método aceptables para métodos de análisis indirectos de estos parámetros (RGH=2.73%, HUM=3.30%).

Tabla 4. Resultados de las correlaciones lineales entre los parámetros ‘Contenido Graso en Húmedo’ y ‘ Contenido en Humedad’ con las variables texturales de ‘Contraste’, para la regresión lineal simples, y las variables texturales ‘Contraste’, ‘Energía’ y ‘Homogeneidad’ para la correlación lineal múltiple.

	Contenido Graso Húmedo		Contenido en humedad	
	Lineal simple	Lineal múltiple	Lineal simple	Lineal múltiple
Coeficiente de correlación (r).	0,8523	0,8605	0,8694	0,8724
Coeficiente de determinación (R^2)	0,7265	0,7405	0,7559	0,7611
R^2- ajustado	0,7214	0,7255	0,7513	0,7471
Error típico	2,75	2,73	3,28	3,3

En la figura 3 se muestran las rectas de regresión obtenidas entre los valores predichos por la MLR (eq 5)(eq 6) y los valores reales, en calibración y validación, que muestran la viabilidad del análisis textural de las imágenes en la predicción de propiedades composicionales de la masa de aceituna.

$$RGH=-1,6822+5,37514[Contraste]-22,5384[Energía]+24,8251[Homogeneidad] \quad (5)$$
$$HUM=93,0367-9,23238[Contraste]+9,23238[Energía]-39,99978[Homogeneidad] \quad (6)$$

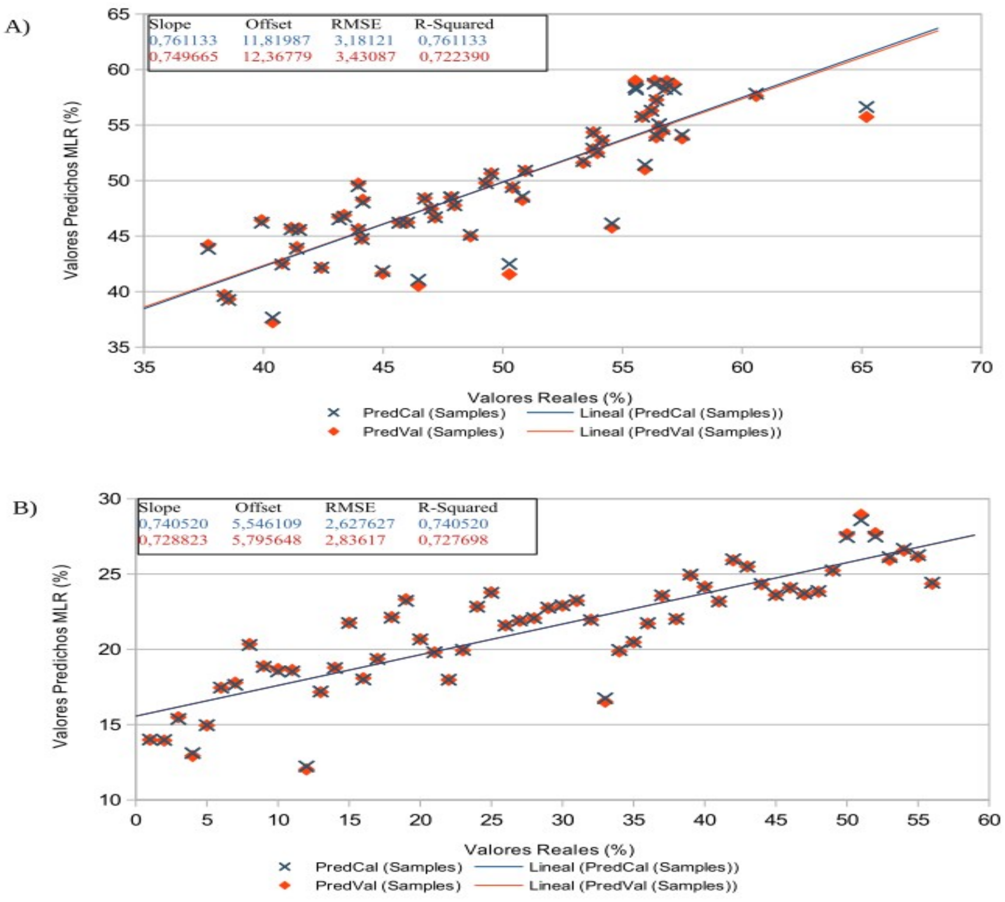


Figura 3. Correlación entre los valores reales y los predichos a partir de los modelos de regresión lineal múltiple obtenidos entre los parámetros HUM (A) y RGH (B) con las variables texturales: Contraste, Energía y Homogeneidad.

Conclusiones

Los resultados obtenidos en éste trabajo sugieren una viabilidad en la aplicación del análisis de imágenes mediante GLCM para extraer información de parámetros texturales de imágenes de masa de aceituna molida y asociación de estos con características de dicha masa. Para el caso en estudio, los cambios físicos en la masa de aceituna, originados por los cambios en la relación HUM/RGH, se manifiestan en cambios estructurales de la masa, como es el cambio de su textura física, dando lugar a imágenes con mayor gama de grises y más complejas, presentando una suficiente correlación con los parámetros GLCM: Contraste, Energía y Homogeneidad, lo que permite que empleando una metodología simple, que puede ser implementada mediante una aplicación a ‘smartphone’, se pueda obtener una información rápida ‘at-line’ de ciertas características de la aceituna que se está procesando. Esta viabilidad de la GLCM permite pensar, además, en la posibilidad de ampliar a otras características de la masa de aceituna, como es el análisis reológico de ésta para ver la evolución de su textura durante la fase de batido, la cual está muy asociada a la extractabilidad y rendimiento industrial del proceso en general.

Agradecimientos

Al laboratorio QUIMIFRESH SL, por facilitar las muestras y su análisis mediante el equipo NIR y a MA. Jiménez Bravo por su ayuda en la edición. Trabajo realizado durante el desarrollo del proyecto de excelencia P10-AGR 6429: ‘Modelado y optimización del proceso de elaboración del aceite de oliva virgen. Proyecto I: Integración de sensores ‘on-line’ y técnicas de redes neuronales para la optimización del proceso’. Proyecto financiando por el ‘Ministerio de Ciencia e Innovación’ y ‘Consejería de Economía, Innovación y Ciencia de la Junta de Andalucía’.

Referencias

[1] RM. Haralick,K. Shanmugam and I. Dinstein. Textural features for image classification. IEEE Transactions on Systems Man and Cybernetics SMC 3(6): 610-621. 1973

[2] R.Ciriza, M. González-Audícana, L. Albizua. Procedimiento simplificado para la caracterización de la textura del cultivo del frutal a nivel de parcela mediante los parámetros de Haralick. Revista de Teledetección 37, 57-66. 2012.

[3] E.S.Gadelmawla. A vision system for surface roughness characterization using the gray level co-occurrence matrix. NDT & E International 37(7), 577-588. 2004. DOI :10.1016/j.ndteint.2004.03.004

[4] C. Malegoria, L. Franzettia, R.Guidettib, E. Casiraghia and R. Rossia. GLCM, an image analysis technique for early detection of biofilm. Journal of Food Engineering 185, 48-55. 2016. DOI: 10.1016/j.jfoodeng.2016.04.001.

[5] N. Zulpe and V. Pawar. GLCM Textural Features for Brain Tumor Classification. International Journal of Computer Science Issues, 9(3), 354-359. 2012 .

[6] B. Dhruv, N. Mittal and M. Modi. Study of Haralick's and GLCM Texture Analysis on 3D Medical Images. InternationalJournal of Neuroscience 129(4):1-29.2018.
DOI: 10.1080/00207454.2018.1536052.

[7] MA. Tahir, A. Bouridane and F. Kurugollu. An FPGA Based Coprocessor for GLCM and Haralick Texture Features and their Application in Prostate Cancer Classification. Analog Integrated Circuits and Signal Processing. 43(2), 205-215. 2005.

[8] M. Karimi, M. Fathl, Z. Sheykholeslam, B. Sahralyan and F. Naghipoor. Effect of Different Processing on Quality Factors and Image Texture Features of Bread. Journal of Bioprocessing & Biotechniques 2:127. 2012. DOI: 10.4172/2155-9821.1000127.

[9] I. Arzate-Vázquez, JJ. Chanona-Pérez, MJ. Perea-Flores, G. Calderón Domínguez, MA. Moreno-Armendáriz, H. Calvo, S. Godoy-Calderón, R. Quevedo and G. Gutiérrez-López. Image Processing Applied to Classification of Avocado Variety Hass (Persea americana Mill.) During the Ripenning Process. Food Bioprocess Technology 4: 1307.2011. DOI: 10.1007/s11947-011-0595-6.

[10] G. ElMasry, N. Wang, A. ElSayed and M. Ngadi. Hyperspectral imaging for nondestructive determination of some quality attributes for strawberry. Journal of Food Engineering 81:1, 98-107. 2007. DOI: 10.1016/j.jfoodeng.2006.10.016.

[11] D. Joseph Sampson, Y. Ki Changa, H.P. Vasantha Rupasinghe, Q. UZ Zaman. A dual-view computer-vision system for volume and image texture analysis in multiple apple slices drying. Journal of Food Engineering 127, 49-57. 2014. DOI: 10.1016/j.jfoodeng.2013.11.016.

[12] P. Mishra, A. Nordon, MS. Mohd Asaari, G. Lian and S. Redfern. Fusing spectral and textural information in near-infrared hyperspectral imaging to improve green tea classification modelling. Journal of Food Engineering 249, 40-47. 2019. DOI:10.1016/j.jfoodeng.2019.01.009.

[13] DM. Martínez Gila, P. Cano Marchal, J Gómez Ortega and J. Gámez García. Expert System for Monitoring the Malaxing State of the Olive Paste Based on Computer Vision. Sensors 18,2227. 2018. DOI:10.3390/s18072227.

[14] D. Aguilera Puerto, O. Cáceres Moreno, DM. Martínez Gila, J Gómez Ortega and J Gámez García. Online system for the identification and classification of olive fruits for the olive oil production process. Journal of Food Measurement and Characterization 13,716-727. 2019. DOI:10.1007/s11694-018-9984-0.

[15] B. Pathak and D. Barooah. Texture Analysis based on the Gray-Level Co-ocurrence Matrix Considering Possible Orientations. International Journal of Advanced in Electrical, Electronics and Instrumentation Engineering 2:9, 4206-4212. 2013.

[16] SW. Zucker and D. Terzopoulos. Finding structure in Co-occurrence matrices for texture analysis. Computer Graphics and Image Processing 12:3, 286-308. 1980. DOI: 10.1016/0146-664X(80)90016-7.

[17] S.A. Matz. Food Texture. Westport, Connecticut: AVI Publ. Co., Inc. 1962.

[18] D. Barranco, R. Fdez.-Escobar, L. Rallo. El Cultivo del Olivo. Ediciones Paraninfo S.A. Madrid (España). 2017.

[19] Abir Sadkaoui. Influence of Fruit Characteristics and Olive Paste Preparation Conditions on Process Yield ofVirgin Olive Oil. PhD Thesis. Jaén University (España). 2017.
http://hdl.handle.net/10953/882.

