4

Algorithm Selection: From Meta-Learning to Hyper-Heuristics

Laura Cruz-Reyes[1], Claudia Gómez-Santillán[1],
Joaquín Pérez-Ortega[2], Vanesa Landero[3],
Marcela Quiroz[1] and Alberto Ochoa[4]
[1]Instituto Tecnológico de Cd. Madero
[2]Centro Nacional de Investigación y Desarrollo Tecnológico
[3]Universidad Politécnica de Nuevo León
[4]Universidad de Ciudad Juárez
México

1. Introduction

In order for a company to be competitive, an indispensable requirement is the efficient management of its resources. As a result derives a lot of complex optimization problems that need to be solved with high-performance computing tools. In addition, due to the complexity of these problems, it is considered that the most promising approach is the solution with approximate algorithms; highlighting the heuristic optimizers. Within this category are the basic heuristics that are experience-based techniques and the metaheuristic algorithms that are inspired by natural or artificial optimization processes.

A variety of approximate algorithms, which had shown satisfactory performance in optimization problems, had been proposed in the literature. However, there is not an algorithm that performs better for all possible situations, given the amount of available strategies, is necessary to select the one who adapts better to the problem. An important point is to know which strategy is the best for the problem and why it is better.

The chapter begins with the formal definition of the Algorithm Selection Problem (ASP), since its initial formulation. The following section describes examples of "Intelligent Systems" that use a strategy of algorithm selection. After that, we present a review of the literature related to the ASP solution. Section four presents the proposals of our research group for the ASP solution; they are based on machine learning, neural network and hyper-heuristics. Besides, the section presents experimental results in order to conclude about the advantages and disadvantages of each approach. Due to a fully automated solution to ASP is an undecidable problem, Section Five reviews other less rigid approach which combines intelligently different strategies: The Hybrid Systems of Metaheuristics.

2. The Algorithm Selection Problem (ASP)

Many optimization problems can be solved by multiple algorithms, with different performance for different problem characteristics. Although some algorithms are better than others on average, there is not a best algorithm for all the possible instances of a given problem. This phenomenon is most pronounced among algorithms for solving NP-Hard problems, because runtimes for these algorithms are often highly variable from instance to instance of a problem (Leyton-Brown et al., 2003). In fact, it has long been recognized that there is no single algorithm or system that will achieve the best performance in all cases (Wolpert & Macready, 1997). Instead we are likely to attain better results, on average, across many different classes of a problem, if we tailor the selection of an algorithm to the characteristics of the problem instance (Smith-Miles et al., 2009). To address this concern, in the last decades researches has developed technology to automatically choose an appropriate optimization algorithm to solve a given instance of a problem, in order to obtain the best performance.

Recent work has focused on creating algorithm portfolios, which contain a selection of state of the art algorithms. To solve a particular problem with this portfolio, a pre-processing step is run where the suitability of each algorithm in the portfolio for the problem at hand is assessed. This step often involves some kind of machine learning, as the actual performance of each algorithm on the given, unseen problem is unknown (Kotthoff et al., 2011).

The Algorithm Selection Problem (ASP) was first described by John R. Rice in 1976 (Rice, 1976) he defined this problem as: learning a mapping from feature space to algorithm performance space, and acknowledged the importance of selecting the right features to characterize the hardness of problem instances (Smith-Miles & Lopes, 2012). This definition includes tree important characteristics (Rice, 1976):

a. *Problem Space*: The set of all possible instance of the problem. There are a big number of independent characteristics that describe the different instances which are important for the algorithm selection and performance. Some of these characteristics and their influences on algorithm performance are usually unknown.
b. *Algorithm Space*: The set of all possible algorithms that can be used to solve the problem. The dimension of this set could be unimaginable, and the influence of the algorithm characteristics is uncertain.
c. *Performance Measure*: The criteria used to measure the performance of a particular algorithm for a particular problem and see how difficult to solve (hard) is the instance. There is considerable uncertainly in the use and interpretation of these measures (e. g. some prefer fast execution, others effectiveness, others simplicity).

Rice proposed a basic model for this problem, which seeks to predict which algorithm from a subset of the algorithm space is likely to perform best based on measurable features of a collection of the problem space: Given a problem subset of the problem space P, a subset of the algorithm space A, a mapping from P to A and the performance space Y. The Algorithm Selection Problem can be formally defined as: for a particular problem instance $p \in P$, find the selection mapping $S(p)$ into the algorithm space A, such that the selected algorithm $a \in A$ maximizes the performance measure $||y||$ for $y(a,p) \in Y$. This basic abstract model is illustrated in Figure 1 (Rice, 1976; Smith-Miles & Lopes, 2012).

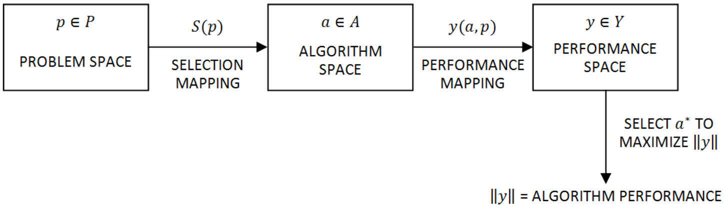


Fig. 1. The Algorithm Selection Problem (ASP)

The Figure 2 shows the dimensions of ASP and allows see a higher level of abstraction scope. There are three dimensions: 1) in the *x*-axis expresses a set of algorithms of solution {*s*, *t*, *w*, *y*, *z*}, 2) *z*-axis shows a set of instances of the problem {*a*, *b*, *c*, *d*}, and a new instance *e* to solve, 3) in the *y*-axis the set of values of the results of applying the algorithms to each of the instances is represented by vertical lines. As shown in figure, to solve the instance *a* and *b* the algorithms have different performances, it is noteworthy that no algorithm is superior to others in solving all instances. Moreover, as shown in figure the algorithm *s* has a different performance by solving each of the instances. Finally the problem to be solved is to select for the new instance *e* the algorithm that will solve better.

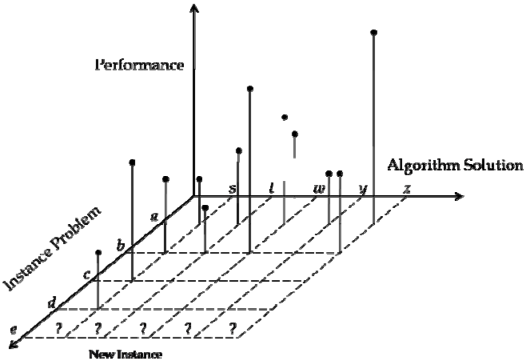


Fig. 2. Dimensions of algorithm selection problem

As we can see in the definition of the Algorithm Selection Problem there are three principal aspects that must be tackled in order to solve the problem:

a. The *selection of the set features of the problem* that might be indicative of the performance of the algorithms.
b. The *selection of the set of algorithms* that together allow to solve the largest number of instances of the problem with the highest performance.
c. The *selection of an efficient mapping mechanism* that permits to select the best algorithm to maximize the performance measure.

Some studies have been focused in construct a suitable set of features that adequately measure the relative difficulty of the instances of the problem (Smith-Miles et al., 2009; Messelis et al., 2009; Madani et al., 2009; Quiroz, 2009; Smith-Miles & Lopes, 2012). Generally there are two main approaches used to characterize the instances: the first is to identify problem dependent features based on domain knowledge of what makes a particular instance challenging or easy to solve; the second is a more general set of features derived from landscape analysis (Schiavinotto & Stützle, 2007; Czogalla & Fink, 2009). To define the set of features that describe the characteristics of the instances is a difficult task that requires expert domain knowledge of the problem. The indices of characterization should be carefully chosen, so as to permit a correct discrimination of the difficulty of the instances to explain the algorithms performance. There is little that will be learned via a knowledge discovery process if the features selected to characterize the instances do not have any differentiation power (Smith-Miles et al., 2009).

On the other hand, portfolio creation and algorithm selection has received a lot of attention in areas that deal with solving computationally hard problems (Leyton-Brown et al., 2003; O'Mahony et al., 2008). The current state of the art is such that often there are many algorithms and systems for solving the same kind of problem; each with its own performance on a particular problem. Machine learning is an established method of addressing ASP (Lobjois & Lemâitre, 1998; Fink, 1998). Given the performance of each algorithm on a set of training problems, we try to predict the performance on unseen problems (Kotthoff et al., 2011). There have been many studies in the area of algorithm performance prediction, which is strongly related to algorithm selection in the sense that supervised learning or regression models are used to predict the performance ranking of a set of algorithms, given a set of features of the instances (Smith-Miles & Lopes, 2012).

In the selection of the efficient mapping mechanism a challenging research goal is to design a run-time system that can repeatedly execute a program, learning over time to make decisions that maximize the performance measure. Since the right decisions may depend on the problem size and parameters, the machine characteristics and load, the data distribution, and other uncertain factors, this can be quite challenging. Some works treats algorithms in a black-box manner: each time a single algorithm is selected and applied to the given instance then a regression analysis or machine learning techniques are used to build a predictive model of the performance of the algorithms given the features of the instances (Lobjois & Lemâitre, 1998; Fink, 1998; Leyton-Brown et al., 2003; Ali & Smith, 2006). Other works focus on dynamic selection of algorithm components while the instance is being solved. In that sense, each instance is solved by a mixture of algorithms formed dynamically at run-time (Lagoudakis & Littman, 2000; Samulowitz & Memisevic, 2007; Streeter et al., 2007). The use of efficient mapping mechanism in intelligent systems is described in the next section.

3. Applications of algorithm selection to real world and theorists problems

The principles applied to ASP can be used in a wide range of applications in the real world and theoretical. Generally an application that solves a real problem is an extended version of parameters and constraints in another application that solves a theoretical problem. The nature of the algorithm selection problem is dynamic because it must incorporate new knowledge periodically, in order to preserve the efficacy of selection strategies. This section describes some applications to real-world complex problems, such as knowledge discovery and data mining, bioinformatics and Web services. It also describes some applications to solve complex theoretical problems; some examples are NP-hard problems, also called combinatorial optimization problems.

3.1 Bioinformatics

In (Nascimento et al., 2009) the authors investigate the performance of clustering algorithms on gene expression data, by extracting rules that relate the characteristics of the data sets of gene expression to the performance achieved by the algorithms. This represents a first attempt to solve the problem of choosing the best cluster algorithm with independence of gene expression data. In general, the choice of algorithms is basically driven by the familiarity of biological experts to the algorithm, rather than the characteristics of the algorithms themselves and of the data. In particular, the bioinformatics community has not reached consensus on which method should be preferably used. This work is directly derived from the Meta-Learning framework, originally proposed to support algorithm selection for classification and regression problems. However, Meta-Learning has been extended to other domains of application, e.g. to select algorithms for time series forecasting, to support the design of planning systems, to analyze the performance of meta-heuristics for optimization problems. Meta-Learning can be defined by considering four aspects: (a) the problem space, P, (b) the meta-feature space, F, (c) the algorithm space, A and (d) a performance metric, Y. As final remark, authors demonstrated that the rule-based ensemble classifier presented the most accuracy rates in predicting the best clustering algorithms for gene expression data sets. Besides, the set of extracted rules for the selection of clustering algorithms, using an inductive decision tree algorithm, gave some interesting guidelines for choosing the right method.

3.2 WEB services

In recent years, many studies have focused on developing feasible mechanisms to select appropriate services from service systems in order to improve performance and efficiency. However, these traditional methods do not provide effective guidance to users and, with regard to ubiquitous computing, the services need to be context-aware. In consequence, the work achieved by (Cai et al., 2009) proposed a novel service selection algorithm based on Artificial Neural Network (ANN) for ubiquitous computing environment. This method can exactly choose a most appropriate service from many service providers, due to the earlier information of the cooperation between the devices. Among the elements that exist in the definition of a service, Z represents the evaluation value of respective service providers' service quality, and its value is calculated with a function that involves the time and the conditions of current context environment, e.g. user context, computing context, physical context, with a division into static and dynamical information.

Among the advantages of using ANN to solve the service selection problem, is that, the method can easily adapt the evaluation process to the varying context information, and hence, it can provide effective guidance so that lots of invalid selecting processes can be avoided. The neural network selected was Back Propagation (BP) because is the most commonly used; however, this algorithm was improved with a three-term approach: learning rate, momentum factor and proportional factor. The efficiency of such algorithm was obtained because adding the proportional factor enhanced the convergence speed and stability. In conclusion, the authors claim, that this novel service selection outperforms the traditional service selection scheme.

3.3 Learning systems

In (Bradzil et al., 2003) is described a meta-learning method to support selection of candidate learning algorithms. Bradzil et al. use the Instance-Based Learning (IBL) approach because IBL has the advantage that the system is extensible; once a new experimental result becomes available, it can be easily integrated into the existing results without the need to reinitiate complex re-learning. In this work a k-Nearest Neighbor (k-NN) algorithm to identify the datasets that are most similar to the one is used. The distance between datasets is assessed using a relatively small set of data characteristics, which was selected to represent properties that affect algorithm performance; it is used to generate a recommendation to the user in the form of a ranking. The prediction, is constructed by aggregating performance information for the given candidate algorithms on the selected datasets. They use a ranking method based on the relative performance between pairs of algorithms. This work shown how can be exploited meta-learning to pre-select and recommend one or more classification algorithms to the user. They claimed that choosing adequate methods in a multistrategy learning system might significantly improve its overall performance. Also it was shown that meta-learning with k-NN improves the quality of rankings methods in general.

3.4 Knowledge discovery and data mining

In (Hilario & Kaousis, 2000) is addressed the model selection problem in knowledge discovery systems, defined as the problem of selecting the most appropriate learning model or algorithm for a given application task. In this work they propose framework for characterizing learning algorithms for classification as well as their underlying models, using learning algorithm profiles. These profiles consist of metalevel feature-value vectors, which describe learning algorithms from the point of view of their representation and functionality, efficiency, resilience, and practicality. Values for these features are assigned on the basis of author specifications, expert consensus or previous empirical studies. Authors review past evaluations of the better known learning algorithms and suggest an experimental strategy for building algorithm profiles on more quantitative grounds. The scope of this paper is limited to learning algorithms for classification tasks, but it can be applied to learning models for other tasks such as regression or association.

In (Kalousis & Theoharis, 1999) is presented an Intelligent Assistant called NOEMON, which by inducing helpful suggestion from background information can reduce the effort in classifier selection task. For each registered classifier, NOEMON measures its performance in order to collect datasets for constituting a morphologic space. For suggest the most appropriate classifier, NOEMON decides on the basis of morphological similarity between the new dataset and the existing collection. Rules are induced from those measurements and accommodated in a knowledge database. Finally, the suggestions on the most appropriate classifier for a dataset are based on those rules. The purpose of NOEMON is to supply the expert with suggestions based on its knowledge on the performance of the models and algorithms for related problems. This knowledge is being accumulated in a knowledge base end is updated as new problems as are being processed.

3.5 Scheduling problem

In (Kadioglu et al., 2011) the main idea is taken from an algorithm selector called Boolean Satisfiability (SAT) based on nearest neighbor classifier. On one hand, authors presented two extensions to it; one of them is based on the concept of distance-based weighting, where they assign larger weights to instances that are closer to the test instance. The second extension, is based on clustering-based adaptive neighborhood size, where authors adapt the size of the neighborhood based on the properties of the given test instance. These two extensions show moderate but consistent performance improvements to the algorithm selection using Nearest-Neighbor Classification (Malitsky et al., 2011). On the other hand, authors developed a new hybrid portfolio that combines algorithm selection and algorithm scheduling, in static and dynamic ways. For static schedules the problem can be formulated as an integer program, more precisely, as a resource constrained set covering problem, where the goal is to select a number of solver-runtime pairs that together "cover" (i.e., solve) as many training instances as possible. Regarding dynamic schedules, the column generation approach works fast enough when yielding potentially sub-optimal but usually high quality solutions. This allows us to embed the idea of dynamic schedules in the previously developed nearest-neighbor approach, which selects optimal neighborhood sizes by random sub-sampling validation. With SAT as the testbed, experimentation demonstrated that author's approach can handle highly diverse benchmarks, in particular a mix of random, crafted, and industrial SAT instances, even when deliberately removed entire families of instances from the training set. As a conclusion, authors presented a heuristic method for computing solver schedules efficiently, which O'Mahony (O'Mahony et al., 2008) identified as an open problem. In addition, they showed that a completely new way of solver scheduling consisting of a combination of static schedules and solver selection is able to achieve significantly better results than plain algorithm selection.