Inteligencia Artificial 22 (64) 143-151
doi: 10.4114/intartif.vol22iss64pp143-151

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Computational Model for Organizational Learning in Research And Development Centers (R&D)

Marco Javier Suárez Barón[1], José Fdo. López[2], Carlos Enrique Montenegro-Marin[3] Paulo Alonso Gaona García [4] and Franklin Guillermo Montenegro-Marin [5]

[1] UPTC University, Faculty of Systems and Computing, Sogamoso, Colombia – mailto:marco.suarez@uptc.edu.co

[2] FUSM, Bogotá, Colombia – mailto:jflopezq@hotmail.com

[3] Universidad Distrital Francisco José de Caldas, Bogotá, Colombia - cemontenegrom@udistrital.edu.co

[4] Universidad Distrital Francisco José de Caldas, Bogotá, Colombia – pagaonag@udistrital.edu.co

[5] Universidad de Cundinamarca, Soacha, Colombia – fmontenegro@ucundinamarca.edu.co

Abstract This work explains for a computational model design focused organizational learning in R&D centers. We explained the first stage of this architecture that enables extracting, retrieval and integrating of lessons learned in the areas of innovation and technological development that have been registered by R&D researchers and personnel in social networks corporative focused to research. In addition, this article provides details about the design and construction of organizational memory as a computational learning mechanism within an organization. The end result of the process is discusses the management of the extraction and retrieval of information as a technological knowledge management mechanism with the goal of consolidating the Organizational Memory.

Keywords: Computational Architecture, Strategic Knowledge Management, Social Networks.

1 Introduction

Social networks are currently considered as the main tool for sharing information and data. In this research, information and data will be taken from lessons learned contained in specialized social networks focused on research and such as Research gate[1], LinkedIn[6] and Blogs[1]. Nevertheless, most of the research experiences and knowledge are not registered or used, and information is not being properly exploited in those networks.

As a solution for that, the use of computational models like Machine Learning or Deep Learning enables the structuring and integration of specialized knowledge acquired from significant experiences, such as lessons learned[7]. The application of these models allows for greater flexibility in the acquisition process and facilitates the capture, recovery, transfer, and reuse of knowledge. Implementing these technological platforms will provide the entire organizational structure with a crucial tool for decision-making and strategic planning on R & D issues.

The paper is structured as follows: Section 2 describes the theoretical background, which involves Knowledge Management (KM) process and methods, models to manage knowledge on social systems and Learning technologies and organizational strategies to exchange knowledge; Section 3 details our proposed framework to combine and summarize research information and data to obtain final lessons learned; Section 4 presents results and discussion on the main components and phases of this computational model desing. Finally, in Section 5, we present our main conclusions and further works.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

2 Background

The generation of new knowledge is used for decision-making in non-simulated and simulated environments within the Learning process in the network. Planned R & D entities are thus created to optimize processes, reduce costs, increase innovation, and consider new projects. Specifically, these entities will be the context in which individuals and the organization will learn more [2].

Despite scientific advances on this subject, there are still gaps in the analysis of information from social networks of investigative type whose content specifically covers issues related to science, technology and innovation. The social networks studied in this article and the information registered in these networks correspond specifically to academic, business and scientific social networks, although, for this work we used information contained in Twitter social network for the extracting and integrating process.

The application of Machine Learning enables the structuring and integration of specialized knowledge acquired from significant experiences, such as lessons learned [3]. The application of these models allows for greater flexibility in the acquisition process; it facilitates the capture, recovery, transfer, and reuse of knowledge.

Social media has increased interest in our daily activities, and the user profile of each individual is considered a significant source of information [3]. Both Web sites and social networks are potential tools for the management, updating, and exchange of information and knowledge in fields that are interested in knowing the knowledge. The basic, thoughts, ideas, relationships and activities of each individual in their environment, such as marketing.

On the basis of this concept, Knowledge Management (KM) theories center on mechanisms to help maintain knowledge within an organization. According to [4], as KM theory evolved, different models were proposed for innovation management in companies from multiple sectors in France and Germany, which have led us to focus our work primarily on the concept of Personal Knowledge Management (PKM), one of the most recent lines of work in this field [5].

3 Methodology

To further boost competitiveness in an R&D center, the organization needs to learn from its current and past experiences in R&D. Additionally, it is advisable to take advantage of their experiences, physical repositories in this domain of knowledge since a large volume of information and trends in R + D + i has been published in corporate social networks.

Additionally, the process of designing the proposed architecture revealed two methodological perspectives in the existing technologies to support knowledge administration systems, which correspond to two discrete dimensions of knowledge management, according to [18] and to [19]. These two perspectives were used as the methodology for producing this paper and are explained below.

Firstly, the proposed knowledge management system is based on the process-centered perspective, understanding knowledge management as primarily a social communication process that can be improved by considering aspects of support to collaborative group work [14].

In the computational architecture, the process-centered approach focuses on individuals, as the most significant source of knowledge within an organization, and upholds the idea of resolving the cooperation problems amongst them through a process to achieve their social commitment to transfer and share knowledge.

The second approach is known as the product centric perspective [19], which in this project is directed at the creation, storage, and reuse of knowledge documents in the organizational memory, grounded in computer sciences. This approach was at the core of this research project, primarily because it is based on the idea of explicitly stating, documenting, and formalizing knowledge for use as a tangible resource, and attempts to present correct sources of information to the final users at the appropriate time.

Organizational memory should provide mechanisms for storage and use of all formal and informal knowledge that exists within the organization [2]. Organizational memory derives from documents, good practice guides, manuals, and books that help improve the performance of the members of the organization.

Through this approach, we noted the importance that knowledge management has gained, even from a strictly economic perspective, which has led to the rise of numerous information technology-based tools. These tools provide mechanisms for shaping the individual knowledge of employees into the collective knowledge of the community [20]. Table 1 shows how the proposed architecture differs from other learning models, given the lack of consensus on the definition of many of the concepts and terms used in the organizational learning [21] and knowledge management fields.

Table 1. Contributions to Scientific Knowledge in Research. Source: Authors

Component	Functional features and original scientific contributions
Metamodel	Integrates three levels of tasks (processing, knowledge, and learning), while the current models do not integrate these three functional levels.
Ontology	Ontology structure.
Social Analysis	Facilitates the necessary changes and transformations in R&D centers
Extraction and retrieval of lessons learned	Applied in syntactic/morphological analysis of the grammatical structure in texts that contain lessons learned.
Semantic analysis of lessons learned	Assists in self-correlation and establishing relationships of similarity and contrast between the strategies.
Organizational Memory (OM)	Real time creation of the OM or ranges defined by the user based on a model of creation of semantically analyzed packets of information.
Machine Learning	Application of information analysis models and application of algorithms and methods for text analysis

4 Foundations and Discussion

4.1 Framework Design

In the framework purposed, the process-centred approach focuses on individuals, as the most significant source of knowledge within an organization, and upholds the idea of resolving the cooperation problems amongst them through a process to achieve their social commitment to transfer and share knowledge. The basic methods used in this approach, such as Computer-Supported Cooperative Work, Workflow Administration, or processes training, among others, seek to foster communication and collaboration between individuals [24].

The framework used in this research project enabled the authors to standardize concepts, practices, and criteria to apply to the proposed metamodel and served as a reference for confronting and resolving new test cases of a similar nature.

This framework also includes the promotion of new forms of knowledge capture, based on sources of information, such as lessons learned, that circulate in social networks. The generation of new knowledge is used for decision-making in non-simulated and simulated environments within the Learning process in the network. Planned R & D entities are thus created to optimize processes, reduce costs, increase innovation, and consider new projects.

Specifically, these entities will be the context in which individuals and the organization will learn more. The framework objectives described above are summarized in the functional components presented in Figure 1.

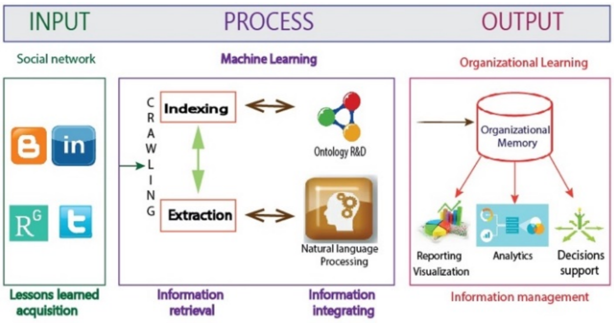
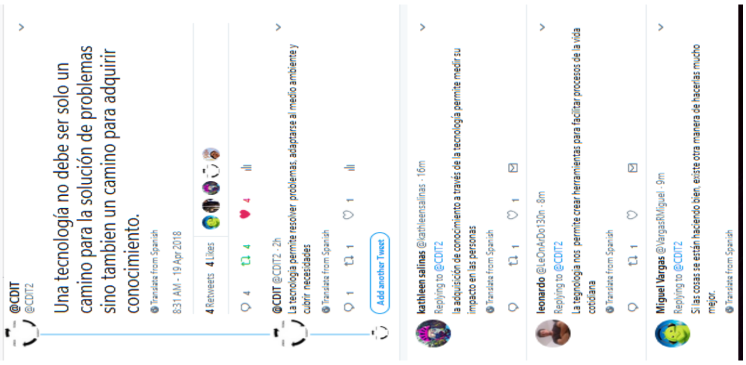


Figure. 1. Learning and organizational knowledge framework. Source: Authors

Lessons learnt acquisition.

In order to register information, we propose individual knowledge management (from tacit Knowledge to Explicit Knowledge). In this paper the information stored is called as lessons learned through social networks. Therefore, creating profiles for each individual or group of people is imperative for knowledge generation. On the basis of real time information retrieval (IR) algorithms, textual information is acquired and analyzed for lessons learned in the ranges or periods of time established by the users.

The acquisition of a lesson learnt, in the architecture represents the relationship between the result of a process, project, indicators, conditions or causes that align to the strategic plan of R & D for research center. The Figure. 2 shows and example of lessons learned registered in social network twitter.



@CDIT
@CDIT2

Una tecnología no debe ser solo un camino para la solución de problemas sino tambien un camino para adquirir conocimiento.

Translate from Spanish

8:31 AM - 19 Apr 2018

4 Retweets 4 Likes

4 4 4

@CDIT @CDIT2 · 2h
La tecnología permite resolver problemas, adaptarse al medio ambiente y cubrir necesidades
Translate from Spanish
1 1 1

Add another Tweet

kathleen salinas @kathleensalinas · 16m
Replying to @CDIT2
la adquisición de conocimiento a través de la tecnología permite medir su impacto en las personas
Translate from Spanish
1 1

leonardo @LeOnArDo130n · 8m
Replying to @CDIT2
La tegnologia nos permite crear herramientas para facilitar procesos de la vida cotidiana
Translate from Spanish
1

Miguel Vargas @VargasRMiguel · 9m
Replying to @CDIT2
Si las cosas se están haciendo bien, existe otra manera de hacerlas mucho mejor.
Translate from Spanish

Figure. 2. Example of lessons learned registered in social network twitter.

To carry out the information extraction process, an application has been implemented for these three social networks; the application based on Python-social-auth technology allows the development in an agile way and provides the connection to numerous social networks with little configuration of parameters. The framework is integrated with certain profiles, this application allows access to tweets, retweets and mentions that refer to textual structures of topics related to R & D lessons, the text structures are identified with a # hashtag that will be defined by the research group or groups of researchers associated with the R + D centers.

The mathematical model applied to obtain the associated trends (A, P, D) is showed in (1). The model analyze each lesson learned as an entity named defined "category" taken in ontology R&D [20]. An example of this category can be resources, dates, places or processes.

$$A_{c} = \sum_{n=1}^{\infty}(A,P,D) \quad \forall_{n} > 0 \quad (1)$$

Where:

P ➔*Weight (I like it, comments*): evaluate the number of Likes or retweets mad linked to each lesson registered.

D➔ *Registration time*= Determine the line time from lesson registered to first response; e.g. hours, days, minutes.

n➔Number of arcs= Represent the thread or sequence for each lesson learnt.

Ac➔ *Relevant publications*= Similarity R&D terms for P, e.g. Synonyms, folksonomies, Hashtags.

$A\chi$ ➔ Identify the content relevance for extracting. If the relation is equal to zero, then the lesson learnt is not candidate for acquisition.

A scenario of analysis is give in the Table 1. In this case, the lesson "*#CDDI Una tecnología no debe ser solo un camino para la solución de problemas sino también un camino para adquirir conocimiento.*" is retrieved from twitter social network, see Figure. 2.

The relevance result is:

Table 2. Analysis of relevance for lessons learnt acquisition process.

Lesson	Social Net	P	D	Ac	n	$A\chi$
1	Twitter	1	1hour	3	4	3
Thread 1.1	Twitter	1	3 min	2	1	2
Thread 1.2	Twitter	0	11 min	1	0	0
Thread 1.3	Twitter/Facebook	0	0	1	0	0

The results can then be used to calculate aggregations, identify trends and produce reports, dashboards and performance measures.

Retrieval and Integrating

This component of the computational architecture makes it possible to determine the set of categories, groups, and trends related to the current status of knowledge acquisition and management on R&D-related issues from lessons learned that have been structured through web service extraction. This process requires the application of linguistic techniques, using natural language processing [1].

However, the source data for this process is nominal data and unstructured text containing information on the concepts, profiles, categories, description, codes, events, and control, along with the terminology of the set of lessons learned in knowledge management for R&D.

The information integration process involves a level of processing in which the application of ontology is crucial, given that the ontology enables the integration of specialized vocabulary into the knowledge domain. The tasks of indexing terms and linguistic concepts involved in ontologies make it possible to classify the topics, categories, entities (persons), and attributes of the entities mentioned in the lessons learned that are extracted.

Figure. 3 shows how the vocabulary "corpus" of data ontology allows the semantic indexing of scenarios such as: HR training, prototypes, patents, scientific articles, software registers. After the retrieval process all lessons learned are integrated and stored in a NoSQL structure.

For example, the word "management" can be changed in the word "administration"; To solve this lexical problem, this research adapts two approaches to the method of lexical variation ontological "lexical variation ontology". First, the method is applied in the English language corpus; in this case the variation must be made to a new corpus adaptable to the Spanish language since the language in which the lessons learned are recorded is Spanish. The main objective is to present a lexical ontology acquisition method that allows the variation of the noun and the verb through the generation of the corpus and the integration with the ontology of R + D data.

The other hand, the grammatical decomposition aims to understand the semantic behavior of each word as an entity contained in the R & D data ontology; Terms such as articles, connectors, links are discarded in the analysis process since they are not part of the set of terms included in the Ontology.

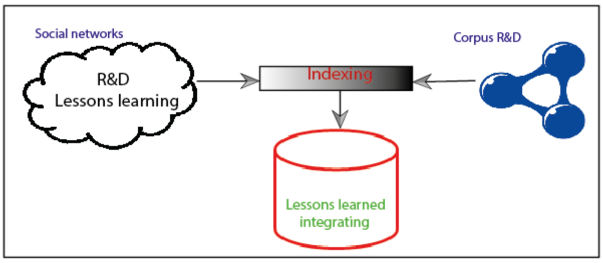


Figure. 3. R&D organizational learning process.

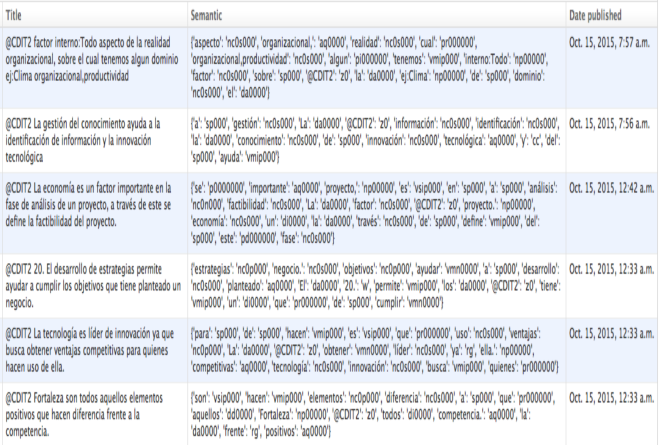
The model requires machine learning techniques for social analysis is a mechanism that is thought-out to implementation in the second stage of this project., this is a We Development an experimental non-probabilistic prediction prototype focused unstructured information lexical analysis; lessons recorded in social networks for corporative environments, which can be used and extended to other types of organizational R&D structures, either government or private.

Information Management (I.M)

This component involves the storage subsystem that offers the opportunity to integrate the necessary repositories and supports about the structural conformation of the lesson learned into the computational architecture. The information initially captured on corporative social networks allows real time collection of lessons learned and documents from each social network [2]. In the architecture, the I.M proposes the collection of information packages from Research gate, blogs, LinkedIn, and digital repositories; this workflow is supported by information integrating repository explained in Figure. 3.

The type of information to be considered for extracting and social analysis stems from the tacit and explicit knowledge of the R&D staff. Within the organization, is relevant the organizational maturity regarding the use and application of corporate social networks as a collaborative tool for organizational knowledge transfer.

Figure. 4 displays the standardized interface in order to optimize the ability to search, retrieve and analyze the texts of lessons learned extracted. In this case, the capture and extraction of texts from the twitter social network is presented. Through the use of text and semantic analysis techniques, like Latent Semantic Indexing, LSI, [22], it is possible to learn about the trends and reality of the knowledge that is being generated by the work teams, using the dissemination of lessons learned from each member of these teams. The result involves entities and concepts that are analyzed lexically and syntactically. Meanwhile, the semantic (structural) analysis given to each learned lesson make it possible to identify entities (see section 4.2) that are or are not contained within the R + D vocabulary.



Title	Semantic	Date published
@CDIT2 factor interno:Todo aspecto de la realidad organizacional, sobre el cual tenemos algun dominio ej:Clima organizacional,productividad	{'aspecto': 'nc0s000', 'organizacional,': 'aq0000', 'realidad': 'nc0s000', 'cual': 'pr000000', 'organizacional,productividad': 'nc0s000', 'algun': 'pi000000', 'tenemos': 'vmip000', 'interno:Todo': 'np00000', 'factor': 'nc0s000', 'sobre': 'sp000', '@CDIT2': 'z0', 'la': 'da0000', 'ej:Clima': 'np00000', 'de': 'sp000', 'dominio': 'nc0s000', 'el': 'da0000'}	Oct. 15, 2015, 7:57 a.m.
@CDIT2 La gestión del conocimiento ayuda a la identificación de información y la innovación tecnológica	{'a': 'sp000', 'gestión': 'nc0s000', 'La': 'da0000', '@CDIT2': 'z0', 'información': 'nc0s000', 'identificación': 'nc0s000', 'la': 'da0000', 'conocimiento': 'nc0s000', 'de': 'sp000', 'innovación': 'nc0s000', 'tecnológica': 'aq0000', 'y': 'cc', 'del': 'sp000', 'ayuda': 'vmip000'}	Oct. 15, 2015, 7:56 a.m.
@CDIT2 La economía es un factor importante en la fase de análisis de un proyecto, a través de este se define la factibilidad del proyecto.	{'se': 'p0000000', 'importante': 'aq0000', 'proyecto,': 'np00000', 'es': 'vsip000', 'en': 'sp000', 'a': 'sp000', 'análisis': 'nc0n000', 'factibilidad': 'nc0s000', 'La': 'da0000', 'factor': 'nc0s000', '@CDIT2': 'z0', 'proyecto.': 'np00000', 'economía': 'nc0s000', 'un': 'di0000', 'la': 'da0000', 'través': 'nc0s000', 'de': 'sp000', 'define': 'vmip000', 'del': 'sp000', 'este': 'pd000000', 'fase': 'nc0s000'}	Oct. 15, 2015, 12:42 a.m.
@CDIT2 20. El desarrollo de estrategias permite ayudar a cumplir los objetivos que tiene planteado un negocio.	{'estrategias': 'nc0p000', 'negocio.': 'nc0s000', 'objetivos': 'nc0p000', 'ayudar': 'vmn0000', 'a': 'sp000', 'desarrollo': 'nc0s000', 'planteado': 'aq0000', 'El': 'da0000', '20.': 'w', 'permite': 'vmip000', 'los': 'da0000', '@CDIT2': 'z0', 'tiene': 'vmip000', 'un': 'di0000', 'que': 'pr000000', 'de': 'sp000', 'cumplir': 'vmn0000'}	Oct. 15, 2015, 12:33 a.m.
@CDIT2 La tecnología es líder de innovación ya que busca obtener ventajas competitivas para quienes hacen uso de ella.	{'para': 'sp000', 'de': 'sp000', 'hacen': 'vmip000', 'es': 'vsip000', 'que': 'pr000000', 'uso': 'nc0s000', 'ventajas': 'nc0p000', 'La': 'da0000', '@CDIT2': 'z0', 'obtener': 'vmn0000', 'líder': 'nc0s000', 'ya': 'rg', 'ella.': 'np00000', 'competitivas': 'aq0000', 'tecnología': 'nc0s000', 'innovación': 'nc0s000', 'busca': 'vmip000', 'quienes': 'pr000000'}	Oct. 15, 2015, 12:33 a.m.
@CDIT2 Fortaleza son todos aquellos elementos positivos que hacen diferencia frente a la competencia.	{'son': 'vsip000', 'hacen': 'vmip000', 'elementos': 'nc0p000', 'diferencia': 'nc0s000', 'a': 'sp000', 'que': 'pr000000', 'aquellos': 'dd0000', 'Fortaleza': 'np00000', '@CDIT2': 'z0', 'todos': 'di0000', 'competencia.': 'aq0000', 'la': 'da0000', 'frente': 'rg', 'positivos': 'aq0000'}	Oct. 15, 2015, 12:33 a.m.

Figure. 4. Standardized interface of lessons learnt extracted using crawling

After extracting and filtering the lessons learned from unstructured sources, such as blogs, tweets, and organizational forums, the next stage is to create an information management component for constructing and organizing the organizational memory (OM). This is a continuous process and is at the core of the proposed platform. The lessons learnt filtering is supported by use of semantic indexing, in our approach the tool applied was ontology R&D [1].

The tasks of filtering and integrating information or lessons learned are based on the R&D ontology. The task of populating the organizational memory will be based on topics related to innovation and technological development for an R&D center. The purpose of designing the OM is to structure informal, case-based information. The OM also facilitates the automatic capture, retrieval, transfer, and reuse of knowledge. In information management, OM is defined as a flexible structure that enables the consolidation [23], in one sole repository, of all lessons learned on issues relevant to the R&D knowledge domain. Therefore, the design of the OM begins with the individual memory of each member of the R&D center and concludes with the creation of the collective memory.

In view of the above, organizational learning allows us to understand the impact of the opinions and perceptions of the human resources of the R&D center in relation to certain knowledge or experiences, for example, technological management. The R&D center can carry out periodic, offline analyses, through reports prepared on the basis of an analysis of the data from the OM obtained and formalized in real time. The framework allows for the incorporation into this analysis of an immense amount of spontaneous and real time information from social networks, forums, and blogs, to assess their impact on the thematic trends and behaviors and, thus, rapidly reveal both critical events and competitive advantages.