3.6 Traveling salesman problem

In (Kanda et al., 2011), the work is focused in the selection of optimization algorithms for solving TSP instances; this paper proposes a meta-learning approach to recommend optimization algorithms for new TSP instances. Each instance is described by meta-features of the TSP that influences the efficiency of the optimization algorithms. When more than one algorithm reaches the best solution, the multi-label classification problem is addressed applying three steps: 1) decomposition of multi-label instances into several single-label instances, 2) elimination of multi-label instances, and 3) binary representation, in order to transform multi-label instances into several binary classification problems. Features were represented as a graph. The success of this meta-learning approach depended on the correct identification of the meta-features that best relate the main aspects of a problem to the performances of the used algorithms. Finally the authors claimed that it is necessary to define and expand the set of metafeatures, which are important to characterize datasets in order to improve the performance of the selection models.

3.7 Satisfiability problem

In (Xu et al., 2009) is described an algorithm for constructing per-instance algorithm portfolios for SAT. It has been widely observed that there is no single "dominant" SAT solver; instead, different solvers perform best on different instances. SATzilla is an automated approach for constructing per-instance algorithm portfolios for SAT that use so-called empirical hardness models to choose among their constituent solvers. This approach takes as input a distribution of problem instances and a set of component solvers, and constructs a portfolio optimizing a given objective function (such as mean runtime, percent of instances solved, or score in a competition). The algorithm selection approach is based on the idea of building an approximate runtime predictor, which can be seen as a heuristic approximation to a perfect oracle. Specifically, they use machine learning techniques to build an empirical hardness model, a computationally inexpensive predictor of an algorithm's runtime on a given problem instance based on features of the instance and the algorithm's past performance. By modeling several algorithms and, at runtime, choosing the algorithm predicted to have the best performance; empirical hardness models can serve as the basis for an algorithm portfolio that solves the algorithm selection problem automatically.

3.8 Vehicle routing problem

In (Ruiz-Vanoye et al., 2008) the main contribution of this paper is to propose statistical complexity indicators applied to the Vehicle Routing Problem with Time Windows (VRPTW) instances in order that it allows to select appropriately the algorithm that better solves a VRPTW instance. In order to verify the proposed indicators, they used the discriminant analysis contained in SPSS software, such as a machine learning method to find the relation between the characteristics of the problem and the performance of algorithms (Perez et al., 2004), as well as the execution of 3 variants of the genetic algorithms and the random search algorithm. The results obtained in this work showed a good percentage of prediction taking into account that this based on statistical techniques and not on data-mining techniques. By means of the experimentation, authors conclude that it is possible to create indicators applied to VRPTW that help appropriately to predict the algorithm that better solves the instances of the VRPTW.

4. Related work on automatic algorithm selection

In this section some examples of related works of the reviewed literature are classified by Methods or methodologies utilized for establishing the relation between the problems and algorithms, and solve the algorithm selection problem. 2.1. Solution Environments, where the algorithm selection problem is boarded, are described in section 2.2.

4.1 Simple statistical tests

The most common method to compare experimentally algorithms consists in the complementary use of a set of simple well-known statistical tests: The Sign, Wilcoxon and Friedman tests, among others. The tests are based on the determination of the differences in the average performance, which is observed experimentally: if the differences among the algorithms are significant statistically, the algorithm with the best results is considered as superior (Lawler 1985). Reeves comments that a heuristic with good averaged performance, but with high dispersion, has a very high risk to show a poor or low performance in many instances (Reeves 1993). He suggests as alternative to formulate for each algorithm, a utility function adjusted to a gamma distribution, whose parameters permit to compare the heuristics on a range of risk value.

4.2 Regression analysis

Gent and Walsh make an empirical study of the GSAT algorithm, it is an approximation algorithm for SAT, and they apply regression analysis to model the growth of the cost of obtaining the solution with the problem size (Gent 1997).

In (Cruz 1999), Pérez and Cruz present a statistical method to build algorithm performance models, using polynomial functions, which relate the performance with the problem size. This method first generates a representative sample of the algorithms performance, and then the performance functions are determined by regression analysis, which finally are incorporated in an algorithm selection mechanism. The polynomial functions are used to predict the best algorithm that satisfies the user requirements.

The performance of local search algorithms Novelty and SAPS for solving instances of the SAT problem were analyzed by (Hutter 2006). The authors used linear regression with linear and quadratic basis functions to build prediction models. Firstly, they built a prediction model, using problem features and algorithm performance, to predict the algorithm run time. Secondly, they build another prediction model, using problem features, algorithm parameter settings and algorithm performance. This model is used to automatically adjust the algorithm's parameters on a per instance basis in order to optimize its performance.

4.3 Functions of probability distribution

Frost finds that the performance of the algorithms to solve CSP instances can be approximated by two standard families of functions of continuous probability distribution (Frost 1997). The resoluble instances can be modeled by the Weibull distribution and the instances that are not resoluble by the lognormal distribution. He utilizes four parameters to generate instances: number of constraints, number of prohibited value pairs per constraint, the probability of a constraint existing between any pair of variables, the probability each constraint is statistically independent of the others, and the probability that a value in the domain of one variable in a constraint will be incompatible with a value in the domain of the other variable in the constraint.

Hoos and Stuzle present a similar work to Frost. They find that the performance of algorithms that solve the SAT instances can be characterized by an exponential distribution (Hoos 2000). The execution time distribution is determined by the execution of *k* times of an algorithm over a set of instances of the same family, using a high time as stop criteria and storing for each successful run the execution time required to find the solution. The empirical distribution of the execution time is the accumulated distribution associated with these observations, and it allows projecting the execution time *t* (given by the user) to the probability of finding a solution in this time. A family is a set of instances with the same value of the parameters that are considered critical for the performance.

An algorithm portfolio architecture was proposed in (Silverthorn 2010). This architecture employs three core components: a portfolio of algorithms; a generative model, which is fit to data on those algorithms past performance, then used to predict their future performance; and a policy for action selection, which repeatedly chooses algorithms based on those predictions. Portfolio operation begins with offline training, in which a) training tasks are drawn from the task distribution, b) each solver is run many times on each training task, and c) a model is fit to the outcomes observed in training. In the test phase that follows, repeatedly, (1) a test task is drawn from the same task distribution, (2) the model predicts the likely outcomes of each solver, (3) the portfolio selects and runs a solver for some duration, (4) the run's outcome conditions later predictions, and (5) the process continues from (2) until a time limit expires.

The models of solver behavior are two latent class models: a multinomial mixture that captures the basic correlations between solvers, runs, and problem instances, and a mixture of Dirichlet compound multinomial distributions that also captures the tendency of solver outcomes to recur. Each model was embedded in a portfolio of diverse SAT solvers and evaluated on competition benchmarks. Both models support effective problem solving, and the DCM-based portfolio is competitive with the most prominent modern portfolio method for SAT (Xu 2009).

4.4 Functions of heuristic rules

Rice introduced the poly-algorithm concept (Rice 1968) in the context of parallel numeric software. He proposes the use of functions that can select, from a set of algorithms, the best to solve a given situation. After the Rice work, other researchers have formulated different functions that are presented in (Li 1997, Brewer 1995). The majority of the proposed functions are simple heuristic rules about structural features of the parameters of the instance that is being solved, or about the computational environment. The definition of the rules requires of the human experience.

The objective of the proposed methodology in (Beck 2004) is to find the best solution to a new instance, when a total limit time T is given. Firstly, the selection strategies for a set of algorithms A were formulated and denominated as prediction rules, these are: Selection is based on the cost of the best solution found by each algorithm; Selection is based on the change in the cost of the best solutions found at 10 second intervals; Selection is based on the extrapolation of the current cost and slope to a predicted cost at T.

These rules are applied for the training dataset and the optimal sampling time t* (required time to select the algorithm with the less solution error) is identified for each of them. After, when a new instance is given, each prediction rule is utilized to find the algorithm with the best found solution in a time tp = |A| x t*, and it is executed in the remaining time *tr* = T - tp. One of the advantages is that the methodology can be applied to different problems and algorithms. Nevertheless, the new dataset must have similarity with the training dataset.

4.5 Machine learning

The algorithm selection problem is focused by Lagoudakis and Littam in (Lagoudakis 2000) as a minimization problem of execution total time, which is solved with a Reinforced Learning algorithm (RL). Two classical problems were focused: selecting and ordering. A function that predicts the best algorithm for a new instance using its problem size is determined by means of training. The learned function permits to combine several recursive algorithms to improve its performance: the actual problem is divided in subproblems in each recursive step, and the most adequate algorithm in size is used for each of them. This work is extended to backtracking algorithms to SAT problem in (Lagoudakis 2001).

A system (PHYTHIA-II) to select the most appropriated software to solve a scientific problem is proposed in (Houstis 2002). The user introduces the problem features (operators of the equation, its domain, values of the variables, etc.) and time requirements and allowed error. The principal components of PHYTHIA-II are the statistical analysis, pattern extraction module and inference engine. The first consists in ranking the algorithms performance data by means of Friedman rank sums (Hollander 1973). The second utilizes different machine learning methods to extract performance patterns and represent them with decision and logic rules. The third is the process to correspond the features of a new problem with the produced rules; the objective is to predict the best algorithm and the most appropriated parameters to solve the problem.

The METAL research group proposed a method to select the most appropriate classification algorithm for a set of similar instances (Soares 2003). They used a K-nearest neighborhood algorithm to identify the group of instances from a historical registry that exhibit similar features to those of a new instance group. The algorithm performance on instances of historical registry is known and is used to predict the best algorithms for the new instance group. The similarity among instances groups is obtained considering three types of problem features: general, statistical and derived from information theory.

A Bayesian approach is proposed in (Guo, 2004) to construct an algorithm selection system which is applied to the Sorting and Most Probable Explanation (MPE) problems. From a set of training instances, their features and the run time of the best algorithm that solves each instance are utilized to build the Bayesian network. Guo proposed four representative indexes from the Sorting problem features: the size of the input permutation and three presortedness measures. For the MPE problem he utilizes general features of the problem and several statistical indexes of the Bayesian network that represents the problem.

A methodology for instance based selection of solver's policies that solves instances of the SAT problem was proposed by (Nikolic 2009). The policies are heuristics that guide the search process. Different configurations of these policies are solution strategies. The problem structure of all instances was characterized by indices. The problem instances were grouped by the values of these indices, forming instances families. All problem instances were solved by all solution strategies. The best solution strategy for each family is selected. The k-nearest neighbor algorithm selects the solution strategy for a new input instance. The results of the performance of the algorithm ARGOSmart, that performs the proposed methodology, were superior to ARGOSAT algorithm.

5. Approaches to building algorithm selectors

In this chapter we solve ASP with two approaches: meta-learning and hyper-heuristics. The meta-learning approach is oriented to learning about classification using machine learning methods; three methods are explored to solve an optimization problem: Discriminant Analysis (Pérez, 2004), C4.5 and the Self-Organising Neural Network. The hyper-heuristic approach is oriented to automatically produce an adequate combination of available low-level heuristics in order to effectively solve a given instance (Burke et al., 2010); a hyper-heuristic strategy is incorporated in an ant colony algorithm to select the heuristic that best adjust one of its control parameter.

5.1 Selection of metaheuristics using meta-learning

In this section a methodology based on Meta-Learning is presented for characterizing algorithm performance from past experience data. The characterization is used to select the best algorithm for a new instance of a given problem. The phases of the methodology are described and exemplified with the well known one-dimensional Bin-Packing problem.

5.1.1 Algorithms for the solution of the Bin Packing Problem

The Bin Packing Problem (BPP) is an NP-hard combinatorial optimization problem, in which the objective is to determine the smallest number of bins to pack a set of objects. For obtaining suboptimal solutions of BPP, with less computational effort, we used deterministic and non-deterministic algorithms. The algorithm performance is evaluated with the optimal deviation percentage and the processing time (Quiroz, 2009).

The deterministic algorithms always follow the same path to arrive at the same solution. The First Fit Decreasing (FFD) algorithm places the items in the first bin that can hold them. The Best Fit Decreasing (BFD) places the items in the best-filled bin that can hold them. The Match to First Fit (MFF) algorithm is a variation of FFD, wich uses complementary bins for holding temporarily items. The Match to Best Fit (MBF) algorithm is a variation of BFD and, like MFF uses complementary bins. The Modified Best Fit Decreasing (MBFD) partially pack the bins in order to find a "good fit" item combination.

The Non-Deterministic Algorithms do not obtain the same solution in different executions, but in many cases they are faster than deterministic algorithms. The Ant Colony Optimization (ACO) algorithm builds a solution with each ant: it starts with an empty bin; next, each new bin is filled with "selected items" until no remaining item fits in it; finally, a "selected item" is chosen stochastically using mainly a pheromone trail (Ducatelle, 2001). In the Threshold Accepting (TA) algorithm, a new feasible solution is accepted if the difference with the previous solution is within a threshold temperature; the value of the temperature is decreased each time until a thermal equilibrium is reached (Pérez, 2002).

5.1.2 Methodology

The methodology proposed for performance characterization and its application to algorithm selection consists of three consecutive phases: Initial Training, Prediction and Training with Feedback. Figure 3 depicts these phases.

In the *Initial Training Phase*, two internal processes build a past experience database: the Problem Characterization Process obtains statistical indices to measure the computational complexity of a problem instance and, the Algorithm characterization Process solves instances with the available algorithms to obtain performance indices. The Training Process finally builds a knowledge base using the Problem and Algorithms Database. This knowledge is represented through a learning model, which relates the algorithms performance and the problem characteristics. In the *Prediction Phase*, The relationship learned is used to predict the best algorithm for a new given instance. In the *Training with*

Feedback phase, the new solved instances are incorporated into the characterization process for increasing the selection quality. The relationship learned in the knowledge base is improved with a new set of solved instances and is used again in the prediction phase.

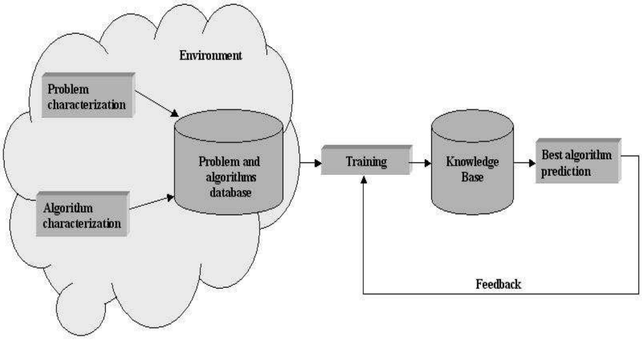

Environment
Problem characterization
Algorithm characterization
Problem and algorithms database
Training
Knowledge Base
Best algorithm prediction
Feedback

Fig. 3. Phases of the algorithm selection methodology

Initial training phase

The steps of this phase are shown in Figure 4 In step 1 (Characteristics Modeling) indices are derived for measuring the influence of problem characteristics on algorithm performance (see Equations 1 to 5). In step 2 (Statistical Sampling) a set of representative instances are generated with stratified sampling and a sample size derived from survey sampling. In step 3 (Characteristics Measurement) the parameter values of each instance are transformed into indices. In step 4 (Instances Solution) instances are solved using a set of heuristic algorithms. In Step 5 (Clustering) groups are integrated in such a way that they are constituted by instances with similar characteristics, and for which an algorithm outperformed the others. Finally, in step 6 (Classification) the identified grouping is learned into formal classifiers.