The Information Management component receives all packages of content in specific intervals of time (for example, daily or weekly) and analyses them to identify what is being mentioned in the R&D center in relation to the technological and social variables, e.g. sentiments and emotions of what is being said about topics like technological management. The correlational analysis is combined with mathematical models and algorithms that accompany the factorial analysis. These two inputs can be applied to obtain the trends associated with each lesson learned in terms of the entity mentioned, the defined category, and relevant and non-relevant topics at the R&D center.

4.2 The organizational Learning Process

The organizational learning process in the proposed architecture involves all activities related to knowledge storage and retrieval, and provides support by creating document repositories, forums, among other tools, to provide access to knowledge that serves for decision-making purposes at any given moment; thus, running the organizational memory like a cycle of Knowledge Management processes. The way the organizational memory is structured can establish six (6) categories of organizational memory of section 3.

From the perspective of business modelling language, ontologies provide a precise description of the concepts of the R&D domain and the relationships between these concepts. Therefore, in organizational learning processes, ontology offers a basic vocabulary that is useful for strategic knowledge management and establishes two levels of abstraction: for knowledge management and for the representation of knowledge.

The most important function of an ontology is the need to reach a consensus on the knowledge of the domain within an organization, so that the knowledge represented is not the subjective perspective of an individual but, rather, is shared and accepted by a community committed to the principle of organizational culture, facilitating communication and interoperability amongst the members of the R&D Centre.

Finally, Figure. 5 shows an of example dashboard obtained of the information that we have obtained from the previous processes and that feed the "tableau tool" for the comparison of the trends of lessons learned regarding the strategic axes of the R + D centres in period time one month. The analysis shows that in September 2017 there was a greater opinion tendency on R & D Management (45.76%) as in the month of October 2017 (33.93%) and the trend of publications with respect to R + D projects is greater with 75.00%.

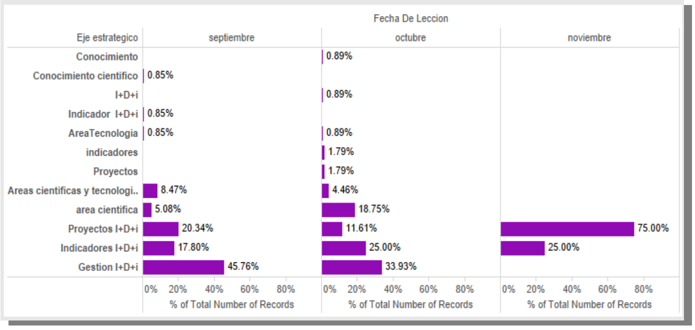


Figure. 5. Querying of the trends of lessons learned.

5 Conclusions and Future Research

In this paper is proposed the design of general architecture of a computational model driven to extraction, integrating and analysis not structured information obtained from scientific and academic social networks. The aim in this research is development an organizational learning system that apply new computational algorithms like natural language processing that allow organizational learning to be more effective and specific in R&D centres.

The validation of the learning metamodel demonstrate that the components of the architecture in the innovation, research development domain, allows indexing and integrating information from the lessons learned through data ontologies. In the metamodel, the ontology was conceived as the conceptual scheme of data in relation to a specific subject but not as a mechanism of knowledge acquisition, in this case were terms semantically related to R & D; this structure was also defined as a hierarchy of concepts that are characterized by properties and the relationship between all these entities. However, for the analysis of lessons from lessons learned with more complex lexical structures, the tokenization and standardization process is limited given that the set of labelling applied is still under testing for the Spanish language. From the point of view of the vocabulary used for indexing terms, the set of terms is limited to specialized concepts on technological surveillance, innovation, technological development and innovation. Finally, the ontology maintenance process is crucial in the growth of knowledge on the subject.

References

[1] Pico, B & Suárez, M.: Organizational Memory Construction Supported In Semantically Tagged. International Journal of Applied Engineering Research, 41744-41748 (2015).

[2] Kirwan, C.: Making Sense of Organizational Learning: Putting Theory Into Practice. Farnham: Gower Publishing Limited (2013).

[3] Chiha, R., Ayed, M. B.: Towards an Approach Based on Ontology for Semantic-Temporal Modeling of Social Network Data. In International Conference on Intelligent Systems Design and Applications, pp. 708-717. Springer (2016).

[4] Fam, D.: Facilitating communities of practice as social learning systems: a case study of trialling sustainable sanitation at the University of Technology Sydney (UTS). Knowledge Management Research & Practice, 391-399 (2017).

[5] Haas, M.R; Hansen M.T.: Different knowledge, different benefits: Toward a Productivity perspective on knowledge sharing in organizations. Strategic Management Journal, vol. 28, pp. 1133-1153 (2010)

[6] Liana Razmerita, Kathrin Kirchner, Frantisek Sudzina, (2009) "Personal knowledge management: The role of Web 2.0 tools for managing knowledge at individual and organisational levels", Online Information Review, Vol. 33 Issue: 6, pp.1021-1039, https://doi.org/10.1108/14684520911010981

[7] Tan, Wei ; Blake, M. Brian ; Saleh, Iman ; Dustdar, Schahram. / Social-network-sourced big data analytics. In: IEEE Internet Computing. 2013 ; Vol. 17, No. 5. pp. 62-69.

[8] Sinclaire, J. K., & Vogus, C. E.: Adoption of social networking sites: an exploratory adaptive structuration perspective for global organizations. Information Technology and Management, vol. 12(4), 293-314 (2011).

[9] Chow, W. S., & Chan, L. S.: Social network, social trust and shared goals in organizational knowledge sharing. Information & management, 45(7), 458-465 (2008).

[10] Takeuchi, Riki; A Critical Review of Expatriate Adjustment Research Through a Multiple Stakeholder View: Progress, Emerging Trends, and Prospects ; Journal of Management ; Vol 36, Issue 4, pp. 1040 – 1064; First Published January 26, 2010; https://doi.org/10.1177/0149206309349308.

[11] Pirró, G.; Mastroianni, C.; Talia, D.: A framework for distributed knowledge management: Design and implementation. Future Generation Computer Systems. vol 26, 38-49 (2010). doi: 10.1016/j.future.2009.06.004

[12] Myong-Hun, C., & Harrington, J.: Individual Learning and Social Learning: Endogenous Division of Cognitive Labor in a Population of Co-Evolving Problem-Solvers. Administrative Sciences, 53-75 (2013).

[13] Breslin, J.; Decker, S.: The future of social networks on the internet: the need for semantics. IEEE Internet Computing, vol. 11(6), 86-90, (2007). doi: 10.1109/MIC.2007.138.

[14] Fernández-Mesa, A., Ferreras-Méndez, J., Alegre, J., & Chiva, R.: Shedding New Lights on Organisational Learning, Knowledge and Capabilities. Newcastle: Cambridge Scholars Publishing (2014).

[15] López-Quintero, J., Cueva Lovelle, J., González Crespo, R., & García-Díaz, V.: A personal knowledge management metamodel based on semantic analysis and social information. Soft Computing, 1-10 (2016).

[16] Kamasat, R., Yozgat, U., & Yavuz, M.: Knowledge process capabilities and innovation: testing the moderating effects of environmental dynamism and strategic flexibility. Knowledge Management Research & Practice, 356-368 (2017).

[17] Espinoza Mejía, M., Saquicela, V., Palacio Baus, K., & Albán, H.: Extracción de preferencias televisivas desde los perfiles de redes sociales. Politécnico, 34(2), 1-9 (2014).

[18] Peis, E., Herrera Viedma, E., Montero, Y. H., & Herrera Torres, J. C.: Análisis de la web semántica: estado actual y requisitos futuros. El profesional de la información, 12(5), 368-376 (2003).

[19] Abecker, A.; Bernardi, A.; Hinkelmann, K.; Kuhn, O.: Toward a Technology for Organizational Memories. IEEE Intelligent, vol. 13(3), 40-48 (1998). doi: 0.1109/5254.683209.

[20] Suárez Barón,M.J. 2017. Applying Social Analysis for Construction of Organizational Memory of R&D Centers from Lessons Learned. In Proceedings of the 9th International Conference on Information Management and Engineering (ICIME 2017). ACM, New York, NY, USA, 217-220. DOI: https://doi.org/10.1145/3149572.3149604

[21] Barão, A., de Vasconcelos, J., Rocha, Á., & Pereira, R.: Research Note: A knowledge management approach to capture organizational learning networks. International Journal Of Information Management (2017). doi:10.1016/j.ijinfomgt.2017.07.013.

[22] Różewski, P., Jankowski, J., Bródka, P., & Michalski, R.: Knowledge workers' collaborative learning behavior modeling in an organizational social network. Computers In Human Behavior, 1248-1260 (2015).

[23] Van Grinsven, M., & Visser, M.: Empowerment, knowledge conversion and dimensions of organizational learning. The learning organization, 18(5), 378-391 (2011).

[24] Suárez Barón M.J., López J.F., Montenegro-Marin C.E., Gaona García P.A. (2018). Design of a Computational Model for Organizational Learning in Research and Development Centers (R&D). In: Simari G., Fermé E., Gutiérrez Segura F., Rodríguez Melquiades J. (eds) Advances in Artificial Intelligence - IBERAMIA 2018. IBERAMIA 2018. Lecture Notes in Computer Science, vol 11238. Springer, Cham

Inteligencia Artificial, 22(64) (2019), 152-165
doi: 10.4114/intartif.vol22iss64pp152-165

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Augmenting Scalable Communication-Based Role Allocation for a Three-Role Task

Gustavo Martins[1,3], Paulo Urbano[1,A]

[1]BioISI, Faculdade de Ciências da Universidade de Lisboa, Lisbon, Portugal.
[A]pub@di.fc.ul.pt
[3]Instituto de Telecomunicações, Lisbon, Portugal

Abstract In evolutionary robotics role allocation studies, it is common that the role assumed by each robot is strongly associated with specific local conditions, which may compromise scalability and robustness because of the dependency on those conditions. To increase scalability, communication has been proposed as a means for robots to exchange signals that represent roles. This idea was successfully applied to evolve communication-based role allocation for a two-role task, with one communication channel. However, it was necessary to reward signal differentiation in the fitness function, which is a serious limitation as it does not generalize to tasks where the number of roles is unknown a priori. We show that rewarding signal differentiation is not necessary to evolve communication-based role allocation strategies for the referred two-role task, and we improve reported scalability, while requiring less a priori knowledge. We extend the previous work to a three-role task and we propose and compare two cognitive architectures, to increase the number of communication channels, and several fitness functions to evolve scalable controllers. Our results suggest that communication might be useful to evolve role allocation strategies for increasingly complex tasks.

Keywords: role allocation, collective robotics, swarm, NEAT, artificial evolution, evolutionary robotics.

1 Introduction

Research on evolutionary collective robotics [2] [17] [20] for homogeneous robots suggest that role differentiation, which is fundamental for cooperation in natural and artificial systems, is triggered by differences in local physical interactions. In [2], a group of four robots was evolved to collectively navigate toward a light. In the most successful strategy, the robot in the front right position assumed the guide role, setting the path, while the robots in other positions followed. This strategy clearly depended on the specific positions of the robots in the group. In [17], a team of three robots was evolved for the ability to navigate as a group. The most successful strategy relied on two phases: (i) robots negotiate their positions until they reach a line formation; (ii) the first robot moves backwards while the others move forward following the first robot. If any robot is removed, the remaining robots cease motion. In [20], a group of five robots was evolved to guard a nest and forage simultaneously. The environment had two variations that required different behaviors to maximize fitness: (i) most robots stayed in the nest while others foraged; or (ii) fewer robots stayed in the nest while others foraged. Each variation had a corresponding nest color that robots could detect. The nest was divided into six sectors and each robot was randomly placed in a sector. The study concluded that the role each robot assumed depended on the nest color and the robot's position at the nest. For the above studies, the role assumed by a robot depends directly on very specific local conditions potentially decreasing robustness and scalability of the evolved strategies.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

To evolve scalable and robust solutions for role allocation, one study [8] proposed endowing robots with one communication channel, so that role allocation might be negotiated through the exchange of signals. Communication allowed robots to emit a signal with a numeric value in the range [0, 1] and the goal was to have one robot emitting a high value, and the others a low value. Although scalable solutions were evolved, there was no actual behavioral task for the robots to execute. Another study [9] further extends this work by introducing a two-role, double patrolling task (Figure 1), which we replicate, and further develop into a three-role task in our study.

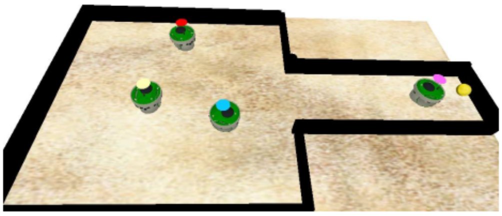


Figure 1: The double patrolling task environment. One robot must enter the corridor to find a light source while all others keep away. Reprinted with permission.

To evolve suitable controllers for the double patrolling task, Gigliota et al. [9] rewarded signal differentiation in the fitness function. This reward constitutes a priori knowledge, which is knowledge that is included by the system designer, to favor the evolution of a desired behavior but also constrain the evolutionary process [10]. Including less a priori knowledge imposes fewer restrictions on evolution, which can lead to more adaptive behaviors [14]. Furthermore, the authors [9] used a fixed-topology neuroevolution algorithm. However, different topologies result in different evolved behaviors and scalability for the same task [7]. To avoid the shortcomings of manual design [18], we substitute the fixed-topology neuroevolution algorithm by NEAT [19], a well known and widely applied, neuroevolution algorithm, that combines the search for appropriate network weights with a search for appropriate network topology.

Our goal is to evolve controllers with less dependency on a priori knowledge than required in previous research [9]. We aim to attain more robust and general solutions for the evolution of communication-based role allocation. We replicate the double patrolling task, and conduct experiments with a novel fitness function that does not reward signal differentiation. To determine the impact of the number of robots used during evolution on the scalability of the evolved solutions, we vary the number of robots used during evolution. Although communication-based role allocation was evolved in [9], the relevance of communication for performing the task and for evolving scalable strategies was not fully determined, because no control experiment without communication ability was conducted. To assess the relevance of communication for the task, we conduct experiments with no communication ability. We follow our previous work [11] and introduce a more challenging, three-role task, where two lights exist and which requires more than one communication channel. The goal is to have exactly one robot at each light while all other robots avoid approaching the lights. To increase the number of communication channels available to robots and evolve scalable controllers, we propose and conduct experiments with two cognitive architectures with two and three communication channels and five and six role sensors and actuators, while we also experiment fitness functions with and without reward for signal differentiation.

2 Related Work

Gigliota et al. [8] proposed the ability of robots to communicate their adopted roles by means of a numeric signal, to increase scalability of role allocation strategies. This approach facilitated the evolution of scalable strategies for the double patrolling task [9], where a group of robots has to allocate a sole robot to move to a light source while the other robots must stay away from light. In the most successful strategy, one robot emits a signal with a high value and assumes the role of exploring the environment to find the light while the other robots emit low values and avoid moving. Although evolution was conducted for groups of four robots, the evolved strategy was scalable for groups of 2-8 robots. The robots' controllers were neural networks evolved by means of an artificial evolutionary process. Artificial evolution depends on, amongst other factors, a fitness function to measure the quality of the evolved behaviors. Fitness function design is known to be a critical aspect of evolutionary robotics [3], [16], [15], [6].

Two fitness functions were used in the referred study [9], BF and CRF. The first, BF (Eq. 1), rewards a closer distance between light and the closest robot, while rewarding a wider distance between light and all other robots.

$$BF = 0.75 \times BFC1 + 0.25 \times BFC2 \tag{1}$$

$BFC1$ (Eq. 2) rewards controllers able to have one robot close to light while $BFC2$ (Eq. 3) rewards controllers able to have all other robots away from light. These components are computed as follows:

$$BFC1 = \frac{1}{T} \times \sum_{t}^{T} \frac{\max(0, (M - d_t(L, light)))}{M} \tag{2}$$

$$BFC2 = \frac{1}{T \times (N-1)} \times \sum_{t}^{T} \sum_{i}^{F} \frac{\min(M, d_t(F_i, light))}{M} \tag{3}$$

where $M = 0.9$ m is a maximal distance, $d_t(L, light)$ is the distance between light and its closest robot at instant t, T is the number of total time steps of a trial, F_i is robot i – excluding the closest robot to light, $d_t(F_i, light)$ is the distance between light and robot F_i at instant t and N is the number of robots.

The second fitness function, CRF (Eq. 5), extends BF to include a reward for signal differentiation, CFC: one robot emits a high signal and all other robots emit low signals.

$$CRF = 0.8 \times BF + 0.2 \times CFC \tag{4}$$

where CFC (Eq. 5) is computed as follows:

$$CFC = \frac{\sum_{t}^{T} \sum_{i}^{N} O_{t,max} - O_{t,i}}{T \times (N-1)} \tag{5}$$

where $O_{t,max}$ is the highest value emitted at instant t, $O_{t,i}$ is the value emitted by robot i at instant t, and N is the number of robots in the group.

To illustrate how these functions behave, let us assume one robot at 0.0 m from light, a second robot inside the corridor at 0.25 m from light and all other robots in the home area at 0.75 m from light. One robot emits a 1.0 signal and all others emit 0.0. Table 1 shows fitness values according to BF and CRF, for such a situation, with different numbers of robots.

Table 1: BF and CRF fitness values for different numbers of robots.

Robots	BFC1	BFC2	BF	CFC	CRF
2	1	0.28	0.82	1	0.86
4	1	0.65	0.91	1	0.93
6	1	0.72	0.93	1	0.94
10	1	0.77	0.94	1	0.95

As the number of robots increases, fitness increases, with no performance improvement, because the second robot inside the corridor weights less in the group of robots supposed to be in the home area. The higher the number of robots, the lower the negative impact on fitness of a second robot in the corridor and the less effective BF becomes to evolve suitable controllers. Furthermore, BF is inadequate to measure performance when the number of robots in a group may vary, because variation in the number of robots has an impact in the fitness value, without any change in performance. The CRF fitness function shares the above limitations because BF is a component of CRF.

In [9], with BF the authors were not able to evolve communication use or any strategy to perform the task. Only with CRF were they able to evolve communication use and a scalable strategy, suggesting that communication is a potential solution to support scalability for role allocation strategies. However, CRF rewards signal differentiation, which is undesirable because it forces a specific communicational scheme – one robot emitting a high signal and all other robots emitting low signals – hindering the system's ability to find other potentially suitable communicational schemes. We avoid such reward with the introduction of a novel fitness function presented in the next section (Eq. 6).

3 Two-Role Task

3.1 Evolving role allocation with less a priori knowledge

To evolve scalable strategies for the two-role task in [9], it was necessary to reward a signal differentiation scheme – one robot emits a high signal and all other robots emit low signals – in the fitness function, CRF. This approach, which resulted in the evolution of controllers able to perform the given two-role task, was possible because the authors were able to find, a priori, the above signal differentiation scheme which suits the two roles required for the task: one robot near the light and all other robots far away. However, as the number of roles needed to perform more complex tasks increases it is likely that the complexity of the required communicative behavior also increases. Therefore, it is unclear how rewarding signal differentiation could be used when a signal differentiation scheme that suits the roles required to perform the task can not be found a priori. As the task complexity increases, it becomes increasingly challenging to determine the specific signal differentiation scheme a priori to reward in the fitness function.

To avoid the reward for signal differentiation used in CRF and the dependency of BF on the number of robots, we introduce the TCD fitness function (Eq. 6). This function accounts for the existence of a sole robot in the corridor and the distance between light and the closest robot to light. TCD does not account for the number of robots as BF nor the communicative behavior as CRF.

$$TCD = \frac{1}{T} \times \sum_{t}^{T} \begin{cases} D_{t,light} & r = 1 \\ -D_{t,light} & r > 1 \\ 0 & r = 0 \end{cases} \tag{6}$$

where T is the number of time steps, r is the number of robots in the corridor (a robot is inside the corridor when its body center is inside the corridor) and $D_{\mathrm{t,light}}$ is determined as shown in Equation 7.

$$D_{t,light} = \frac{\max(0, (Range - d_t(L, light)))}{Range} \tag{7}$$

where $Range$ is the light sensor range and $d_t(L, light)$ is the distance between light and its closest robot, at instant t. The closer to light this robot is, the higher $D_{\mathrm{t,light}}$ is in range [0,1]. Fitness is $D_{\mathrm{t,light}}$ if there is only one robot in the corridor and $-D_{\mathrm{t,light}}$ when two or more robots are in the corridor. Otherwise, fitness is zero. To avoid negative fitness values, if the accumulated fitness up to instant t is less than zero, fitness is set to zero.

3.2 Experimental Setup

We use simulated e-puck [12] robots, that have a body diameter of 7.4 cm and distance between wheels of 5.2 cm. Robots have two independent wheel actuators to set the speed of each wheel, in [-0.1, 0.1] m/s, and a role actuator to emit a signal containing a decimal number in [0, 1]. Robots have eight obstacle sensors, equally spaced on the perimeter of the circular body, which measure the proximity of another robot or a wall, within 0.2 m; eight light sensors, also placed on the perimeter of the body, which measure the proximity to light, within 0.3 m; one non-directional role sensor with a range of 1.2 m which perceives the highest signal emitted by any other robot from any position; and a sensor to perceive the signal emitted by the robot itself in the previous time step. Each robot in the group is controlled by a copy of the same neural network and each input neuron receives values from one sensor and each output neuron sends values to one actuator. To simulate noise, each of these values is multiplied by a random number in range [0.95, 1.05]. Values coming from the sensors into the network are normalized in [0, 1] where closer proximity is represented by a higher value. Neural network output values are also normalized in [0, 1].

The experimental environment we replicated is composed of a 0.6x0.6 m area – the home – with an opening to a 0.2 m wide and 0.5 m long corridor (Figure 1). At the end of the corridor, there is a light source that robots cannot perceive from home. At the beginning of each experimental trial, robots emit a random signal and are placed in random positions and orientations in the home area, ensuring that no robots are colliding. Our experiments were run in the JBotEvolver [5] simulation platform.

In each experiment, we conducted 30 evolutionary runs of 1000 generations. The population has 100 individuals. In every generation, 15 trials with random initial conditions are generated and every individual in a generation faces the same set of trials. The fitness of an individual – a neural network – is the average fitness obtained in all those trials. A trial has a maximum duration of 2000 time steps. However, if a collision occurs the trial is terminated immediately to promote solutions that do not rely on or cause collisions. The NEAT implementation we used is Neat4J [13] with standard parameters.

In our experiments, we use the two fitness functions described above, BF (Eq. 1) and CRF (Eq. 4), as well as a new fitness function, TCD (Eq. 6). We conducted separate evolutions for groups of four robots, six robots, and ten and two robots where the numbers two and ten were randomly chosen by the simulator for each trial. In some experiments, we removed the robots' ability to communicate by removing the role sensor, as control – these experiments' names have the prefix ”noRole”.