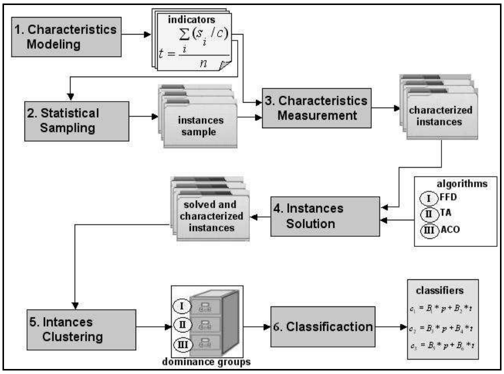

1. Characteristics Modeling
indicators
$t = \frac{\sum_i (s_i / c)}{n}$
2. Statistical Sampling
instances sample
3. Characteristics Measurement
characterized instances
algorithms
I FFD
II TA
III ACO
4. Instances Solution
solved and characterized instances
5. Intances Clustering
I
II
III
dominance groups
6. Classificaction
classifiers
$c_1 = B_1 * p + B_2 * t$
$c_2 = B_3 * p + B_4 * t$
$c_3 = B_5 * p + B_6 * t$

Fig. 4. Steps of the initial training phase

We propose five indices to characterize the instances of BPP:

Instance size p expresses a relationship between instance size and the maximum size solved, where, *n* is the number of items, *maxn* is the maximum size solved

$$p = \frac{n}{maxn} \tag{1}$$

a. *Constrained capacity t* expresses a relationship between the average item size and the bin size. The size of item *i* is s_i and the bin size is *c*.

$$t = \frac{\sum_i (s_i / c)}{n} \qquad 1 \le i \le n \tag{2}$$

b. *Item dispersion d* expresses the dispersion degree of the item size values.

$$d = \sigma\ (t) \tag{3}$$

c. *Number of factors f* expresses the proportion of items whose sizes are factors of the bin capacity.

$$f = \frac{\sum_i factor(c, s_i)}{n} \qquad 1 \le i \le n \tag{4}$$

d. *Bin usage b* expresses the proportion of the total size that can fit in a bin of capacity *c*.

$$b = \begin{cases} 1 & \text{if} \quad c \ge \sum_i s_i \\ \dfrac{c}{\sum_i s_i} & \text{otherwise} \end{cases} \qquad 1 \le i \le n \tag{5}$$

Prediction phase

The steps of this phase are shown in Figure 5. For a new instance, step 7 (Characteristics Measurement) calculates its characteristic values using indices. Step 8 uses the learned classifiers to determine, from the characteristics of the new instance, which group it belongs to. The algorithm associated to this group is the expected best algorithm for the instance.

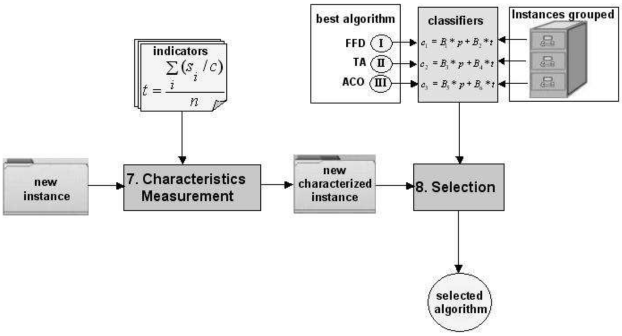

best algorithm
FFD I
TA II
ACO III
classifiers
$c_1 = B_1 * p + B_2 * t$
$c_2 = B_3 * p + B_4 * t$
$c_3 = B_5 * p + B_6 * t$
Instances grouped
indicators
$t = \frac{\sum_i (s_i / c)}{n}$
new instance
7. Characteristics Measurement
new characterized instance
8. Selection
selected algorithm

Fig. 5. Steps of the prediction phase

Training and FeedBack phase

The steps of this phase are shown in Figure 6. The objective is to feedback the system in order to maintain it in a continuous training. For each new solved and characterized instance, step 9 (Instance Solution) obtains the real best algorithm. Afterwards, step 10 (Patterns Verification) compares the result, if the prediction is wrong and the average accuracy is beyond an specified threshold, then the classifiers are rebuilt using the old and new dataset; otherwise the new instance is stored and the process ends.

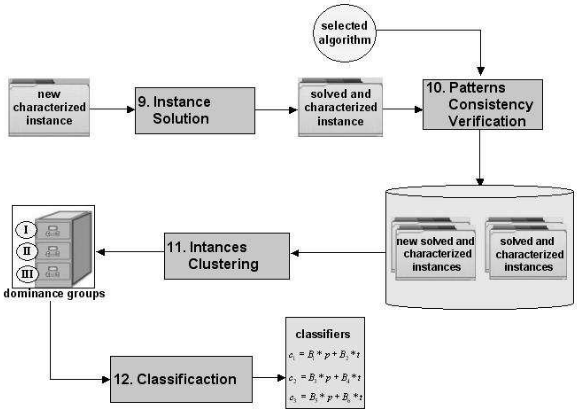

selected algorithm
new characterized instance
9. Instance Solution
solved and characterized instance
10. Patterns Consistency Verification
new solved and characterized instances
solved and characterized instances
11. Intances Clustering
I
II
III
dominance groups
12. Classificaction
classifiers
$c_1 = B_1 * p + B_2 * t$
$c_2 = B_3 * p + B_4 * t$
$c_3 = B_5 * p + B_6 * t$

Fig. 6. Steps of the training with feedback phase

5.1.3 Experimentation

For test purposes 2,430 random instances of the Bin-Packing problem were generated, characterized and solved using the seven heuristic algorithms described in Section 5.1.1. Table 1 shows a small instance set, which were selected from the sample.

Instance	Problem characteristic indices					Real best algorithms
	p	*b*	*t*	*f*	*d*	
E1i10.txt	0.078	0.427	0.029	0.000	0.003	FFD,TA
E50i10.txt	0.556	0.003	0.679	0.048	0.199	BFD,ACO
E147i10.txt	0.900	0.002	0.530	0.000	0.033	TA

Table 1. Example of random intances with their characteristics and the best algorithms

The K-means clustering method was used to create similar instance groups. Four groups were obtained; each group was associated with a similar instances set and an algorithm with the best performance for it. Three algorithms had poor performance and were outperformed by the other four algorithms. The Discriminant Analysis (DA) and C4.5 classification methods were used to build the algorithm selector. We use the machine learning methods available in SPSS version 11.5 and Weka 3.4.2, respectively. Afterwards, for validating the system, 1,369 standard instances were collected [Ross 2002]. In the selection of the best algorithm for all standard instances, the experimental results showed an accuracy of 76% with DA and 81% with C4.5. This accuracy was compared with a random selection from the seven algorithms: 14.2%. For the instances of the remaining percentage (100-76%), the selected algorithms generate a solution close to the optimal.

The selection system with feedback was implemented using a neural network, particularly the Self-Organizing Map (SOM) of Kohonen available in Matlab 7.0. The best results were obtained with only two problem characteristic indices (*p*,*t*) in a multi-network. The accuracy increased from 78.8% in 100 epochs up to 100% in 20,000 epochs. These percentages correspond to the network with initial-training and training-with-feedback, respectively. The SOM was gradually feedback with all the available instances. Using all indices (*p*,*b*,*t*,*f*,*d*) the SOM only reached 76.6% even with feedback.

5.2 Selection of heuristics in a hyper-heuristic framework

A hyper-heuristic is an automated methodology for selecting heuristics to solve hard computational search problems (Burke et al., 2009; Burke et al., 2010; Duarte et al., 2007). Its methodology is form by a high-level algorithm that, given a particular problem instance and a number of low-level heuristics or metaheuristic, can select and apply an appropriate low-level heuristic or metaheuristic at each decision step. These procedures on their way to work raise the generality at which search strategy can operate. General scheme for design a hyper-heuristic is shown in Figure 7.

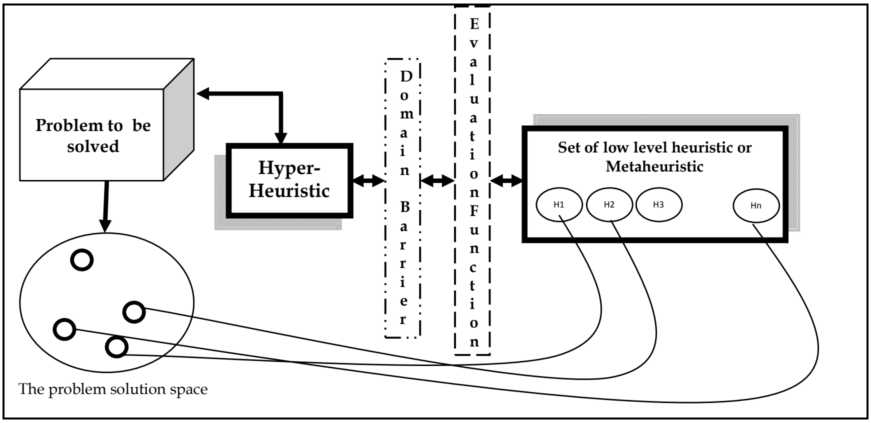


Fig. 7. Hyper-heuristic Elements

The first low-level algorithms build a solution incrementally; starting with an empty solution with the goal is to intelligently select the next construction heuristics or metaheuristic to gradually build a complete solution (Garrido, & Castro, 2009).

5.2.1 Representative examples

SQRP is the problem of locating information in a network based on a query formed by keywords. The goal of SQRP is to determine the shortest paths from a node that issues a query to nodes that can appropriately answer it (by providing the requested information). Each query traverses the network, moving from the initiating node to a neighboring node and then to a neighbor of a neighbor and so forth, until it locates the requested resource or gives up in its absence. Due to its complexity (Michlmayr, 2007) solutions proposed to SQRP typically limit to special cases.

Hyper-Heuristic_AdaNAS (HH_AdaNAS) is an adaptive metaheuristic algorithm, which resolves SQRP (Hernandez, 2010). This algorithm was created from AdaNAS (Gómez et al., 2010). The *high-level algorithm* is formed by HH_AdaNAS, which use as solution algorithm AdaNAS, that is inspired by an ant colony and the set of *low-level heuristics* are included in the algorithm called HH_TTL. The goal of hyperheuristic HH_TTL is to define by itself in real time, the most adequate values for time to live (TTL) parameter during the execution of the algorithm. The main difference between AdaNAS and HH_AdaNAS are: when applying the modification of the TTL and the calculation of the amount of TTL to be allocated. In the Figure 8 we show HH_AdaNAS is form by AdaNAS + HH_TTL.

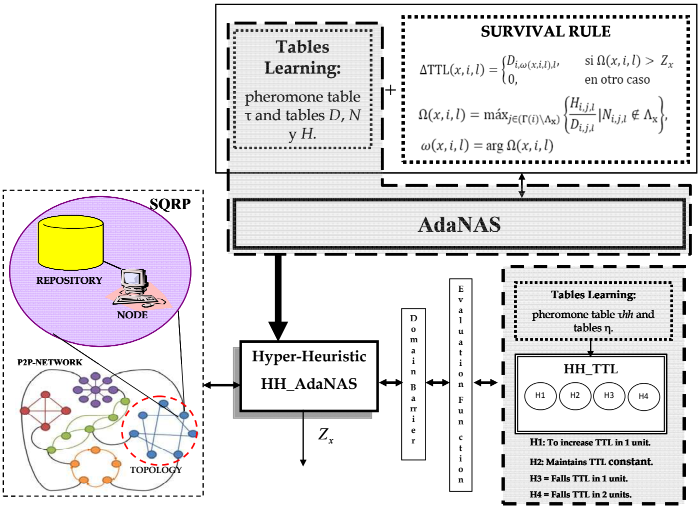

Tables Learning:
pheromone table τ and tables D, N y H.
+
SURVIVAL RULE
$\Delta\mathrm{TTL}(x,i,l) = \begin{cases} D_{i,\omega(x,i,l),l}, & \text{si } \Omega(x,i,l) > Z_x \\ 0, & \text{en otro caso} \end{cases}$
$\Omega(x,i,l) = \text{máx}_{j\in(\Gamma(i)\setminus\Lambda_x)} \left\{ \frac{H_{i,j,l}}{D_{i,j,l}} | N_{i,j,l} \notin \Lambda_x \right\},$
$\omega(x,i,l) = \arg \Omega(x,i,l)$
AdaNAS
SQRP
REPOSITORY
NODE
P2P-NETWORK
TOPOLOGY
Hyper-Heuristic
HH_AdaNAS
Z_x
Domain Barrier
Evaluation Function
Tables Learning:
pheromone table τ*hh* and tables η.
HH_TTL
H1
H2
H3
H4
H1: To increase TTL in 1 unit.
H2: Maintains TTL constant.
H3 = Falls TTL in 1 unit.
H4 = Falls TTL in 2 units.

Fig. 8. HH_AdaNAS is form by AdaNAS + HH_TTL.

Data structures of HH_AdaNAS

HH_AdaNAS inherited some data structures of AdaNAS, as the pheromone table τ and the tables *H*, *D* and *N*. Besides the data structures of the high level metaheuristics, are the structures that help to select the low-level heuristic these are the pheromone table τ*hh* and the table hiperheuristic visibility states η. All the tables stored heuristic information or gained experience in the past. The relationship of these structures is shown in Figure 9.

When HH_AdaNAS searches for the next node, in the routing process of the query, is based on the pheromone table τ and tables *D*, *N* y *H*; these tables are intended to give information on distant *D*, *H* is a table that records the successes of past queries and number of documents *N* which are the closest nodes that can satisfy the query. In the same way, when HH_TTL chooses the following low level heuristic, through data structures τ*hh* and η. The memory is composed of two data structures that store information of prior consultations. The first of these memories is the pheromone table τ*hh* which has three dimensions, and the other memory structure is the table hiper-heuristic visibility states η, which allows the hiperheuristic know in what state is SQRP. Is to say, if is necessary to add more TTL, because the amount of resources found are few and decreases the lifetime.

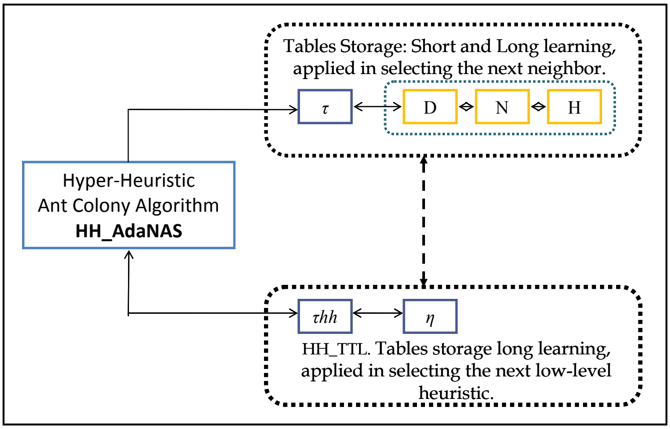

Tables Storage: Short and Long learning, applied in selecting the next neighbor.
τ
D
N
H
Hyper-Heuristic
Ant Colony Algorithm
HH_AdaNAS
τhh
η
HH_TTL. Tables storage long learning, applied in selecting the next low-level heuristic.

Fig. 9. Storage structures of HH_AdaNAS.

The pheromone table τ is divided into *n* two-dimensional tables, one corresponding to each node *i* of the network. These tables contain only entries for a node fixed *i*, therefore, its dimensions are at most $|L| \times |\Gamma(i)|$, where *L* is the dictionary, which defines the keywords allowed for consultation and Γ (*i*) is the set of neighboring nodes of *i*. Each in turn contains a two-dimensional table $|m| \times |h|$, where *m* is the states visibility set of the problem and *h* is the available heuristics set. The pheromone table is also called learning structure long.

The visibility state table η expresses the weight of the relation between SQRP-states and TTL-heuristics and was inspired by the deterministic survival rule designed by Rivera (Rivera G. 2009). Table η is formed by the combination of $|m| \times |h|$, where a visibility state m_i is identified mainly by α, which depends on the node selected by AdaNAS to route the query SQRP. The variable α in Equation 6 contributes to ensure that the node selected by HH_AdaNAS, in the future, not decreases the performance of the algorithm. A TTL-heuristic is intelligently selected according with the past performance given by its pheromone value, and its visibility value, given by an expert. The Figure 10 shows the visibility state table used in this work.

	h_1	h_2	h_3	h_4
m_1	1	0.75	0.5	0.25
m_2	0.75	1	0.5	0.5
m_3	0.5	0.5	1	0.75
m_4	0.25	0.5	0.75	1

Fig. 10. Visibility state table

$$\alpha=(H_{i,j,l} / D_{i,j,l}) / Z_x \qquad (6)$$

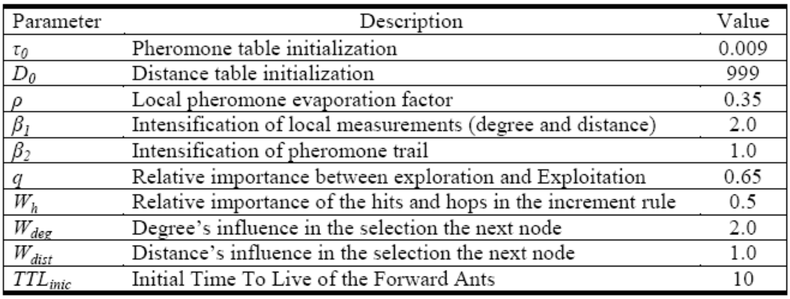
Where $H_{i,j,l}$ indicates the number of documents consistent with the query l, $D_{i,j,l}$ indicates the length of the route to obtain the documents, i represented the current node and j is the node chosen, and Z_x is a measure of current performance. In this work the visibility states are: $m_1 = (\alpha > 1)\&(TTL < D)\&(\ TTL\ != 1)$, $m_2 = (\alpha > 1)\&(TTL < D)\&(\ TTL = 1)$, $m_3 = (H = 0)\ ||((\ \alpha > 1)\&(TTL \geq D))\ ||\ ((\ \alpha \leq 1)\&(TTL = 1))$ and $m_4 = (\ \alpha \leq 1)\&(TTL > 1)$. All the visibility states are calculated to identify which heuristic will be applied to TTL.