3.3 Results

We post-evaluated the evolved controllers following the methodology used in [9] to allow for a direct comparison: only the controller that achieved the highest fitness during evolution is post-evaluated; the number of robots in the group varies between two and ten (we extended to twelve); each group with a different number of robots is post-evaluated in 100 trials; the post-evaluation function measures the percentage of time steps, within the last 100, when there is a sole robot in the corridor; satisfactory performance is one that shows a minimum post-evaluation fitness of 80%.

3.3.1 Evolution with Four Robots

In Figure 2, we show the post-evaluation results for groups evolved with four robots.

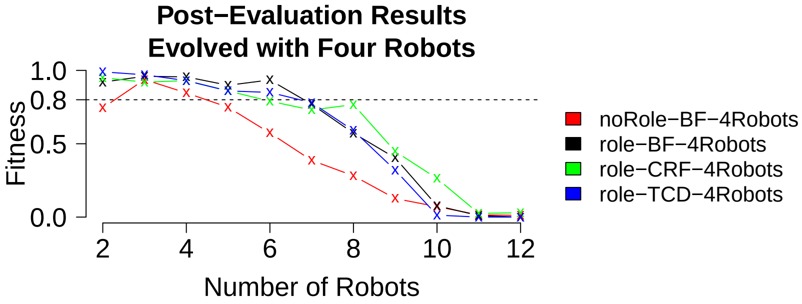


Figure 2: Post-Evaluation results when evolved with four robots

With no communication ability For the noRole-BF-4Robots controller, performance is above 80% for groups of three and four robots, showing that communication is not strictly necessary to perform the task. Robots move in straight paths from corner to corner. To avoid collisions, robots change paths and eventually one robot enters the corridor and moves to the light. With less robots in the home area, there are less paths interfering and thus, it is less probable that another robot enters the corridor. If another robot enters the corridor, it moves towards the light, perceives another robot at the light and leaves the corridor. This strategy does not scale because the higher the number of robots, the higher the probability that interferences occur causing robots to enter the corridor.

With communication ability Controllers role-BF-4Robots and role-TCD-4Robots, show a common strategy and a similar performance (Friedman $p = 0.179$), scaling for two to six robots. Robots move in random directions emitting a 0.0 signal, until a robot finds the light and emits a 1.0 signal. All other robots continue to emit 0.0, leaving the corridor if inside, and change their motion pattern to small orbits in place, decreasing the probability of entering the corridor. However, a robot might enter the corridor to avoid a collision. In such case, the robot moves to the light, detects another robot nearby and leaves. The higher the number of robots, the more challenging it is for robots to maintain an orbital path in the home area while avoiding collisions, and the higher the probability of extra robots entering the corridor.

The role-CRF-4Robots controller scales for groups of two to five robots and shows a different strategy: before any robot enters the corridor one robot emits a 1.0 signal – the leader – and all other robots – the followers – emit a 0.0 signal. The leader moves along the wall while the followers orbit in place. If the leader detects another robot in the way, it relays the leadership to the detected robot and becomes a follower. Eventually, the leader enters the corridor and finds the light. Exceptionally, a follower enters the corridor to avoid a collision, moves to the light and becomes a leader. The previous leader, still in the home area, becomes a follower. A similar relay strategy was evolved in [7], with CRF, where different topologies where manually chosen and evaluated. Interestingly, for the other top controllers, a different strategy was evolved. Robots move in random directions in the home area; a robot enters the corridor to avoid a collision, finds the light and emits a 1.0 signal. The other robots maintain their behavior but avoid the corridor entrance. As the number of robots increase, the ability to avoid the corridor entrance decreases due to path interferences between robots.

The strategy evolved with the BF fitness function is scalable for 2-6 robots while in the previous work [9], no strategy that performs the task was evolved with BF. This improvement over previous work, where a fixed-topology neuroevolutionary algorithm was used, illustrates how a non fixed-topology neuroevolutionary algorithm may be more powerful when it comes to explore the solutions space. Furthermore, scalability is observed only in the communicative controllers, which suggest that communication is a relevant factor to the evolution of scalable role allocation strategies.

3.3.2 Evolution with Six Robots

In Figure 3, we show the post-evaluation results for groups evolved with six robots.

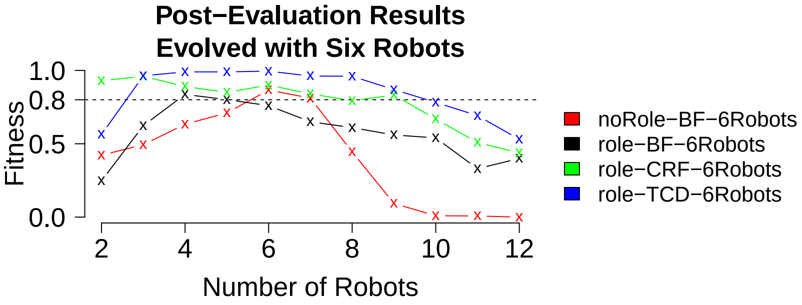


Figure 3: Post-Evaluation results when evolved with six robots

With no communication ability Controller noRole-BF-6Robots shows satisfactory performance for six and seven robots but does not scale. Robots describe small elliptical paths in place, avoiding other robots within the obstacles sensor range. In this process, one of the robots enters the corridor and moves to the light. If another robot enters the corridor, it detects the first robot and leaves.

With communication ability Controller role-BF-6Robots shows poor performance. When a robot finds the light and emits 1.0 all other robots also emit 1.0 and spin in place, even if that place is inside the corridor. This is a crucial difference to the previous communicative strategies evolved, where extra robots inside the corridor would leave. This strategy attained the highest fitness during evolution because the BF fitness function allows high fitness when more than one robot is inside the corridor, in spite of that behavior being the opposite of the desired.

For the role-CRF-6Robots and role-TCD-6Robots controllers, performance improved for seven, eight and nine robots, when compared to the 4-Robots experiments. For both controllers, the evolved strategy is similar to the main strategy described earlier: robots explore the environment until one robot finds the light and emits a 1.0 signal; if another robot is in the corridor, it leaves; robots in the home area change their behavior to avoid entering the corridor, but if another robot enters the corridor, it detects the robot at the light and leaves. Exceptionally, controller role-TCD-6Robots shows poor performance for two robots, because robots follow the walls instead of exploring in random directions. Robots cannot distinguish walls from fellow robots and when the group is composed of two robots distant from any wall, they follow each other in a circle as if they were following a wall, entering into a deadlock. Nevertheless, the strategy evolved with the TCD fitness function shows a higher post-evaluation fitness when compared to the strategy evolved with CRF (Friedman $p = 7.78 \times 10^{-5}$ for the null hypothesis of a similar performance for CRF and TCD).

3.3.3 Evolution with Two and Ten Robots

In Figure 4, we show the post-evaluation results for groups evolved with two and ten robots.

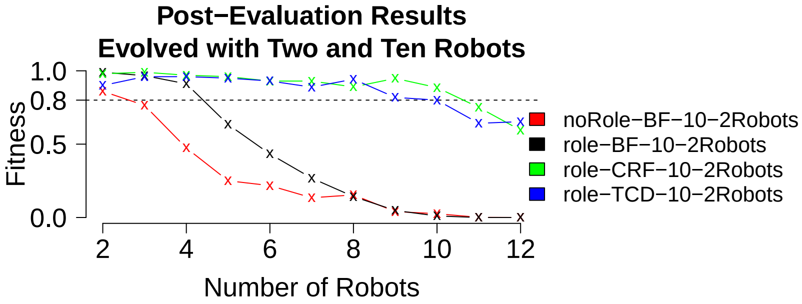


Figure 4: Post-Evaluation results when evolved with two and ten robots

With no communication ability For the noRole-BF-10-2Robots controller, robots follow the wall, in the home area. The first robot entering the corridor, moves to the light. The other robots always enter the corridor but leave after detecting the robot at the light. This strategy works well for two robots because the second robot spends more time exploring the home area before re-entering the corridor than inside the corridor, thus minimizing the time an extra robot spends inside the corridor. As the number of robots increases, though, the time extra robots spend inside the corridor, increases as well, resulting in a lower post-evaluation fitness.

With communication ability For the role-BF-10-2Robots controller, an alternative communication use has evolved. Robots explore the environment, emitting a 1.0 signal, instead of 0.0. When a robot moves to the light emits a 0.0 signal. If the group is composed of only two robots, the remaining robot in the home area changes the motion pattern to small orbits in place and the task is accomplished. However, if the group is composed of more than two robots, the robots in the home area do not change the motion pattern, for two reasons: (i) robots can only perceive the highest signal being emitted and, (ii) robots in the home area are emitting 1.0, the highest possible signal. Thus, the 0.0 signal being emitted by the robot at the light is not perceived by any robot. This instance illustrates how the desired solution was so deeply hardwired in the robots' design: by having a role sensor that only detects the highest signal being emitted, the system designers forced the robot at the light to use the highest possible signal, to inform it has found the light, in order to assure that the signal is perceived by the other robots. The communicational scheme evolved in this experiment was possible because the BF fitness function attains higher fitness for larger groups of robots, as illustrated in Table 1. In this experiment, the evolved communicational scheme attains high BF fitness for two and ten robots because it allows the task to be performed with precisely two robots and also because a group of ten robots is large enough to attain high fitness. In other words, according to BF, the evolved strategy is adequate for two robots and not so inadequate for ten robots. However, post-evaluation results show that, after all, the behavior is not advantageous for a high number of robots.

For the role-CRF-10-2Robots controller, the evolved strategy is the main communicative strategy described earlier because the CRF fitness function forces the desired communicational scheme. For the role-TCD-10-2Robots controller, when the robots at the home area receive the 1.0 signal emitted by the robot at light, they spin in place and avoid the corridor, as seen before. Controllers role-TCD-10-2Robots and role-CRF-10-2Robots show no statistically significant difference in performance (Friedman $p = 0.062$) and the highest post-evaluation performance of all experiments, increasing previously reported scalability in [9] of 2-8 robots to 2-10 robots. TCD requires less a priori knowledge and is thus preferable over CRF.

4 Three-Role Task

For the three-role task experiments, we changed the environment, evolutionary setup and the sensors and actuators. To increase the number of roles to three, we introduced a light at the left side of the environment, at 1.40 m from the right light. We also removed the walls to decrease the duration of the experiments. At the start of a trial, robots are randomly placed at the center, between the lights. The goal is that exactly one robot moves to each light while the other robots avoid lights. Figure 5 shows one possible initial state and Figure 6 shows one desired final state. Numbers represent IDs.

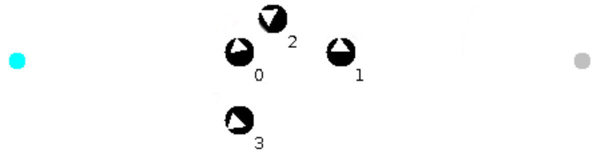


Figure 5: One possible initial position for the three-role task environment.

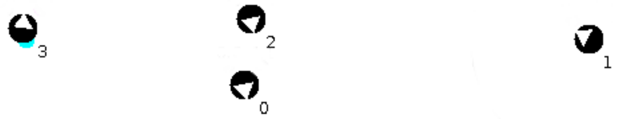


Figure 6: One desired final position for the three-role task environment.

On one hand, removing the walls decreases simulation time. On the other hand, it allows robots to wander off. To avoid robots wandering off, we increased the light sensors range to 2 m, allowing robots to use lights as reference points. An alternative to increase the light sensors range, would be to introduce an absolute reference point, and the corresponding sensor, which would increase the evolutionary search space and, thus, increase simulation time. Each robot has eight light sensors to detect the right light and eight light sensors to detect the left light. The role sensor range was increased to 2 m, because of the larger dimensions of the three-role task environment. The obstacle sensors range was decreased to 10 cm, as these sensors are used only to avoid collisions between robots and a larger range is unnecessary. The wheels actuators were maintained as in the two-role task. For each experiment, we conducted ten evolutionary runs, with 1500 generations, a population of 100 individuals, a trial duration of 400 steps and groups of four robots.

We experimented two architectures for increasing the number of channels: OneRoleAct, which relies on one role actuator and TwoRoleAct, which relies on two role actuators. In OneRoleAct, robots have one role actuator, one sensor to perceive the own role in the previous time step and three role sensors which are binary. Each role sensor is sensitive to a specific interval and outputs the value 1.0 when it detects a role actuator emitting in the interval it is sensitive to, or 0 otherwise. The first role sensor is sensitive to the interval $[0,\frac{1}{3}[$, the second sensor to $[\frac{1}{3},\frac{2}{3}[$ and the third sensor to $[\frac{2}{3},1]$. The three role sensors represent three channels. In TwoRoleAct, robots have two sets composed of one role actuator, one role sensor which perceives the highest value being emitted by role actuators within range and one sensor to perceive the own role in the previous time step. The two sets represent two channels and the actuators and sensors of a channel are unable to exchange information with the actuators and sensors of the other channel. For each architecture, we conduct two sets of experiments, in a total of four experiments. In each set of experiments we use two fitness functions: one behavioral fitness function, that considers only the positional behavior of the robots, and a fitness function that also considers signal differentiation.

Behavioral Fitness Function for OneRoleAct and TwoRoleAct The behavioral fitness function we used for OneRoleAct and TwoRoleAct, TCD_{lights}, shown in Equation 8, rewards a situation where one robot is close to a light, another robot is close to the other light while the other robots are away from any light (adapted from TCD, Eq. 6).

$$TCD_{lights} = \frac{1}{N} \times \sum_{n}^{N} TCD_{light_n} \tag{8}$$

where TCD_{lights} is the behavioral fitness for an environment with N lights and TCD_{light_n} is the behavioral fitness associated with light n, computed as shown in Equation 9.

$$TCD_{light_n} = \frac{1}{T} \times \sum_{t}^{T} \begin{cases} D_{t,light_n} & r = 1 \\ -D_{t,light_n} & r > 1 \\ 0 & r = 0 \end{cases} \tag{9}$$

where T is the number of time steps, r is the number of robots closer than 0.40 m to light n and $D_{t,light_n}$ is computed as shown in Equation 10.

$$D_{t,light_n} = \frac{\max(0, (K - d_t(L, light_n)))}{K} \tag{10}$$

where K is a maximum distance a robot may be from light n to increase fitness, arbitrarily defined as 0.30 m, and $d_t(L, light_n)$ is the distance between light n and its closest robot, at instant t. The closer to light n this robot is, the higher $D_{t,light_n}$ is in interval [0,1]. Fitness is $D_{t,light_n}$ if there is only one robot at a distance of at most 0.40 m from light n and $-D_{t,light_n}$ when two or more robots are closer than 0.40 m from light n. Otherwise, fitness is zero. To avoid negative fitness values, if the accumulated fitness up to instant t is less than zero, fitness is set to zero. We use the threshold 0.40 m because it was the value used for the two-role task, as it was the distance from the corridor entrance to light.

Signal Differentiation Reward Fitness Function for OneRoleAct The fitness function that includes a reward for signal differentiation for the OneRoleAct experiment, $Comm_{OneRoleAct}$, shown in Equation 11, rewards a situation where one robot emits in $[0,\frac{1}{3}[$, another robot emits in $[\frac{2}{3},1]$ and all other robots in $[\frac{1}{3},\frac{2}{3}[$.

$$Comm_{OneRoleAct} = 0.75 \times BF_{lights} + 0.25 \times CS \tag{11}$$

where BF_{lights_n} is computed as shown above in Equation 8 and CS is the component that rewards signal differentiation, according to the number of robots emitting a signal on each channel, computed as shown in Equation 12.

$$CS = \frac{1}{T} \times \sum_{t}^{T} O_t + D_t + H_t \tag{12}$$

where T is the number of time steps, $O_t = 1/3$ when there is exactly one robot emitting in $[0, \frac{1}{3}[$, otherwise is zero, $D_t = 1/3$ when all robots except two are emitting in $[\frac{1}{3},\frac{2}{3}[$, otherwise is zero, and $H_t = 1/3$ when there is exactly one robot emitting in $[\frac{2}{3},1]$, otherwise is zero.

Signal Differentiation Reward Fitness Function for TwoRoleAct The fitness function that includes a reward for signal differentiation for the TwoRoleAct experiment, $Comm_{TwoRoleAct}$, shown in Equation 13, rewards a situation where one robot emits a high value on the first channel and a low value on the second channel, one robot emits a low value on the first channel and a high value on the second channel and all other robots emit low values on both channels.

$$Comm_{TwoRoleAct} = 0.75 \times BF_{lights} + 0.25 \times CM \tag{13}$$

where BF_{lights} is computed as shown in Equation 8 and CM is the component that rewards signal differentiation, computed as shown in Equation 14.

$$CM = 0.75 \times \frac{1}{A} \times \sum_{a}^{A} SD_a + 0.25 \times \frac{1}{T} \times \sum_{t}^{T} R_t \tag{14}$$

where A is the number of channels, T is the number of time steps and R is a reward for not having the same robot emitting the highest signal on both channels. At each time step, when the highest signals on all channels are emitted by different robots, R_t is set to one, otherwise R_t is set to zero. We introduced R to increase the probability of evolving solutions where the highest signal on each channel is emitted by different robots. SD_a is the signal differentiation reward for channel a, computed as shown in Equation 15.

$$SD_a = \frac{\sum_{t}^{T} \sum_{i}^{N} O_{at,max} - O_{at,i}}{T \times (N-1)} \tag{15}$$

where $O_{at,max}$ is the highest value emitted at instant t, on channel a, $O_{at,i}$ is the value emitted by robot i, on channel a, at instant t, and N is the number of robots in the group.

4.1 Results

Figure 7 shows boxplots that represent the highest fitness controller for each run. The dotted lines represent the minimum fitness obtained when robots complete the task in an expeditious manner, i.e., after an initial phase with a duration of less than 80 steps, where role negotiation may happen, two robots move towards the lights while the other robots avoid the lights. Experiment names have the suffix "-Comm" when the signal differentiation reward is present and no suffix when the the reward is not present.

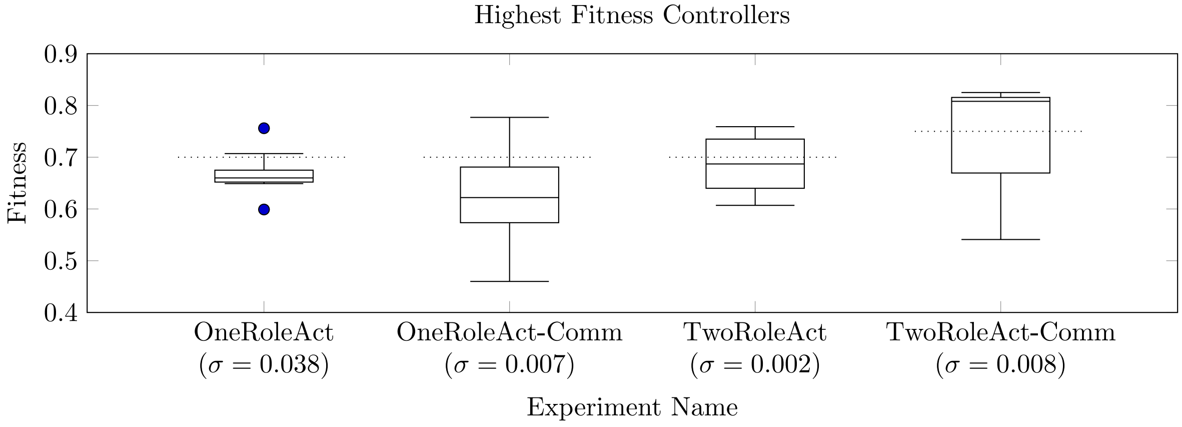


Figure 7: Boxplots for the highest performing controller for each run.

One controller able to perform above the minimum performance threshold evolved in all experiments. TwoRoleAct and TwoRoleAct-Comm evolved a higher number of controllers able to perform above the minimum threshold, when compared to OneRoleAct and OneRoleAct-Comm.

4.1.1 Post evaluation

We conducted 100 post evaluation trials for the best controller of each run. The post-evaluation function measures the percentage of time steps, within the last 100, when there is exactly one robot at a distance of 10 cm or less from each light and no other robots closer than 30 cm to any light. This three-role task post-evaluation function is more challenging than the two-role task post-evaluation function, as it requires robots to be closer to light – 10 cm instead of 50 cm. Satisfactory performance is one that shows a minimum post-evaluation fitness of 80%. Figure 8 shows the post-evaluation results for the controller which resulted in the highest post-evaluation fitness for each experiment.

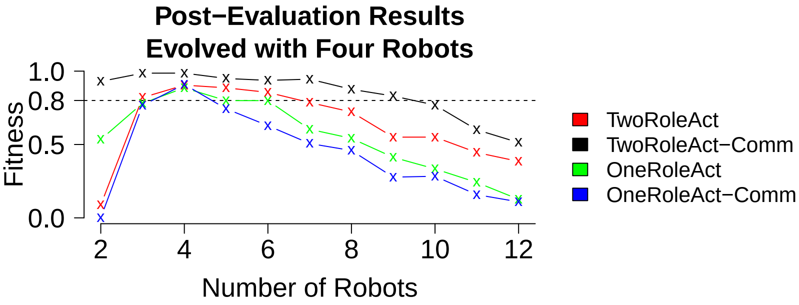
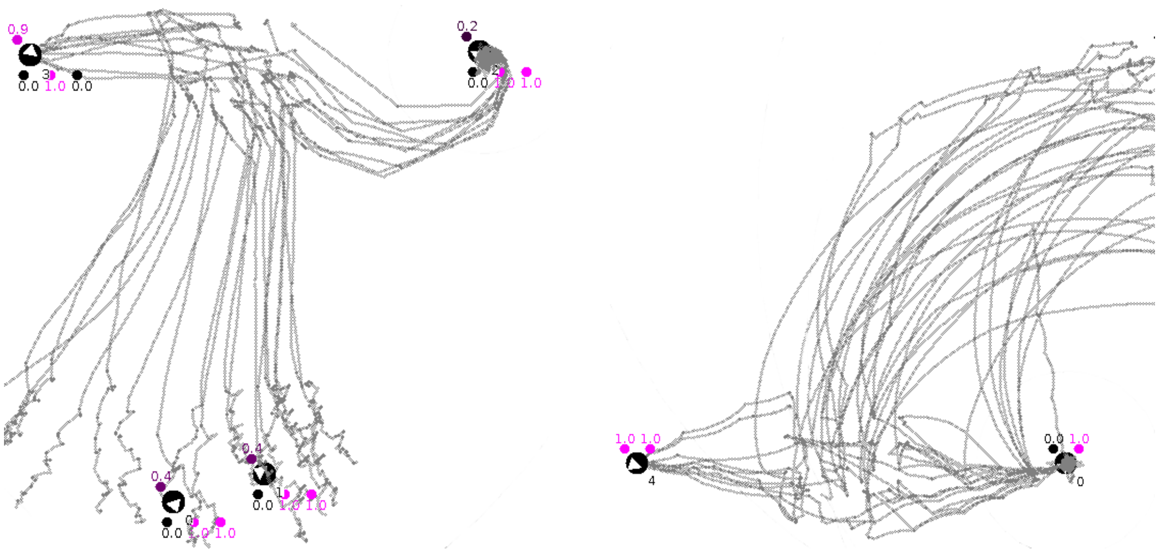


Figure 8: Post-Evaluation results for the three-role experiments.