5.2.2 Experimentation

The experimental environment used during experiments, and the results obtained are presented in this section. **Software:** Microsoft Windows 7 Home Premium; Java programming language, Java Platform, JDK 1.6; and integrated development, Eclipse 3.4. **Hardware:** Computer equipment with processor Intel (R) Core (TM) i5 CPU M430 2.27 GHz and RAM memory of 4 GB. **Instances:** It has 90 different SQRP instances; each of them consists of three files that represent the topology, queries and repositories. The description of the features can be found in (Cruz et al. 2008).

The average performance was studied by computing three performance measures of each 100 queries: **Average hops**, defined as the average amount of links traveled by a Forward Ant until its death that is, reaching either the maximum amount of results required or running out of TTL. **Average hits**, defined as the average number of resources found by each Forward Ant until its death, and **Average efficiency**, defined as the average of resources found per traversed edge (hits/hops). The initial Configuration of HH_AdaNAS is shown in Table 2. The parameter values were based on values suggested of the literature as (Dorigo & Stützle, 2004; Michlmayr, 2007; Aguirre, 2008 and Rivera, 2009).

In this section we show experimentally that HH_AdaNAS algorithm outperforms the AdaNAS algorithm. Also HH_AdaNAS outperforms NAS (Aguirre, 2008), SemAnt (Michlmayr, 2007) and random walk algorithms (Cruz et al., 2008), this was reported in (Gómez et al., 2010), so HH_AdaNAS algorithm is positioned as the best of them.



Parameter	Description	Value
τ_0	Pheromone table initialization	0.009
D_0	Distance table initialization	999
ρ	Local pheromone evaporation factor	0.35
β_1	Intensification of local measurements (degree and distance)	2.0
β_2	Intensification of pheromone trail	1.0
q	Relative importance between exploration and Exploitation	0.65
W_h	Relative importance of the hits and hops in the increment rule	0.5
W_{deg}	Degree's influence in the selection the next node	2.0
W_{dist}	Distance's influence in the selection the next node	1.0
TTL_{inic}	Initial Time To Live of the Forward Ants	10

Table 2. Shows the assignment of values for each HH_AdaNAS parameter.

In this experiment, we compare the HH_AdaNAS and AdaNAS algorithms. The performance achieved is measured by the rate of found documents and the experiments were conducted under equal conditions, so each algorithm was run 30 times per instance and used the same configuration parameters for the two algorithms, which is described in Table 2.

The Figure 11 shows the average efficiency performed during a set of queries with HH_AdaNAS and AdaNAS algorithms; for the two algorithms the behavior is approximately the same. The algorithm HH_AdaNAS at the beginning the efficiency is around 2.38 hits per hop in the first 100 queries and the algorithm AdaNAS start approximately at 2.37 hits per query also in the top 100 queries. Analyzing at another example of the experiment, after processing the 11 000 queries at the end the efficiency increases around 3.31 hits per hop for the algorithm HH_AdaNAS and the algorithm AdaNAS at 3.21 hits per query. Finally, due to the result we conclude that HH_AdaNAS achieves a final improvement in performance of 28.09%, while AdaNAS reaches an improvement of 26.16%.

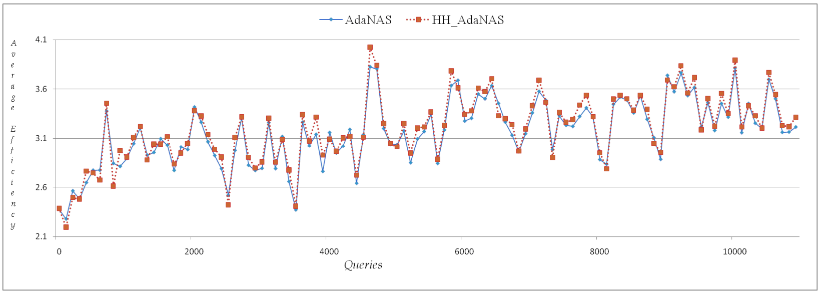


Fig. 11. The average efficiency performed during 11,000 queries with two algorithms.

6. Hybrid systems of metaheuristics: an approximate solution of ASP

The majority of problems related with ASP have a high level of complexity, according to application domains. An alternative solution is the use of Hybrid Systems based on Heuristics and Metaheuristics. Algorithm selection has attracted the attention of some research in hybrid intelligent systems, for which many algorithms and large datasets are available. Hybrid Intelligent Systems seek to take advantage of the synergy between various intelligent techniques in solving real problems (Ludermir et al., 2011).

6.1 Relation of meta-learning and hybridization

Although some algorithms based on Hybrid Systems of Metaheuristics are better than others on average, there is rarely a best algorithm for a given problem according to the complexity and application domain related with the proposal solution. Instead, it is often the case that different algorithms perform well on different problem instances. This condition is most accentuated among algorithms for solving NP-Hard problems, because runtimes of these algorithms are often highly variable from instance to instance.

When algorithms present high runtime variability, one is faced with the problem of deciding which algorithm to use. Rice called this the “algorithm selection problem” (Rice, 1976). The algorithm selection has not received widespread attention. The most common approach to algorithm selection has been to measure the performance of different algorithms on a given instances set with certain distribution, and then select the algorithm with the lowest average runtime.

This “winner-take-all” approach has produced recent and important advances in algorithm design and refinement, but has caused the rejection of many algorithms that has an excellent performance on an specific cases, but result uncompetitive on average. The following two questions emerge from the literature (Leyton-Brown, 2003). How to perform an algorithm selection for a given instance? How to evaluate novel hybrid algorithms?

a. Algorithms with high average running times can be combined to form a hybrid algorithm more robust and with low average running time when the algorithm inputs are sufficiently easy and uncorrelated.
b. New hybrid algorithm design should find more robust solution and focus on problems on which a single algorithm performs poorly.
c. A portfolio of algorithms can also be integrated through the use of hybrid algorithms because the solutions are considering more innovative.

In previous section we use machine learning algorithms to automatically acquire knowledge for algorithm selection, leading to a reduced need for experts and a potential improvement of performance. In general, the algorithm selection problem can be treated via meta-learning approaches. The results of this approach can cause an important impact on hybridization. In order to clarify this point, is important to speculate about how the empirical results of meta-learning can be analyzed from a theoretical perspective with different intentions:

a. Confirm the sense of the selection rules
b. Generate insights into algorithm behavior that can be used to refine the algorithms.

The acquired knowledge is confirmed when the performance of the refined algorithms is evaluated. The knowledge can be used to integrate complementary strategies in a hybrid algorithm.

6.2 Use of hybridization to solve ASP in social domains

The principal advanced in the reduction of Complexity is related with the amalgam of different perspectives established on different techniques which to demonstrate their efficiency in different application domains with good results.

Hybridization of Algorithms is one of the most adequate ways to try to improve and solve different ASP related with the optimization of time. Many applied ASP´s have an impact on social domains specially to solve dynamic and complex models related with human behavior. In (Araiza, 2011) is possible analyze with a Multiagents System the concept of “Social Isolation”, featuring this behavior on the time according with interchanges related with a minority and the associated health effects, when this occurs.

In addition, is possible specify the deep and impact of a viral marketing campaign using a Social Model related with Online Social Networking. In (Azpeitia, 2011), an adequate ASP determines the way on the future of this campaign and permits analyze the track of this to understand their best features.

6.3 Future trends on the resolution of ASP using a hybrid system of metaheuristics

We expected that the future trends for solving ASP with hybridization will be based on models that tend to perform activities according to a selection framework and a dynamic contextual area. The decision of the most appropriate actions requires advanced Artificial Intelligence Technique to satisfy a plethora of application domains in which interaction and conclusive results are needed. This only is possible with Intelligent Systems equipped with high processing speed, knowledge bases and an innovative model for designing experiments, something will happen in this decade.

7. Conclusions

Many real world problems belong to a special class of problems called NP-hard, which means that there are no known efficient algorithms to solve them exactly in the worst case. The specialized literature offers a variety of heuristic algorithms, which have shown satisfactory performance. However, despite the efforts of the scientific community in developing new strategies, to date, there is no an algorithm that is the best for all possible situations. The design of appropriate algorithms to specific conditions is often the best option. In consequence, several approaches have emerged to deal with the algorithm selection problem. We review hyper-heuristics and meta-learning; both related and promising approaches.

Meta-learning, through machine learning methods like clustering and classification, is a well-established approach of selecting algorithms, particularity to solve hard optimization problems. Despite this, comparisons and evaluations of machine learning methods to build algorithm selector is not a common practice. We compared three machine learning techniques for algorithm selection on standard data sets. The experimental results revealed in general, a high performance with respect to a random algorithm selector, but low perform with respect to other classification tasks. We identified that the Self-Organising Neural Network is a promising method for selection; it could reaches 100% of accuracy when feedback was incorporated and the number of problem characteristics was the minimum.

On the other hand, hyper-heuristics offers a general framework to design algorithms that ideally can select and generate heuristics adapted to a particular problem instance. We use this approach to automatically select, among basic-heuristics, the most promising to adjust a parameter control of an Ant Colony Optimization algorithm for routing messages. The adaptive parameter tuning with hyper-heuristics is a recent open research.

In order to get a bigger picture of the algorithm performance we need to know them in depth. However, most of the algorithmic performance studies have focused exclusively on identifying sets of instances of different degrees of difficulty; in reducing the time needed to resolve these cases and reduce the solution errors; in many cases following the strategy "the -winner takes-all". Although these are important goals, most approaches have been quite particular. In that sense, statistical methods and machine learning will be an important element to build performance models for understanding the relationship between the characteristics of optimization problems, the search space that defines the behavior of algorithms that solve, and the final performance achieved by these algorithms. We envision that the knowledge gained, in addition to supporting the growth of the area, will be useful to automate the selection of algorithms and refine algorithms; hiper-heuristics, hybridization, and meta-learning go in the same direction and can complement each other.

8. Acknowledgment

This research was supported in part by CONACYT and DGEST

9. References

Ali, S. & Smith, K. (2006). On learning algorithm selection for classification. *Applied Soft Computing*, Vol. 6, No. 2, (January 2006), pp. 119-38.

Aguirre, M. (2008). *Algoritmo de Búsqueda Semántica para Redes P2P Complejas*. Master's thesis, División de Estudio de Posgrado e Investigación del Instituto Tecnológico de Ciudad Madero, Tamaulipas, México.

Azpeitia, D. (2011). Critical Factors for Success of a Viral Marketing Campaign of Real-Estate Sector at Facebook: The strength of weak learnability. *Proceedings of the HIS Workshop at MICAI*

Beck, J. & Freuder, E. (2004). Simple Rules for Low-Knowledge Algorithm Selection. *Proceedings of the 1st International Conference on Integration of IA and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Nice, France, April 2004, J. Regin and M. Rueher (Ed.). Springer-Verlag Vol. 3011, pp. 50-64.

Brazdil, P. B., Soares C., & Pinto, D. C. J. (2003). Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results. *Machine Learning*, Vol. 50, No. 3, pp. 251–277, ISSN: 08856125

Brewer, E. (1995). High-Level Optimization Via Automated Statistical Modeling. *Proceedings of Principles and Practice of Parallel Programming*, Santa Barbara, CA, July 1995, ACM Press, New York, USA, pp. 80-91

Burke, E., Hyde, M., Kendall, G., Ochoa, G., Özcan, E. & Woodward, J. (2009). Exploring hyper-heuristic methodologies with genetic programming. In: *Computational Intelligence*: Collaboration, Fusion and Emergence, Intelligent Systems Reference Library

Burke, K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E. & Woodward, R. (2010). A Classification of Hyper-heuristic Approaches, In: *International Series in Operations Research & Management Science*, Gendreau, M. and Potvin, J.Y. pp.(449). Springer Science+Business Media, ISBN 978-1-4419-1663-1, NY, USA

Cai, H., Hu X., Lü Q., & Cao, Q. (2009). A novel intelligent service selection algorithm and application for ubiquitous web services environment. *Expert Systems with Applications*, Vol. 36, No. 2, Part 1, pp. 2200-2212, ISSN: 09574174

Cruz, L. (1999). *Automatización del Diseño de la Fragmentación Vertical y Ubicación en Bases de Datos Distribuidas usando Métodos Heurísticos y Exactos*. Master's thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey, México.

Cruz, L., Gómez, C., Aguirre, M., Schaeffer, S., Turrubiates, T., Ortega, R. & Fraire,H.(2008). NAS algorithm for semantic query routing systems in complex networks. In: *International Symposium on Distributed Computing and Artificial Intelligence 2008/ Advances in Soft Computing 2009*. Corchado J., Rodríguez S., Llinas J. & Molina J., pp. (284-292), Springer, Berlin /Heidelberg, ISBN 978-3-540-85862-1, DOI 10.1007/978-3-540-85863-8

Czogalla, J. & Fink, A. (2009). Fitness Landscape Analysis for the Resource Constrained Project Scheduling Problem. *Lecture Notes in Computer Science, Learning and Intelligent Optimization*, Vol. 5851, pp. 104-118

Dorigo, M. & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge, MA., ISBN 0-262-04219-3, EUA

Duarte, A., Pantrigo, J., Gallego, M. (2007). *Metaheurísticas,* Ed. Dykinson S.L. España

Ducatelle, F., & Levine, J. (2001). Ant Colony Optimisation for Bin Packing and Cutting Stock Problems. *Proceedings* of the UK Workshop on Computational Intelligence, Edinburgh

Fink, E. (1998). How to solve it automatically: Selection among Problem-Solving methods. *Proceedings of ICAPS 1998*, pp. 128–136

Frost, D.; Rish, I. & Vila, L. (1997). Summarizing CSP hardness with continuous probability distributions. *Proceedings of the 14th National Conference on AI*, American Association for Artificial Intelligence, pp. 327-333

Garrido, P. & Castro C. (2009). Stable solving of cvrps using hyperheuristics. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO'09)*, ACM, Montreal, Canada, July 2009

Gent, I.; Macintyre, E.; Prosser, P. & Walsh, T. (1997). The Scaling of Search Cost. In: *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence,* pp. 315-320, AAI Press, Retrieved from: https://www.aaai.org/Papers/AAAI/1997/AAAI97-049.pdf

Gómez, C.G., Cruz, L., Meza, E., Schaeffer, E. & Castilla, G.(2010). A Self-Adaptive Ant Colony System for Semantic Query Routing Problem in P2P Networks. *Computación y Sistemas* Vol. 13, No. 4, pp (433-448), ISSN 1405-5546

Guo, H. & Hsu, W. (2004). A Learning-based Algorithm Selection Meta-reasoner for the Real-time MPE Problem. *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, Cairns, Australian, Dec 2004, G. I. Webb and Xinghuo Yu (Ed.), Springer-Verlag Vol. 3339, pp. 307-318

Hernández P. (2010). *Método Adaptativo para el Ajuste de Parámetros de un Algoritmo Evolutivo Hiperheurístico*. Master's thesis, División de Estudio de Posgrado e Investigación del Instituto Tecnológico de Ciudad Madero, Tamaulipas, México

Hilario, M., & Kalousis, A. (2000). Building algorithm profiles for prior model selection in knowledge discovery systems. *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, Vol. 8, No. 2, 2000, pp. 77-88, ISSN: 09691170

Hollander, M. & Wolfe, D. (1973). *Non-parametric Statistical Methods*. John Wiley and Sons. New York, USA

Hoos, H. & Stutzle, T. (2000). Systematic vs. Local Search for SAT. *Journal of Automated Reasoning*, Vol. 24, pp. 421-481

Houstis, E.; Catlin, A. & Rice, J. (2002). PYTHIA-II: A Knowledge/Database System for Managing Performance Data and Recommending Scientific Software, ACM Transactions on Mathematical Software (TOMS) - Special issue in honor of John Rice's 65th birthday, Vol. 26, No. 2, (June 2000)

Hutter, F.; Hamadi, Y.; Hoos, H. & Leyton-Brown, K. (2006). Performance prediction and automated tuning of randomized and parametric algorithms. *Lecture Notes in Computer Science, Principles and Practice of Constraint Programming*, Vol. 4204, pp. 213-228

Kadioglu, S., Malitsky, Y., Sabharwal, A., Samulowitz, H., & Sellmann, M. (2011). Algorithm Selection and Scheduling, *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP2011)*, Italy, September 2011

Kalousis, A., & Theoharis, T. (1999). NOEMON: Design, implementation and performance results of an intelligent assistant for classifier selection, *Intelligent Data Analysis*, Vol. 3, No. 5, pp. 319-337, ISSN: 1088467X

Kotthoff, L.; Gent, I. & Miguel I. (2011). A Preliminary Evaluation of Machine Learning in Algorithm Selection for Search Problems. In: *AAAI Publications, Fourth International Symposium on Combinatorial Search (SoCS)*, Borrajo, Daniel and Likhachev, Maxim and López, Carlos Linare, pp. 84-91, AAAI Press, Retrieved from: http://www.aaai.org/ocs/index.php/SOCS/SOCS11/paper/view/4006

Lagoudakis, M. & Littman, M. (2000). Algorithm Selection Using Reinforcement Learning. *Proceedings of the Sixteenth International Conference on Machine Learning*. P. Langley (Ed.), AAAI Press, pp. 511–518

Lagoudakis, M. & Littman, M. (2001). Learning to select branching rules in the dpll procedure for satisfiability. *Electronic Notes in Discrete Mathematics*, Vol. 9, (June 2001), pp. 344-359

Lawler, E.; Lenstra, J.; Rinnooy, K. & Schmoys, D. (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, New York, USA

Leyton-Brown, K.; Nudelman, E.; Andrew, G.; McFadden, J. & Shoham, Y. (2003). A portfolio approach to algorithm selection. *Proceedings of International joint conference on artificial intelligence*, Vol. 18, pp. 1542-3

Li, J.; Skjellum, A. & Falgout, R. (1997). A Poly-Algorithm for Parallel Dense Matrix Multiplication on Two-Dimensional Process Grid Topologies. *Concurrency, Practice and Experience*, Vol. 9, No. 5, pp. 345-389

Lobjois, L. & Lemâitre, M. (1998). Branch and bound algorithm selection by performance prediction. In: *AAAI '98/IAAI '98 Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, Jack Mostow, Charles Rich, Bruce Buchanan, pp. 353-358, AAAI Press, Retrieved from: http://www.aaai.org/Papers/AAAI/1998/AAAI98-050.pdf

Ludermir, T.B.; Ricardo B. C. Prudêncio, R.B.C; Zanchettin, C. (2011). Feature and algorithm selection with Hybrid Intelligent Techniques. *International Journal Hybrid Intelligent Systems* ,Vol. 8, No. 3, pp. 115-116

Madani, O.; Raghavan, H. & Jones, R. (2009). On the Empirical Complexity of Text Classification Problems. *SRI AI Center Technical Report*

Malitsky, Y., Sabharwal, A., Samulowitz, H., & Sellmann M. (2011). Non-Model-Based Algorithm Portfolios for SAT, *Proceedings of the 14th international conference on Theory and Applications of Satisfiability Testing*, Ann Arbor, June 2011

Messelis, T.; Haspeslagh, S.; Bilgin, B.; De Causmaecker, P. & Vanden Berghe, G. (2009). Towards prediction of algorithm performance in real world optimization problems. *Proceedings of the 21st Benelux Conference on Artificial Intelligence*, BNAIC, Eindhoven, pp. 177-183

Michlmayr, E. (2007). *Ant Algorithms for Self-Organization in Social Networks*. PhD thesis, Women's Postgraduate College for Internet Technologies (WIT), Vienna, Austria

Nascimento, A. C. A., Prudencio, R. B. C., Costa, I. G., & de Souto, M. C. P. (2009). Mining rules for the automatic selection process of clustering methods applied to cancer gene expression data, *Proceedings of 19th International Conference on Artificial Neural Networks (ICANN)*, Cyprus, September 2009

Nikolić, M.; Marić, F. & Janičić, P. (2009). Instance-Based Selection of Policies for SAT Solvers. *Lecture Notes in Computer Science, Theory and Applications of Satisfiability Testing*, Vol. 5584, pp. 326-340

O'Mahony, E., Hebrard, E., Holland, A., Nugent, C., & O'Sullivan, B. (2009). Using Case-based Reasoning in an Algorithm Portfolio for Constraint Solving. (2008).

Proceedings of The 19th Irish Conference on Artificial Intelligence and Cognitive Science, Ireland, August 2008

Pérez, O.J., Pazos, R.A., Frausto, J., Rodríguez, G., Romero, D., Cruz, L. (2004). A Statistical Approach for Algorithm Selection. *Lectures Notes in Computer Science*, Vol. 3059, (May 2004) pp. 417-431, ISSN: 0302-9743

Pérez, J., Pazos, R.A., Vélez, L. Rodríguez, G. (2002). Automatic Generation of Control Parameters for the Threshold Accepting Algorithm. *Lectures Notes in Computer Science*, Vol. 2313, pp. 119-127

Quiroz, M. (2009). *Caracterización de Factores de Desempeño de Algoritmos de Solución de BPP*. Master´s thesis, Instituto Tecnológico de Cd. Madero, Tamaulipas, México

Reeves, C. (1993). *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, ISBN: 0-470-22079-1, New York, USA

Rice, J. R. (1976). The algorithm selection problem. *Advances in Computers*, Vol. 15, pp. 65-118

Rice, J.R. (1968). On the Construction of Poly-algorithms for Automatic Numerical Analysis. *Interactive System for Experimental Applied Mathematics*, M. Klerer & J. Reinfelds, (Ed.) Academic Press, Burlington, MA, pp. 301-313

Ruiz-Vanoye, J. A., Pérez, J., Pazos, R. A., Zarate, J. A., Díaz-Parra, O., & Zavala-Díaz, J. C. (2009). Statistical Complexity Indicators Applied to the Vehicle Routing Problem with Time Windows for Discriminate Appropriately the Best Algorithm, *Journal of Computer Science and Software Technology*, Vol. 2, No. 2, ISSN: 0974-3898

Samulowitz, H. & Memisevic, R. (2007). Learning to solve QBF. In: *AAAI-07*, pp. 255-260, retrieved from: https://www.aaai.org/Papers/AAAI/2007/AAAI07-039.pdf

Schiavinotto, T. & Stützle, T. (2007). A review of metrics on permutations for search landscape analysis. *Computers & Operations Research*, Vol. 34, No. 10, (October 2007), pp. 3143-3153

Silverthorn, B. & Miikkulainen, R. (2010). Latent Class Models for Algorithm Portfolio Methods. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*

Smith-Miles, K. & Lopes, L. (2012). Measuring instance difficulty for combinatorial optimization problems. *Computers & Operations Research*, in press (accepted 6/7/11)

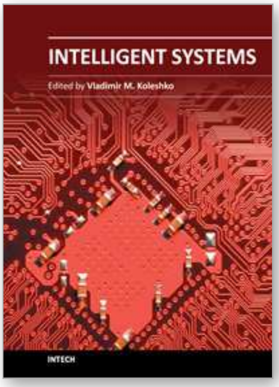
Smith-Miles, K.; James, R.; Giffin, J. & Tu, Y. (2009). Understanding the relationship between scheduling problem structure and heuristic performance using knowledge discovery. In: *Learning and Intelligent Optimization,* LION-3, Vol. 3, Available from: lion.disi.unitn.it/intelligent-optimization/LION3/online_proceedings/35.pdf

Soares, C. & Pinto, J. (2003). Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results. *Machine Learning*, Vol. 50, No. 3, pp. 251-277

Streeter, M; Golovin, D. & Smith, S. F. (2007). Combining multiple heuristics online. In: *AAAI 2007, Proceedings of the 22nd national conference on Artificial intelligence*, Vol. 22, Anthony Cohn, pp. 1197-1203, AAAI Press, Retrieved from: http://www.aaai.org/Papers/AAAI/2007/AAAI07-190.pdf

Wolpert, D. H. & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67–82

Xu, L.; Hutter, F.; Hoos, H. & Leyton-Brown, K. (2009). SATzilla2009: An automatic algorithm portfolio for SAT. In: *Solver description, 2009 SAT Competition*



Intelligent Systems
Edited by Prof. Vladimir M. Koleshko

ISBN 978-953-51-0054-6
Hard cover, 366 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

This book is dedicated to intelligent systems of broad-spectrum application, such as personal and social biosafety or use of intelligent sensory micro-nanosystems such as "e-nose", "e-tongue" and "e-eye". In addition to that, effective acquiring information, knowledge management and improved knowledge transfer in any media, as well as modeling its information content using meta-and hyper heuristics and semantic reasoning all benefit from the systems covered in this book. Intelligent systems can also be applied in education and generating the intelligent distributed eLearning architecture, as well as in a large number of technical fields, such as industrial design, manufacturing and utilization, e.g., in precision agriculture, cartography, electric power distribution systems, intelligent building management systems, drilling operations etc. Furthermore, decision making using fuzzy logic models, computational recognition of comprehension uncertainty and the joint synthesis of goals and means of intelligent behavior biosystems, as well as diagnostic and human support in the healthcare environment have also been made easier.

Laura Cruz-Reyes, Claudia Gómez-Santillán, Joaquín Pérez-Ortega, Vanesa Landero, Marcela Quiroz and Alberto Ochoa (2012). Algorithm Selection: From Meta-Learning to Hyper-Heuristics, Intelligent Systems, Prof. Vladimir M. Koleshko (Ed.), ISBN: 978-953-51-0054-6, InTech, Available from: http://www.intechopen.com/books/intelligent-systems/algorithm-selection-from-meta-learning-to-hyper-heuristics

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

An Interactive Decision Support System Framework for Social Project Portfolio Selection

Laura Cruz-Reyes[1], Fausto A. Balderas J.[1], Cesar Medina T.[1], Fernando López I.[2], Claudia G. Gómez S.[1], and Ma. Lucila Morales R.[1]

[1] Instituto Tecnológico de Ciudad Madero, México
lcruzr@prodigy.net.mx, {tono.21,lmoralesrdz}@gmail.com,
{rpgamer86,cggs71}@hotmail.com
[2] Universidad Autónoma de Nuevo León, México
flopez65@gmail.com

Abstract. In this paper, we present the development of a Decision Support System (DSS) for the project portfolio selection problem, financed with public funds. The selection of the portfolio is a complex optimization problem with multiple subjective criteria, which are difficult to compare. The Decision Maker (DM) has to find a manageable and most preferred set among many non-dominated (efficient) solutions. There is a vast variety of techniques and software for multi-criteria decision making. However, the portfolio selection is an area that requires more and better software. An interactive "Framework" is presented; it is designed to help the DM to select the best portfolio in a flexible way. The Framework is based on the classic decision process of Simon, the SMART method and a friendly man-machine interface. The SMART method was adapted to allow the DM the discovery of his preferences and to express them on the terms of the objective weights and budget constraints. The graphic user interface assist the DM through the visualization of the impact on the change of preferences and also on the projects within a reference portfolio, this way he can make a decision or adjust the necessary changes.

1 Introduction

This article addresses the development of a Decision Support System (DSS) for project portfolio selection to be financed by public funds (SPP, Social Portfolio Problem). Portfolio selection is a complex optimization problem involving conflicting, subjective criteria difficult to compare.

According to Weistroffer [1], the general problem for the implementation of DSS for project selection is in its beginning and there are few studies on it. Weistroffer criticizes the current DSS and mentions that there is not a methodology that is best for all problems of project portfolios.

O. Castillo et al. (Eds.): Recent Advances on Hybrid Intelligent Systems, SCI 451, pp. 377–391.
springerlink.com

By the above, in the present investigation a framework was developed that allows interaction of the decision maker (DM) with the decision process involved in selecting public portfolio.

One of the main tasks of management in government organizations at all levels is to evaluate a set of social impact projects competing for financial support. With an amount to distribute less than the demand, the benefit to all competing projects cannot be granted. The decision on the allocation of resources is regularly held by a person who will make the decision whether or not resources goes to a particular project.

Under certain restrictions determined by the orientation of public policies, quality portfolios projects just be formed which maximize the impact (with ideological connotations of the decision maker) of the chosen solution (see Fig.1). This is an big social problem that the cost of poor solutions is simply immense, but its complexity has prevented so far real progress to solve it.

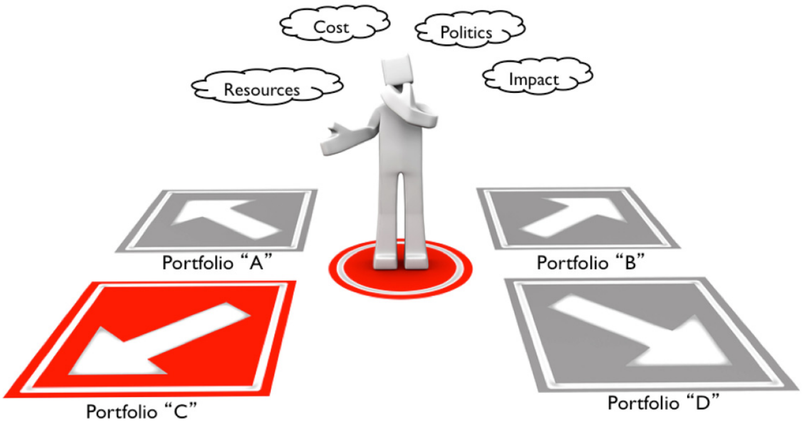


Fig. 1. Decision Maker

Counting on a framework to guide them through an appropriate order of steps, applications and data conversions, large organizations that distribute public resources for public projects may choose, through a framework, projects that have a greater benefit to society and therefore optimize their resources. It is expected that the realization of this project will contribute to decision-makers to make proper use of public resources.

This paper is organized in five parts, starting from the introduction. The second section reviews the general Portfolio Selection Problem that occur during the resources allocation. This problem involves differrent kinds of situation, wich requires different solution. For this reason, the third section describes three solution approaches, based on multi-criteria optimization, for the Social Portfolio Problem (SPP). Besides, it is well known that the decision maker (DM) has to find an acceptable (compromise) solution from among many efficient (non-dominated) solutions given by optimizers. Because the rational capacity of human being is limited for this task, an interactive Decision Suport System (DSS) is imperaty to mainly identify the most preferred solution. Section 4 analyzes the literatura related with frameworks to develope DSS´s. Finally, Section 5 presents our framework proposed for SPP.

2 Project Portfolio Selection

A *project* can be defined as a complex effort, usually less than three years, consisting of interrelated tasks carried out by various organizations, with a clear objective, timetabled and a budget. A *portfolio* is a set of projects undertaken under the administration of an organization. These projects must compete for scarce resources, because usually the resources provided by the organization are not sufficient to carry out all proposed projects for the portfolio. The *project portfolio selection* is the activity of creating a portfolio from the proposed projects of the organization, which met in the best way with the objectives set by the organization, and not exceeding the amount of resources given or breaking other constraints given by managers.

The portfolio selection process use assessment and evaluation techniques divided into three stages:

- *Strategic considerations*. The techniques used at this early stage may help in determining a strategic approach to the overall distribution of the budget for the portfolio.
- *Individual assessment of projects*. Techniques at this stage can be used for evaluating projects independently of the others.
- *Portfolio Selection*. This last stage deals with the portfolio selection based on the parameters of candidate projects. For Example, the project synergy is important because the value of the portfolio is different from the sum of the values of the individual projects.

Following this three-stage process, Archer [13] proposes a set of suggestions that specify the requirements that addresses each phase:

Stategic Considerations Phase

Strategic decisions relating to the portfolio approach and budget considerations must be made in a broader context that takes into consideration internal and external business factors before the portfolio is selected.

A framework for selecting projects must be sufficiently flexible to allow interested parts to choose a way forward with specific techniques or methodologies which they are comfortable, to analyze relevant data and make decisions about the types of projects on hand.

To simplify the portfolio selection process, it should be organized in a number of stages, allowing decision makers moving logically toward a comprehensive consideration of the projects most likely to be chosen based on theoretical models.

Users should not be overloaded with unneeded data, but should be able to access relevant information when needed.

Project Evaluation Phase

Common Measurements should be chosen that could be calculated separately for each project under consideration. These allow a fair comparison of the projects during the selection process.

The current projects that have reached milestones should be re-evaluated while new projects are being considered for selection. This allows a combined portfolio to be generated without violating the resource constraints at regular intervals due to (a) compliance of the project or neglect, (b) new project proposals, (c) changes in strategic focus, and (e) changes in the environment.