For OneRoleAct and TwoRoleAct architectures we evolved controllers able to obtain 80% of post-evaluation fitness with four robots. However, only the TwoRoleAct controllers show scalability. With no signal differentiation reward, the TwoRoleAct controller scales to groups of three to six robots and when using the signal differentiation reward, TwoRoleAct-Comm, scales to groups of two to nine robots.

4.1.2 Behavior

The behavior of the highest post-evaluation fitness controllers, for ten trials, is shown in Figures 9 and 10. For OneRoleAct, decimal numbers represent the values detected by the role sensors. For TwoRoleAct, decimal numbers represent the values being emitted by the role actuators. The paths followed by the robots during the ten trials are represented by gray trails.

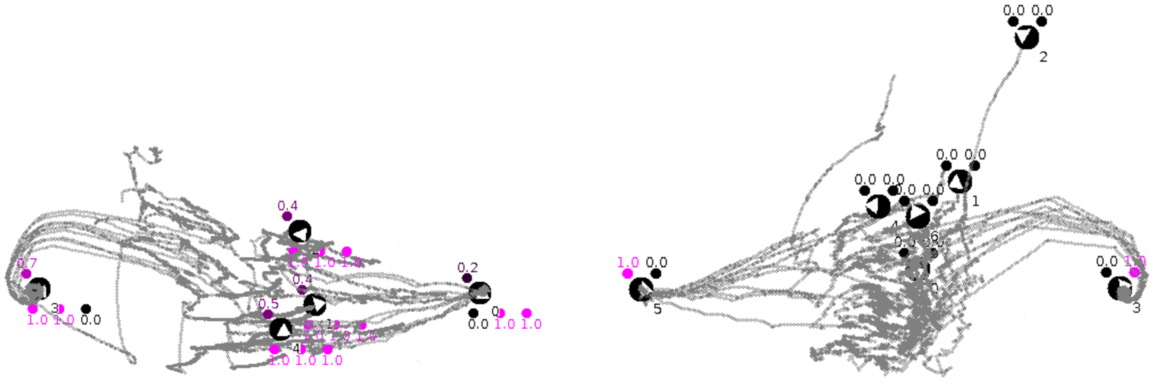


(a) OneRoleAct with 4 robots

(b) TwoRoleAct with 5 robots

Figure 9: Ten trials paths for OneRoleAct and TwoRoleAct with no signal differentiation reward.

For the OneRoleAct controller (Fig. 9 (a)), one robot emits a high value (≈ 1.0) and moves to the left light – at figure top left – while the other robots move South, until one of them emits a low value (≈ 0.2) and moves to the right light – at figure top right. The robots moving South emit ≈ 0.5 and stop at a distance where the lights are within the light sensor range. For the TwoRoleAct controller (Fig. 9 (b)), one robot emits high values (≈ 1) on both channels and moves to the left light while other robot emits low values (≈ 0.0) on both channels and moves to the right light – the two robots at figure bottom. The other robots emit a high value (≈ 1.0) on one channel and a low value (≈ 0.0) on the other channel and move North-East, to a distance where the lights are within the light sensor range – outside the figure. For OneRoleAct and TwoRoleAct, when the robots reach the positions at the lights, communication becomes unstable and we are unable to identify a pattern.



(a) OneRoleAct-Comm with 5 robots

(b) TwoRoleAct-Comm with 7 robots

Figure 10: Ten trials paths for OneRoleAct and TwoRoleAct with signal differentiation reward.

For OneRoleAct-Comm and TwoRoleAct-Comm, a trial begins with a negotiation phase where robots allocate roles. For the OneRoleAct-Comm controller, one robot emits a value in $[0, \frac{1}{3}[$, another robot emits a value in $[\frac{2}{3}, 1]$ and all other robots emit a value in $[\frac{1}{3}, \frac{2}{3}[$. After the negotiation phase, the robots emitting in $[\frac{1}{3}, \frac{2}{3}[$ decrease motion, while the other two robots separate from the group and move to the lights. For the TwoRoleAct-Comm controller, the negotiation phase proceeds until one robot emits a high value (≈ 1.0) with one of the two role actuators, another robot emits a high value (≈ 1.0) with the other role actuator and all other robots emit low values (< 0.4) in both role actuators. After the negotiation phase, the robots emitting high values move to the lights while the other robots decrease motion, or exceptionally move about 1 m North.

Results show that robots learn to associate communication patterns with behaviors. For instance, in TwoRoleAct, emitting a high value on a channel might mean moving to the right light and in OneRoleAct, receiving data simultaneously in all channels means to avoid any light. Furthermore, the relationship between a communication pattern and a behavior is evolved differently for different runs, suggesting that evolution is finding different ways to use communication for role allocation.

5 Discussion and Future Work

We substitute the fixed-topology neuroevolution algorithm, previously used in [9], by NEAT, which is a neuroevolution algorithm that evolves both topology and weights, and we are able to evolve scalable strategies where previously had not been possible. Our results suggest that future research should avoid fixed-topology neuroevolution algorithms.

For the two-role task, we show how to evolve communication-based scalable role allocation strategies without rewarding signal differentiation. We introduce a novel fitness function that does not reward signal differentiation, and demonstrate that it is simple to co-evolve the necessary communicative and non-communicative behavioral aspects, contrary to previous findings [9]. Furthermore, our fitness function is more adequate to measure the quality of the evolved strategies, as it does not suffer from the limitations identified in fitness functions used in earlier research [9]. We show that although communication is not strictly necessary to perform the task, it is a relevant factor to evolve scalable solutions.

Evolving communication is not trivial because evolution must produce both appropriate signals and corresponding reactions [1]. In our research, however, for the two-role task and for the three-role task, we evolve a communicative system without explicit selective pressure for communication use in the fitness function. Nevertheless, for the three-role task, the evolved communicative system shows limited scalability when the reward for signal differentiation is not present. When the signal differentiation reward is present, the evolved controllers show higher post-evaluation performance and scalability. Advances in the field of evolutionary robotics are achieved, amongst other ways, by designing systems able to perform increasingly complex tasks while minimizing the amount of a priori knowledge from the designer [4] [14]. We will research a fitness function that does not require rewarding signal differentiation, to avoid setting a priori a communicative strategy. Focus should be on defining fitness functions that accurately measure the quality of the evolved strategies and not how communication is used.

The presented results show that communication based role allocation might be useful as a role allocation strategy for two and three roles as well as for different numbers of robots. Although we show how two channels might be used to perform a three-role task, the relationship between the number of roles and the minimum number of channels is unclear. Our research suggests that one more channel is needed for every added role, which forces the designer to define a priori the number of roles and channels, thus limiting evolutionary exploration and the discovery of novel solutions. Therefore, we will research a higher decoupling between the number of roles and channels. Our aim is to build a cognitive architecture which allows a low number of channels and a high number of roles, possibly by combining the information on different channels.

Communication-based role allocation is a research path worth pursuing because it might offer scalability and robustness for cooperative multi robot systems. Our goal is to find a generalizable evolutionary setup to evolve scalable and robust communication-based role allocation. Therefore, we will research an evolutionary setup able to evolve strategies for three-role tasks without explicit selective pressure for communication use in the fitness function. We aim to identify the conditions for the emergence of communication-based role allocation strategies for increasingly larger numbers of roles and robots.

Acknowledgements

Fundação para a Ciência e Tecnologia grant SFRH/BD/94432/2013

References

[1] Christos Ampatzis, Elio Tuci, Vito Trianni, and Marco Dorigo. Evolution of signaling in a multi-robot system: Categorization and communication. *Adaptive Behavior*, 16(1):5–26, 2008.

[2] G. Baldassarre, S. Nolfi, and D. Parisi. Evolving mobile robots able to display collective behaviors. *Artificial Life*, 9(3):255–267, 2003.

[3] Mohammad Divband Soorati and Heiko Hamann. The effect of fitness function design on performance in evolutionary robotics: The influence of a priori knowledge. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO'15, pages 153–160, New York, NY, 2015. ACM.

[4] Stephane Doncieux and Jean-Baptiste Mouret. Beyond black-box optimization: a review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, 7(2):71–93, 2014.

[5] Miguel Duarte, Fernando Silva, Tiago Rodrigues, Sancho Moura Oliveira, and Anders Lyhne Christensen. JBotEvolver: A versatile simulation platform for evolutionary robotics. In *14th International Conference on the Synthesis and Simulation of Living Systems–ALIFE*, pages 1–8, New York, NY, 2014. MIT Press, Cambridge, MA.

[6] D Floreano and J Urzelai. Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4â5):431–443, 2000.

[7] Onofrio Gigliotta. Task allocation in evolved communicating homogeneous robots: The importance of being different. In *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*, pages 181–190, Cham, Switzerland, 2016. Springer International Publishing.

[8] Onofrio Gigliotta, Marco Mirolli, and Stefano Nolfi. Who is the leader? dynamic role allocation through communication in a population of homogeneous robots. *Artificial Life and Evolutionary Computation*, pages 167–177, 2010.

[9] Onofrio Gigliotta, Marco Mirolli, and Stefano Nolfi. Communication based dynamic role allocation in a group of homogeneous robots. *Natural Computing*, 13(3):391–402, September 2014.

[10] L. König. *Complex Behavior in Evolutionary Robotics.* Walter de Gruyter GmbH & Co KG, 2015.

[11] Gustavo Martins, Paulo Urbano, and Anders Lyhne Christensen. Using communication for the evolution of scalable role allocation in collective robotics. In Guillermo R. Simari, Eduardo Fermé, Flabio Gutiérrez Segura, and José Antonio Rodríguez Melquiades, editors, *Advances in Artificial Intelligence - IBERAMIA 2018*, pages 326–337, Cham, 2018. Springer International Publishing.

[12] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. volume 1, pages 59–65, Portugal, 2009. IPCB:Instituto PolitÃ©cnico de Castelo Branco.

[13] NEAT4J. Neat4j Java framework, 2006.

[14] Andrew L. Nelson, Gregory J. Barlow, and Lefteris Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, 2009.

[15] Stefano Nolfi. Evolutionary robotics: Exploiting the full power of self-organization. *Connection Science*, 10:167–183, 1998.

[16] Stefano Nolfi and Dario Floreano. *Evolutionary Robotics: The Biology, Intelligence,and Technology.* MIT Press, Cambridge, MA, USA, 2000.

[17] Matt Quinn, Lincoln Smith, Giles Mayley, and Phil Husbands. Evolving controllers for a homogeneous system of physical robots: structured cooperation with minimal sensors. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 361(1811):2321–2343, 2003.

[18] Kenneth O. Stanley and Risto Miikkulainen. Efficient reinforcement learning through evolving neural network topologies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, page 9, San Francisco, CA, 2002. Morgan Kaufmann.

[19] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[20] Elio Tuci, Boris Mitavskiy, and Gianpiero Francesca. On the evolution of self-organised role-allocation and role-switching behaviour in swarm robotics: a case study. *Procedings of the European Conference on Artificial Life (ECAL 2013)*, pages 379–386, 2013.