The filtering should be used, based on carefully specified criteria to eliminate projects to be considered before the portfolio selection process is carried out.

Portfolio Selection Phase

Interactions of projects (synergy) through direct dependencies or resource competition should be considered in selecting the portfolio.

The selection of the portfolio must take into account the time-dependent nature of the resource consumption of projects.

Decision makers must provide interactive mechanisms to control and cancel any portfolio generated by any algorithm or model, and also should receive feedback from the consequences of such changes. The project portfolio selection should be adaptive to environments for decision support groups.

3 Social Portfolio Problems, SPP

The selection of projects of a social portfolio, unlike the selection with other types of projects (Research and Development, Information System and Financial investment), requires a special treatment for the following reasons [2]:

a) The quality of projects is usually described by multiple-criteria that are often in conflict.
b) Often, the requirements are not known accurately. Many concepts have no mathematical support due their entire subjective nature.
c) The heterogeneity among potential projects in a portfolio, making it difficult to compare.
d) Information provided by the DM, is not strong so it can be called *incomplete preference information.*
e) The impact on social prosperity, which is the most important concept of the problem of public project portfolio, is a variable of subjective nature and usually takes very long term to achieve the expected benefit, depends on the DM put a value on this variable for each project.

The above points are characteristic of public projects such as projects focused to education, health, public transport and general prosperity. To our knowledge, there are only two scientific approaches concerning this type of project portfolio [2]:

1) The most used is the cost-benefit analysis. Some statistical techniques can reduce the number of variables of a project to easily represent a monetary value.
2) Using multi-criteria analysis to explore the preferences of the DM as well as manage the inherent complexity of DM. Multi-criteria analysis is a good alternative to overcome the limitations of cost-benefit analysis, since it can handle ambiguous and intangible preferences and conditions of veto. Multi-criteria analysis provides techniques for selecting the best project or a small set of best projects that are equivalent, classifying the projects into several categories according to predefined preferences or priorities given by the DM.

3.1 Multi-criteria Solution Approaches

Using multi-criteria analysis, the decision on which projects should receive funding, may be based on the best individual projects or based on the best portfolio on the set of all feasible portfolios. For the problem of public portfolio is insufficient to compare projects with each other, as this does not guarantee that all the best projects is the best portfolio.

For example, under the scenario of portfolio selection, it is possible to reject a good project (in terms of social impact), because it requires an excessive financing that might conflict with the preferences of a decision maker who wants to encourage more projects.

The preferences of the decision maker, to form a portfolio, can be modeled from different perspectives, using different approaches to reach the goal. These approaches depend on who the decision maker is (A single person or a heterogeneous group), and how much effort the DM is willing to invest in finding the solution to the problem. This work considers the information about the impact of projects; their quality can be obtained from the DM using three different approaches to solving this problem, two of them are described in the following sections.

Value Function

Given a set of premises (insufficient funds, projects that meet minimum requirements of acceptability, ethical conduct, etc.), it is possible to create a value model for portfolios from the perspective of the main decision maker (SDM, Supra Decision Maker). The set of premises to consider might be based on the following assumptions [9]:

a) Each project and each portfolio has a subjective value for the SDM, even if the initial value can not be quantified.
b) The SDM have a consistent system of preferences or has aspirations to build it.

c) The SDM want to invest a considerable amount of mental effort in order to define the consistent set of preferences and produce the value model.

Priority Ranking

Flerida develops a model for the composition of socially oriented portfolios [14]. Information concerning the quality of the projects is in a priority (ranking) of projects, which can be obtained by a suitable application of an adequate multi-criteria method, but the ranking does not take a proper assessment of social impact.

The model provides a preference relation on the portfolio positions in the ranking of projects, project costs and the rejection of the DM toward costly projects. A better portfolio is mainly found through multi-objective optimization respect to the violations of preset preferences of the decision maker and the cardinalities of competing portfolios.

An example of this approach is the division of the ranking in five categories labeled as Vanguard, Medium-High, Medium, Medium-low and Rear. With this characterization four preference relations are built according to the ranking: absolute preference, strict preference, weak preference and indifference.

It is necessary to compare the quality of the portfolios to find the best. The best portfolio is defined not only by the quality of the projects but also by the number of projects it contains. Some discrepancies may be acceptable between the information given by the ranking and the decisions concerning the approval of some projects, provided that this increase in the number of projects in the portfolio. However, this inclusion should be controlled because the admission of unnecessary discrepancies when comparing portfolios is equivalent to underestimate the ranking information. The model of Flerida considers 3 objectives:

1) Number of strong discrepancies (D_s)
2) Number of weak discrepancies (D_w)
3) Portfolio cardinality (n_c)

The portfolio to be elected should be the best solution to the multiobjective problem:

$$\begin{gathered} \textit{Minimize}\ (D_s,\ D_w) \\ \textit{Maximize}\ (n_c) \qquad (1) \\ C \in R_F \end{gathered}$$

Where C denotes portfolios and RF is the feasible region according to budgetary constraints.

4 Frameworks for the Development of Decision Support Systems

In the work presented by Jichang Dong, et al. [3], a framework for portfolio selection is proposed, which is adaptable to the needs of financial organizations and individual investors. It focuses on the implementation of a web-based framework, adopting technologies such as online analytical processing, as an additional tool for analysis, as well as the parallel use of a virtual machine to improve overall performance. This framework, however, leaves many problems unresolved, like the problems of transaction costs, multi periods and incomplete information. On the other hand, it does not use many of the existing methodologies and models because they are too complex and require lots of input data.

RPM is a methodology for decision support to analyze portfolio problems with multiple criteria [4]. The RPM framework extends the use of preferential programming methods in the portfolio problem. RPM is based on the calculation of the set of non-dominated portfolios accord to the incomplete information. It includes performance measures that help to analyze the attractiveness, robustness of portfolios and individual project proposals. According to the authors, this approach provides a systematic and transparent framework for decision support that may have a valuable contribution in the selection of project portfolio, especially when the number of proposals is high. In summary the RPM approach provides:

- A scoring model applicable to the evaluation of the projects.
- Proactive Analysis of uncertainty parameters (robustness).
- Identification of projects to be unquestionably included or excluded.
- Analysis and negotiation of projects on the border.
- Transparency of individual projects through performance measures.
- Tentative/split Conclusions at any stage of the selection process.
- A variety of functions to plot interactively the decision support.

In the research done by Castro [6], it is designed an extensible framework that serves to adjust algorithms for exploration and optimization of the space of the project portfolio of R & D in public organizations by implementing a factorial experiment. The framework can also be used for adjusting parameters in optimization methods for the project portfolio of R & D in public organizations.

The implementation of the framework is incomplete without logbook functions and data recovery; there is no possibility to compare different methods for portfolio optimization.

The framework was designed following a three-tier architecture design paradigm: the Model-View-Controller. The functions are grouped into packages seeking a balance between the internal cohesion of each packet and the coupling with the rest of the packets.

Among the most representative researchers are Dong [3] and Liesio [4]. Table 1 shows these and others. The analysis of related work reveals that the activities that have been less addressed can be seen in column 5 and column 7, they are related with the generation of a reference portfolio and the architecture design, respectivately. This paper proposes a DSS that cover all the analyzed stages of the decision-making through a comprehensive architectural design.

Table 1. Comparative table of frameworks for decision support systems

-	Project Capture		Interaction with the DM	Reference Portfolio Generation	Architechture	
	Data	Preferences			Conceptual	Physical
Dong, et. al [3]	✓	✓	✓	-	✓	-
Castro [6]	✓	-	-	-	-	✓
Liesio [4]	✓	✓	✓	-	✓	-
Lourenco [7]	✓	-	✓	-	-	-
Yeh, et. al. [8]	-	✓	✓	-	✓	-
Este trabajo	✓	✓	✓	✓	✓	✓

5 Framework Proposal

We designed an interactive DSS to assist the DM to select the best portfolio in a flexible manner. The core of the DSS is a Framework, which includes the SMART method and a friendly man-machine interface with the next characteristics:

1) The SMART method was adapted to allow the DM the discovery of their preferences and to express them in terms of weights of the objectives and budget constraints.
2) The Framework is based on the classical process of decision making by Simon [9], and the model-view-controller paradigm.
3) The graphical interface assists the DM by visualizing the effects of changes in their preferences and projects on a reference portfolio, so a decision can be made.

5.1 SMART Method for Preferences Elicitation

To obtain the preferences, the SMART (Simple Multi Attribute Rating Technique) [10] method was adapted; at each stage the DM is involved with the required information:

1) Identify the *alternatives* and relevant *attributes* for the problem and for each attribute assign a value to each alternative. These values form a descending *order* of preference.
2) For each *attribute* determine the associated *absolute weight* expressed as a percentage of preference and according to the established order.
3) For each *alternative* distribute the *absolute weight* associated with the evaluated attribute according to their preferences.

4) Calculate the *relative weights* of each alternative with its attribute, based on the percentages given.
5) Perform sensitivity analysis to check the consistency of preferences; this implies repeating the process until the DM requires it.

5.2 Model-View-Controller

The MVC (Model-View-Controller), see Figure 2, allows us to separate the control logic (what needs to be done but not how), business logic (how things are done) and the presentation logic (how to interact with the user). MVC is recommended for design interactive web applications [11]. Using this type of pattern is possible to achieve higher quality, an easier maintenance and extension. One of the most important things that allow the use of this pattern is to normalize and standardize the software development.

The architecture of the DSS framework was design by the Model-View-Controller pattern with the next general functions:

- In the view layer, the user interface is developed.
- In the controller layer, the problem of portfolio selection of social projects is solved.
- In the Model layer, there are the protocols for: data import, incorporation of obtaining methods and incorporation of optimizers. Also this layer includes two entities: database and libraries.

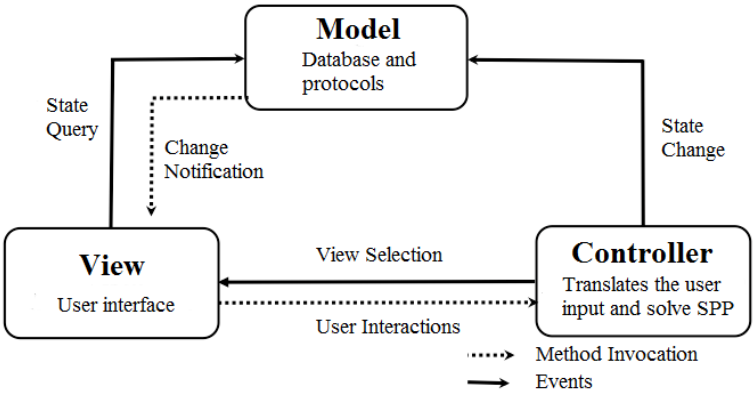


Fig. 2. MVC pattern for SPP

5.3 Decision Making Process Based on Simon

Another perspective of the architecture of the framework was design following the traditional process of decision making of Simon: Intelligence, Design and Selection, as shown in Figure 3. The interaction with the DM is detailed in Figure 4.

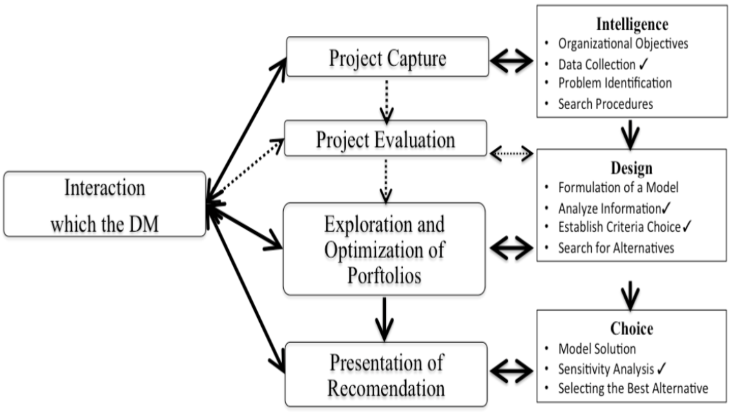


Fig. 3. Decision making process for SPP

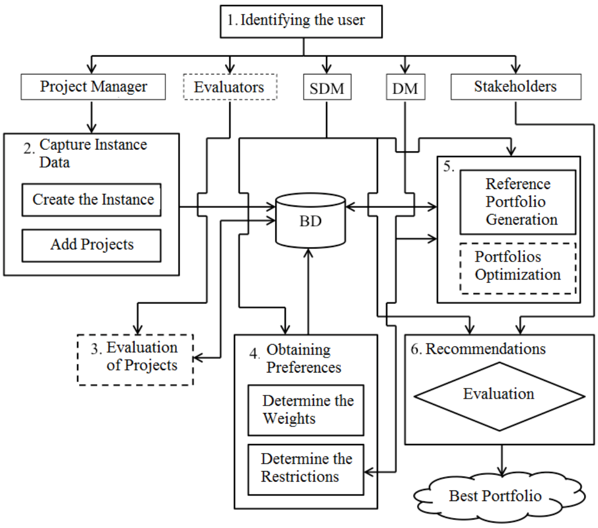


Fig. 4. Interaction with the DM

5.4 *Architecture*

The architecture shown in Figure 5 is the result of the integration of three conceptual independent designs: model-view-controller (MVC in Figure 2), decision making processs (DMP in Figure 3) and user interaction (Figure 4). The modules with solid lines were implemented with simple stategies to show the feasibility of the proposed method for preference elicitation. The dotted lines represent what was left out for future work.

The description of the final architecture follows the DMP phases (intelligence, design and election), for each phase the three layers of MVC are explained. Because in the social portfolio selection users can participate in all processes, restricting their tasks, we decided not to make a roles distinction in this architecture and include it as an access control scheme.

Intelligence

The intelligence phase within the view layer, involve the data capture of the instance, the project evaluation capture, the capture of preferences. All of these are within the Intelligence since they involve an in depth preliminary analysis.

In the controller layer, there are the goals, scales, projects and project evaluation breakdown of preferences methods, each of these in this layer undergoes a consistency and coherence checker.

In the model layer is the protocol for data import, so that data can be processed, these should preferably be in the same format, and the protocol to incorporate techniques of project evaluation and preference disaggregation.

Design

Here we find the capture of expected portfolios provided by the DM. It will be a reference profile of alternatives and used for the comparison of portfolios, all in the view layer, due this is what the end user see. The DM forms his reference portfolios and has the possibility to compare these portfolios against generated by optimizers, recommenders and benchmark portfolios.

In the controller layer, we have the methods of obtaining preferences and portfolio evaluation; here also all of of these undergoes a consistency and coherence checker.

In the Model layer, we need protocols for the incorporation of new methods to get the decision maker's preferences by elicitation and methods for the portfolio evaluation

Election

In the view layer, we have the visual representation of the recommendation, it represents visually the portfolios for the user and the projects that are in the portfolio, and it presents the information in a visual and friendly manner.

In the controller layer, we Optimization Manager, which is where the optimizers developed by the network, in the future it is planned that in this part the algorithms for different approaches to the public portfolio problem will be stored. The architecture provides the existence of three approaches for solution (function value, rank, Fuzzy Preference Relation), which help the DM depending on the amount of effort willing to contribute. Also, here is also the recommendation manager.

The Model layer have the protocol for the incorporation of portfolio optimizers and the protocol to incorporate recommenders. The latter is necesarry to overcome the limited rational capacity of the DM to deal with a big set of optimized portolios.

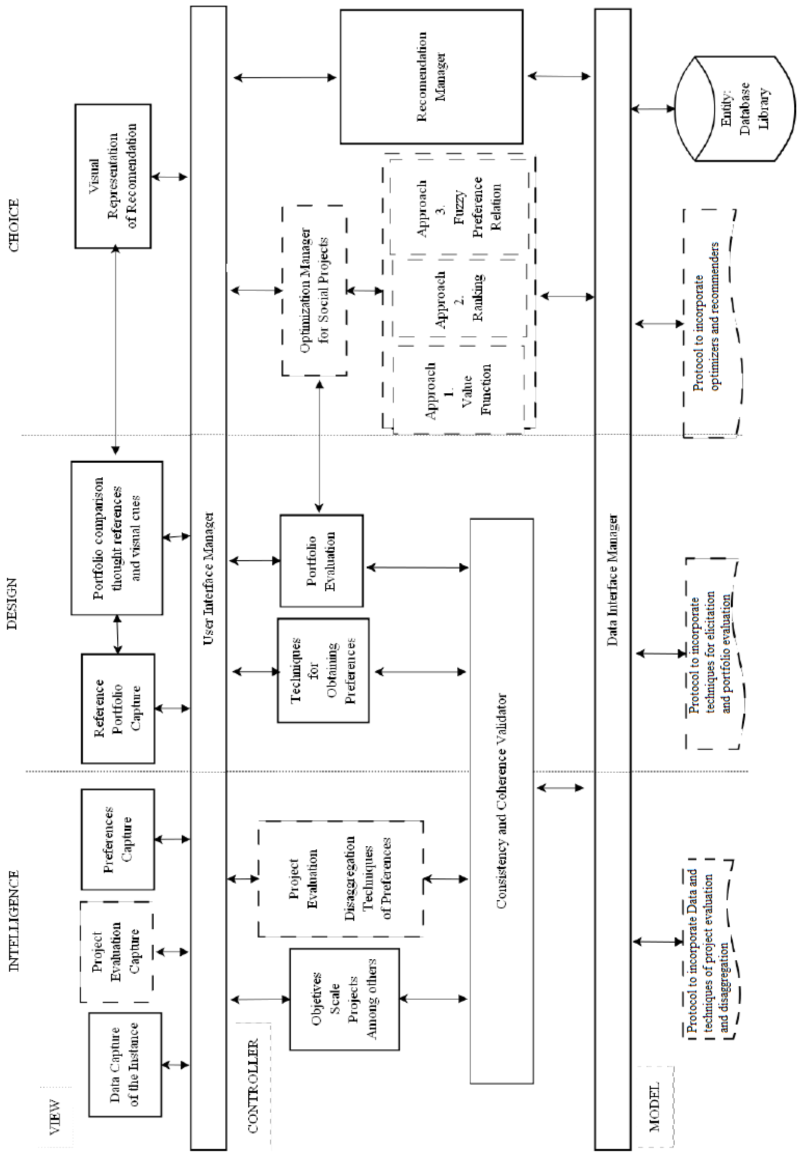
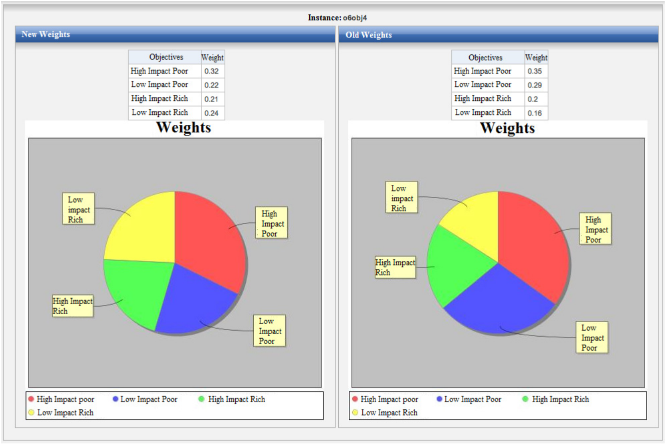

INTELLIGENCE
DESIGN
CHOICE
VIEW
Data Capture of the Instance
Project Evaluation Capture
Preferences Capture
Reference Portfolio Capture
Portfolio comparison thought references and visual cues
Visual Representation of Recomendation
User Interface Manager
CONTROLLER
Objetives Scale Projects Among others
Project Evaluation
Disaggregation Techniques of Preferences
Techniques for Obtaining Preferences
Portfolio Evaluation
Optimization Manager for Social Projects
Approach 1. Value Function
Approach 2. Ranking
Approach 3. Fuzzy Preference Relation
Recomendation Manager
Consistency and Coherence Validator
Data Interface Manager
MODEL
Protocol to incorporate Data and techniques of project evaluation and disaggregation
Protocol to incorporate techniques for elicitation and portfolio evaluation
Protocol to incorporate optimizers and recommenders
Entity: Database Library

Fig. 5. Process architecture

5.5 *User Interface*

The Figures 6 and 7 correspond to two screens showing a part of the interaction of the DM with the DSS. The first shows the result of applying the SMART method to generate two reference portfolios. In the second, the user can choose the final portfolio looking at the impact of the portfolios, which are differentiated by colored bubbles.



Instance: o6obj4

New Weights

Objectives	Weight
High Impact Poor	0.32
Low Impact Poor	0.22
High Impact Rich	0.21
Low Impact Rich	0.24

Old Weights

Objectives	Weight
High Impact Poor	0.35
Low Impact Poor	0.29
High Impact Rich	0.2
Low Impact Rich	0.16

Weights
Low impact Rich
High Impact Poor
High Impact Rich
Low Impact Poor
● High Impact poor ● Low Impact Poor ● High Impact Rich ● Low Impact Rich
Weights
Low impact Rich
High Impact Poor
High Impact Rich
Low Impact Poor
● High Impact poor ● Low Impact Poor ● High Impact Rich ● Low Impact Rich

Fig. 6. Weights Editing

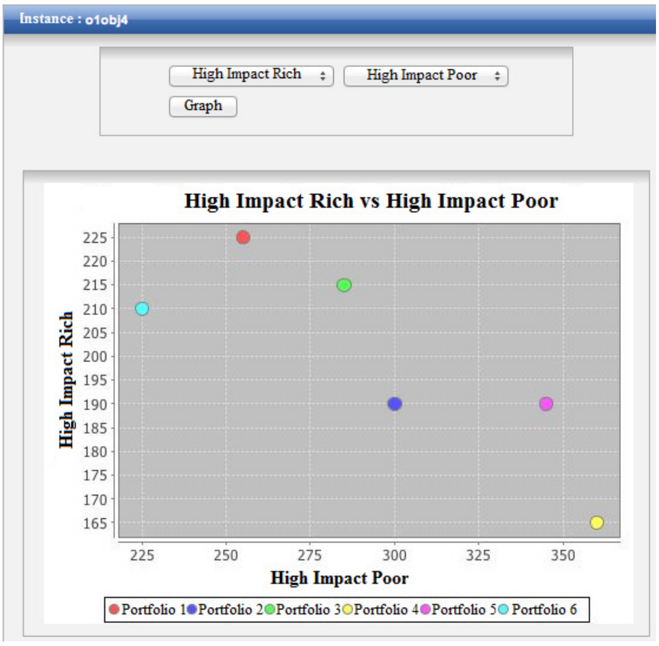


Fig. 7. Presentation of the recommendation

6 Conclusions and Future Work

A DSS was developed for the portfolio selection problem of social projects with the following contributions:

a) A Framework based on MVC that involves the entire decision making process (Intelligence, Design, Election) and its interaction with the DM.
b) A simple method for obtaining preferences of a single DM.
c) A simple and intuitive interface, which facilitates the obtaining of DM preferences, without so many complications, and visually displays the effects of changes in their preferences.

The proposed solution helps to solve some issues involved with the interaction of a single DM whose preferences are difficult to obtain. Because in the subjective world, the opinions vary widely and consensus between DM´s and the organization is hard to find**,** we need a new robust method that consider all this interactions. Besides, as a future work an evaluation in terms of usability is considered.

Other recommendations of future work are:

- Design and implement of the DSS with multiple views, either for each type of user, the available time of the DM, and the access control scheme.
- Develop protocols to have and extensible DSS, that can be easily upgraded with new methods.
- Complement the DSS with different solution approaches to the problem of social project portfolio, as well as the group decision support for these approaches.

References

[1] Weistroffer, H.R., Smith, C.H.: Decision support for portfolio problems. Southern Association of Information Systems (SAIS). Savannah, Georgia (2005)

[2] Fernández, E., Navarro, J.: A genetic search for exploiting a fuzzy preference model of portfolio problem with public projects. Kluwer Academic Publishers (2003)

[3] Dong, J., Du, H.S., Wang, S., Chen, K., Deng, X.: A framework of web based decision support systems for portfolio selection with olap and pvm. Decision Support Systems 37(3), 367–376 (2004)

[4] Liesiô, J.: Portfolio decision analysis for robust project selection and resource allocation. Helsinki University of Technology (2008)

[5] Sprague, R.H.: A framework for the development of decision support systems. MIS Quarterly, 1–26 (1980)

[6] Castro, M.A.: Desarrollo e Implementación de un Framework para la Formación de Carteras de Proyectos de I&D en Organizaciones Publicas. MS Thesis (2007)

[7] Lourenco, J.C., Costa, C.A.: PROBE a multicriteria decision support system for portfolio robustness evaluation. Technical report, Working Paper LSEOR 09.108, London School of Economics, London (2009)

[8] Yeh, C.H., Deng, H., Wibowo, S., Xu, F.: Fuzzy multicriteria decision support for information systems project selection. International Journal of Fuzzy Systems 12(2), 170–174 (2010)

[9] Navarro, J.: Herramientas inteligentes para la evaluación y selección de proyectos de investigación-desarrollo en el sector público. PhD Thesis, Universidad Autónoma de Sinaloa, Sinaloa, México (2005)

[10] Edwards, W.: How to use multiattribute utility measurement for social decision making. IEEE Transactions on Man and Cybernetics 7(5), 326–340 (1977)

[11] Sistemas de Información Cooperativos Universidad de Malaga. Java Server Faces Tutorial. Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga (2011)

[12] Russell, A.: Managing High-Technology Programs and Projects, 2nd edn. Wiley, New York (1992)

[13] Archer, N.P., Ghasemzadeh, F.: An integrated framework for project portfolio selection. International Journal of Project Management 17, 207–216 (1999)

[14] Fernandez, E., Felix, L.F., Mazcorro, G.: Multi-objective optimization of an outranking model for public resources allocation on competing projects. International Journal of Operational Research 5(2), 190–210 (2009)

Chapter 4

Explaining Diverse Application Domains Analyzed from Data Mining Perspective

Alberto Ochoa, Lourdes Margain, Rubén Jaramillo,
Javier González, Daniel Azpeitia, Claudia Gómez,
Jöns Sánchez, Julio Ponce, Sayuri Quezada,
Francisco Ornelas, Arturo Elías, Edgar Conde, Víctor Cruz,
Petra Salazar, Emmanuel García and Miguel Maldonado

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/49947

1. Introduction

This chapter proposal explains the importance of adequate diverses application domains in different aspects in a wide variety of activities of our daily life. We focus our analysis to different activities that use social richness data, analyzing societies to improve diverse situational activities based on a decision support systems under uncertaainty. To this end, we performed surveys to gathering information about salient aspects of modernization and combined them using social data mining techniques to profile a number of behavioural patterns and choices that describe social networking behaviours in these societies.

We will define the terms "Data Mining" and "Decision Support System" as well as their contrast and roles in modern societies. Then we will describe innovative models that captures salient variables of modernization, and how these variables give raise to intervening aspects that end up shaping behavioural patterns in social aspects. We will describe the data mining methodologies we used to extract these variables in each one of these diverse application domains including the analysis of diverse surveys conducted in diverse societies, and provide a comparative analysis of the results in light of the proposed innovative social model.

On the rest proposed chapter, we will describe how our model can be extended to provide a means for identifying potential social public politics. More particularly, we make allusion to behavioural pattern recognition mechanisms that would identify the importance of use techniques from Data Mining. We will close with concluding remarks and extended discussions of our approach and will provide general guidelines for future work in the area of application of Data Mining in diverse application domains, including further analysis on how those public politics organize and operate in social rings, and how they use technology to that end. While our main focus will be on pure social networks such as Facebook. Our literature review will include cases of implementation of correct public politics, and some issues, challenges, opportunities, and trends about this diverses social problems.

The proposal of this chapter is to explain the importance of use Social Data Mining in a wide variety of activities in our daily life, many of these activities, which are online and involve many social networkings using in many ways using Media Richness. Social Data Mining techniques will be useful for answering diverse queries after gathering general information about this given topic. This kind of behaviors will be characterized by take a real implementation of a correct solution, each one of these taking diverse models or multi agents systems for adequate this behavior to obtain information to take decisions that try to improve aspects very important of their lifes organized in different application and fields of knowledge.

First, in section 1 of this Social Data Mining techniques will be useful for answering diverse questions after gathering general information about the given topic. This type of behaviors will be characterized by a real application of a correct solution, each one of these taking diverse models or multi agents systems. This is for adequate this behavior to have information and make decisions that try to improve aspects very important of their lifes organized in different application and fields of knowledge.

First, in section 1 of this chapter explain the concept of Social Data Mining and as this behavior affect in different way to people in differnet aspects in societies´ people –Viral Marketing to determine boughts on inmobilarie sector–. In other sections we explain the way to generate a correct analysis In correspondent sections we explain the way to make a correct analysis of diferent activities of daily life as in Electrical Industry (section 2), Classification of Images and its analysis which explain the effects in their analysis including Medical advances (section 3), a comparative analysis using people profile according to describe a possible social benefits in diverse applications domains (section 4). In section 5 we explain the results obtained in e-commerce data mining and emergent kind of techniques which resolve and propose specific kind of marketing according at life style of consumers, and in the section 6 are try to describe the use of Data Mining to Mobile Ad Hoc Networks Security which will be used to determine the possible changes on our modern society. In section 7 we described another specfic applications domains as: organizational models, organizational climate, zoo applications to classify more vulnerable species or identify the adequate kind of avatars on a roll multigame players and finally our conclusions about the future of Data Mining in diverses uses to different activities of our daily life.

2. Data mining and their use on viral marketing

The use of traditional media like radio, television and newspaper, has been replaced by new digital media like social networks Facebook and Twitter. According to (Salaverría, 2009) the increase of broadband users, both on mobile devices, home and workplace, has raised the replacement of traditional media by digital media. Likewise, mentions that these new tools allow the user to interact with the issuer, thanks to several factors that facilitate interaction such as, frequency of updates, including multimedia such as videos and photographs, among others.

On the other hand, (Orihuela, 2002) mentions that existing Internet interactivity has been subverting the paradigms within communication processes in mass media. As (Salaverría, 2009), mentions the ability of interactivity, customization and upgrade, as central in replacing traditional media to digital (Figure 1).

Figure 1. Interaction between users of social networks.

In their study, (Orihuela, 2002), concludes that the public announcement raised in the new digital media is sufficient justification to redefine the requirements in the media, the procedures and content of information, all within trends changing as a result of network usage.

Due to the above, the social networks like Facebook, are an important tool in the marketing strategy of companies. Its low cost (sometimes zero) helps not only in communicating the customer value, but also improves the customer-consumer relationships. According to (Orihuela, 2002), social networks like Facebook sometimes take characteristics of traditional media, however, incorporate a higher level of interaction.

2.1. Corporate use of social networks

After analyzing the above, we can say that the use of social networks helps greatly reducing advertising costs and implementation of new marketing strategies. But even if there are different tools to monitor and observe the behavior of users, there is little research evidence that reveals different patterns of consumption, transmission of messages or lack of them, and observes the behavior of these consumers on trademarks and their experiences with them within the social networks like Facebook.

According to (Salaverría, 2009) the online advertising industry grew by 800 percent from 2004 to 2009 demonstrating a steady development in which social networks and contextual advertising play an important role in the marketing or advertising on social networks, without But there is no scientific evidence on the behavior of users in such networks and the dissemination of messages received and sent within these networks and what encourages you to do or not.

(Sandoval et al., 2010) mentions that social networks have changed the human relations approach and have potentiating its most important feature: Easy to find and develop relationships with other members with similar interests. Similarly mention that social networking services have proliferated targeting people in specific regions or some similar interests as, ethnic, religious, sexual and political (Figure 5). Thus, the fact of having a community segment showing a potential interest in a particular company or product, is useful when performing a specific marketing strategy. In addition to marketing strategies, companies can use such networks in the recruitment, internal communication and interaction with consumers.

Figure 2. Nested groups of similar interests.

2.2. Research objectives

Having analyzed the use of social networks in business, the importance of the restaurant industry in Mexico and specifically the problem of insecurity in Juarez, perform the following research questions:

- What specific objectives seek restaurant sector companies to use social networks?
- What digital social network use most frequently?
- What percentage of these companies has replaced the traditional media advertising advertising on social networks?
- What marketing strategies used in online social networks?
- What correlation is there between; use of social networks and increased sales?
- When beginning their presence within social networks?
- How many users is made up your network?
- How often publish information within social networks?
- What correlation exists between the periodicity of the publications and the time spent in the network, with the number of users in the network?

2.3. Methodology

The conclusive results of this research were obtained through an exploratory study of the use of social networks in companies in the restaurant industry in Juarez and factorial designs were performed to find some correlations between different variables.

First we made a query of the restaurant industry to recognize his presence in Mexico and in the locality. This was done through the National Chamber of the Restaurant Industry and Seasoned Foods (CANIRAC) and National Chamber of Commerce (CANACO) found in the locality.

From the list of registered companies in the industry by these cameras restaurateur, was searched to select those that were present within the digital social networks, regardless of upgrade or number of users connected to their groups. Were interviewed and application of survey of 20 companies with the largest number of users within your network, to meet their business openly in online social networks, specifically Facebook. Took place through a careful study of social networks to find that participation in that network have to know the frequency and topics of their publications, as well as general information of relevance to publish within their Facebook page. He knew the date they started their activities in the network.
Once the information was held after his capture to be analyzed in statistical software to find relevant values.

3. Competitive learning for self organizing maps used in classification of partial discharge

Competitive learning is an efficient tool for Self Organizing Maps, widely applied in variety of signal processing problems such as classification, data compression, in anothers. With the huge volumes of data being generated from the different systems everyday, what makes a system intelligent is its ability to analyze the data for efficient decision-making based on known or new cluster discovery. **The partial discharge (PD) is a common phenomenon which occurs in insulation of high voltage, this definition is given in [1]. In general, the partial discharges are in consequence of local stress in the insulation or on the surface of the insulation**. We evaluate the performance of algorithms in which competitive learning is applied of partial discharge dataset, quantization error, topological error and time in seconds per training epoch. The result from classification of PD shows that *Winner-takes-all* **(WTA)** has better performance than *Frequency Sensitive Competitive Learning* **(FSCL)** and *Rival Penalized Competitive Learning* **(RPCL).** The first approach in a diagnosis is selecting the different features to classify measured PD activities into underlying insulation defects or source that generate PD's (Figure 3).

The phase resolved analysis investigates the PD pattern in relation to the variable frequency AC cycle (Cheng et al., 2008). The voltage phase angle is divided into small equal windows. The analysis aims to calculate the integrated parameters for each phase window and to plot them against the phase position (ϕ).

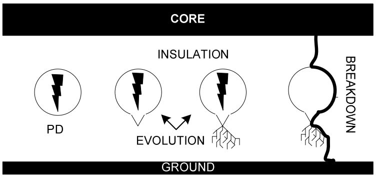


Figure 3. Example of damage in polymeric power cable from the PD in a cavity to breakdown.

- (q_m −ϕ) : the peak discharge magnitude for each phase window plotted against ϕ, where q_m is peak discharge magnitude.

3.1. Self organizing map

The Self Organizing Map developed by Kohonen, is the most popular neural network models (Kohonen, T., 2006 & Rubio-Sánchez, M., 2004). The SOM is a neural network model that implements a characteristics non-linear mapping from the high-dimensional space of input signal onto a typically 2-dimensional grid of neurons. The SOM is a two-layer neural network that consists of an input layer in a line and an output layer constructed of neurons in a two-dimensional grid.

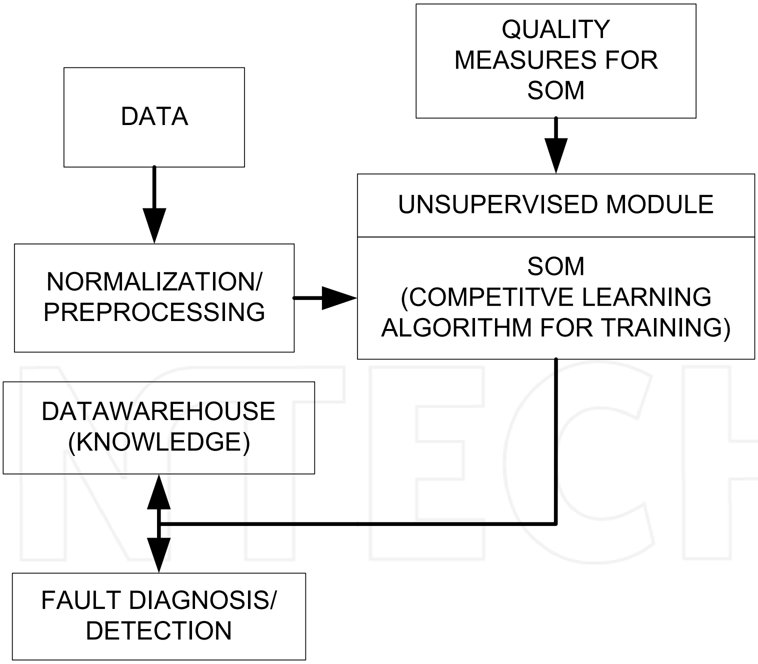

QUALITY MEASURES FOR SOM
DATA
UNSUPERVISED MODULE
NORMALIZATION/ PREPROCESSING
SOM (COMPETITVE LEARNING ALGORITHM FOR TRAINING)
DATAWAREHOUSE (KNOWLEDGE)
FAULT DIAGNOSIS/ DETECTION

Figure 4. The component interaction between SOM.

PD measurements for power cables are generated and recorded through laboratory tests. Corona was produced with a point to hemisphere configuration: needle at high voltage and hemispherical cup at ground. Surface discharge XLPE cable with no stress relief termination applied to the two ends. High voltage was applied to the cable inner conductor and the cable sheath was grounded, this produces discharges along the outer insulation surface at the cable ends. Internal discharge was used a power cable with a fault due to electrical treeing. Were considered the pattern characteristic of univariate phase-resolved distributions as inputs, the magnitude of PD is the most important input as it shows the level of danger, for this reason the input in the SOM the raw data is the peak discharge magnitude for each phase window plotted against (qm −ϕ). **Figure 2** shows the conceptual diagram training. In the cases analyzed, the original dataset is 1 million of items, was used a neurons array of 10×10 cells to extract features. As it is well known, in fact, a too small number of neurons per class could be not sufficient to represent the variability of the samples to be classified, while a too large number in general makes the net too much specialized on the samples belonging to the training set and consequently reduces its generalization capability. Moreover a too large number of neuron per class implies a long training time and a possible underutilization of some of the neural units. PD patterns recognition and classification require an understanding of the traits commonly associated with the different source and relationship between observed PD activity and responsible defect sources. This paper shows the performance of SOM using different competitive learning algorithms to classify measured PD activities into underlying insulation defects or source that generate PD's, its showed that WTA is the better algorithm with less error and training time, but its overall performance are not always satisfactory, being alternative in accord at the performance FSCL or RPCL algorithms.

4. Classification of images using Naive Bayes and J48

This research work's approach is related to artificial vision due to extraction from information contained in images (human faces) by using methods to obtain RGB coloration and statistic values. Extraction takes place by performing several tests of image splitting into different sizes, later classifying sets of instances with data mining techniques, and analyzing classification results to determine which of the algorithms is the best for this particular case.

Knowledge database contains 100 images built from 20 people and 5 pictures each. Mean and standard deviation were employed as statistic values, which are also used as attributes of the instances classified by Naïve Bayes and WEKA J48. It is important to mention that no pixel is disregarded to obtain instances, both of the pixel groups the ones inside face and outside of it are considered. Impact of splitting images into parts.

Table 1 shows that splitting images into both 16 and 64 obtain the same amount of correctly classified instances except with NaiveBayes classifier under cross validation where splitting into 16 obtains 3% better of correctly classified instances, partial conclusion from this table is splitting images into both 16 and 64 is better than splitting them into 4 and no splitting them.

Table 2 shows both Naïve Bayes and J48 under use training test option obtain 100 % of correctly classified instances from splitting images into 4 parts which reveals splitting images helps for classification process.

		Parts			
Classifier	**Test options**	1	4	16	64
NaiveBayes	use training set	94%	100%	100%	100%
J48	use training set	97%	100%	100%	100%
NaiveBayes	cross validation folds 10	85%	90%	97%	94%
J48	cross validation folds 10	71%	86%	85%	85%
NaiveBayes	percentage split 66%	76.4706%	70.5882%	88.2353%	88.2353%
J48	percentage split 66%	64.7059%	55.8824%	64.7059%	64.7059%

Table 1. Results of splitting images into parts including all attributes.

		Parts			
Classifier	**Test options**	1	4	16	64
NaiveBayes	use training set	90%	100%	100%	100%
J48	use training set	92%	100%	100%	100%
NaiveBayes	cross validation folds 10	72%	93%	97%	96%
J48	cross validation folds 10	69%	83%	84%	89%
NaiveBayes	percentage split 66%	58.8235%	88.2353%	94.1176%	94.1176%
J48	percentage split 66%	61.7647%	70.5882%	73.5294%	76.4706%

Table 2. Results of splitting images into parts including best 30% attributes.

4.1. Impact of attribute selection

Table 3 shows that for any test option, both classifiers obtain greater amount of correctly classified instances considering all of the attributes which are 6.

Table 4 shows that both classifiers obtain 100% of correctly classified instances under use training set test option. Under cross validation Naïve Bayes classifies 3% better selecting 8 attributes and J48 classifies 3% considering all of the attributes. Finally under percentage split both classifiers perform better selecting 8 attributes.

		6 attributes	**2 attributes**
Classifier	Test options		
NaiveBayes	use training set	94%	90%
J48	use training set	97%	92%
NaiveBayes	cross validation folds 10	85%	72%
J48	cross validation folds 10	71%	69%
NaiveBayes	percentage split 66%	76.4706%	58.8235%
J48	percentage split 66%	64.7059%	61.7647%

Table 3. Results of attribute selection without splitting images.

		24 attributes	8 attributes
Classifier	Test options		
NaiveBayes	use training set	100%	100%
J48	use training set	100%	100%
NaiveBayes	cross validation folds 10	90%	93%
J48	cross validation folds 10	86%	83%
NaiveBayes	percentage split 66%	70.5882%	88.2353%
J48	percentage split 66%	55.8824%	70.5882%

Table 4. Results of attribute selection splitting images into 4 parts.

Table 5 also shows a 100% of correctly classified instances for both classifiers under use training set test option. Under cross validation , Naïve Bayes classifies equal amount of correctly classified instances selecting 29 attributes as selecting all of them, similar situation occurred with J48 with 1% greater for selecting all of the attributes. Finally, under percentage split both of the classifiers perform better selecting 29 attributes.

		96 attributes	29 attributes
Classifier	Test options		
NaiveBayes	use training set	100%	100%
J48	use training set	100%	100%
NaiveBayes	cross validation folds 10	97%	97%
J48	cross validation folds 10	85%	84%
NaiveBayes	percentage split 66%	88.2353%	94.1176%
J48	percentage split 66%	64.7059%	73.5294%

Table 5. Results of attribute selection splitting images into 16 parts.

Table 6 shows once again a 100 % of correctly classified instances under use training set test option for both cases of attribute selection. Under cross validation and percentage split both of the classifiers perform better selecting 116 attributes.

		384 attributes	116 attributes
Classifier	Test options		
NaiveBayes	use training set	100%	100%
J48	use training set	100%	100%
NaiveBayes	cross validation folds 10	94%	96%
J48	cross validation folds 10	85%	89%
NaiveBayes	percentage split 66%	88.2353%	94.1176%
J48	percentage split 66%	64.7059%	76.4706%

Table 6. Results of attribute selection splitting images into 64 parts.

4.2. Analysis of classifiers effectiveness based on test options

Table 7 shows that J48 performs better than Naïve Bayes without splitting images but not in a significant way. Considering any other splitting image scheme or attribute selection show a 100 % of correctly classified instances.

		Classifiers	
Attributes	Parts	NaiveBayes	J48
All of them	1	94%	97%
30%	1	90%	92%
All of them	4	100%	100%
30%	4	100%	100%
All of them	16	100%	100%
30%	16	100%	100%
All of them	64	100%	100%
30%	64	100%	100%

Table 7. Results of classifiers effectiveness under use training set.

Table 8 shows Naive Bayes performs better than J48 for any splitting image scheme and attribute selection.

		Classifiers	
Attributes	Parts	NaiveBayes	J48
All of them	1	85%	71%
30%	1	72%	69%
All of them	4	90%	86%
30%	4	93%	83%
All of them	16	97%	85%
30%	16	97%	84%
All of them	64	94%	85%
30%	64	96%	89%

Table 8. Results of classifiers effectiveness under cross validation.

Table 9 shows Naïve Bayes performs better than J48 except for selecting best 30% attributes without splitting images. Experiments of splitting images into parts allow concluding that splitting images into 16 parts is enough for satisfactory classification. Statement from previous paragraph can be asserted due to results in Table 1 show splitting images into 64 parts obtains equal amount of correctly classified instances as performing such split into 16 parts, Table 1 even shows a reduction of 3% in correctly classified instances for NaiveBayes classifier under cross validation. Next stage of experiment consisted on selecting best 30%

attributes, which reveals Naïve Bayes generates greater amount of correctly classified instances from splitting images into 16 parts, J48 obtains 5% better in 64 parts under cross validation and 2.9412% in 64 parts under percentage split. Due to improvement for 64 parts is not significant, it is concluded splitting into 16 parts is enough. Experiments of attribute selection allow concluding that selecting best 30% is enough. This can be validated from both table 1 and table 2 which show that splitting images into 64, 16, and 4 parts selecting best 30% obtains greater amount of correctly classified instances than considering all attributes. J48 throws 1% better for splitting into 16 parts and 3% better into 64 parts with all attributes, but this is disregarded due to it is not significant. Experiments analyzing effectiveness of classifiers allow to conclude Naïve Bayes performs better due to it obtains greater amount of correctly classified instances under most splitting images case and test option except for use training set test option and no splitting images.

		Classifiers	
Attributes	Parts	NaiveBayes	J48
All of them	1	76.4706%	64.7059%
30%	1	58.8235%	61.6747%
All of them	4	70.8552%	55.8824%
30%	4	88.2353%	70.5882%
All of them	16	88.2353%	64.7059%
30%	16	94.1176%	73.5294%
All of them	64	88.2353%	64.7059%
30%	64	94.1176%	76.4706%

Table 9. Results of classifiers effectiveness under percentage split.

4.3. Medical visualization in data mining

A field that is becoming a rich area for the application of data mining is that of medical imaging. The tremendous advance in imaging technologies such as X-rays, computed tomography, magnetic resonance, ultrasound and positron emission tomography has led to the generation of vast amounts of data (Figure 5). Scientists are interested, of course, in learning from this data, and data mining techniques are increasingly being applied in these analyses.

There are interesting techniques for finding and describing structural patterns in data as a tool for helping to explain that data and make predictions from it. The data will take the form of a set of examples from the patients. The output takes the form of predictions about new examples. Many learning techniques look for structural descriptions of what is learned, descriptions that can become fairly complex and are typically expressed as sets of rules. Because they can be understood by people, these descriptions serve to explain what has been learned and explain the basis for new predictions. People frequently use data mining to gain knowledge, not just predictions. Databases are rich with hidden information that can be used for intelligent decision making. Classification and predictions are two of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help provide us with a better understanding of the data at large.

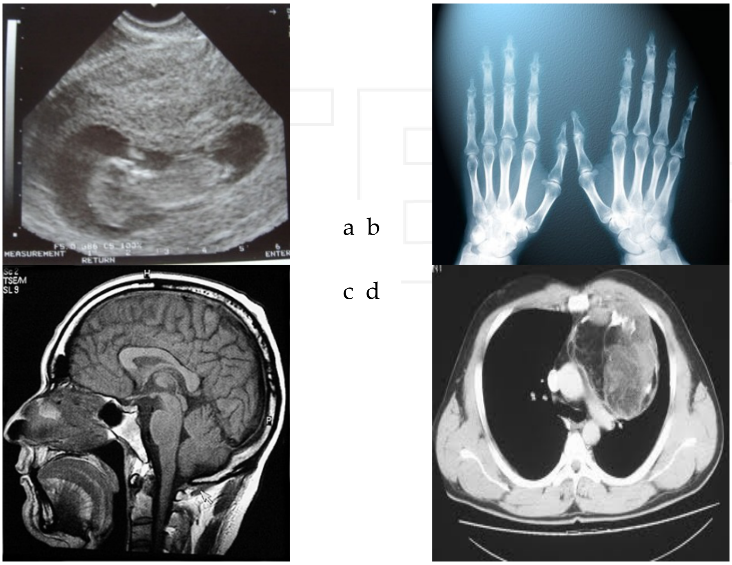


Figure 5. Examples of medical imaging. (a) Ultrasound. (b) A-rays. (c) Magnetic resonance. (d) Computed tomography.

4.3.1. Classification and prediction

A medical research wants to analyze breast cancer data in order to predict which one of three specific treatments a patient should receive. In the example, the data analysis task is classification, where a model o classifier is constructed to predict categorical labels, such as treatment A, treatment B, or treatment C for the medical data. These categories can be represented by discrete values, for example, the values 1, 2, and 3 may be used to represent treatment A, B, and C.

The implementation methods discussed are particularly oriented toward show different tools for analyzes medical data.

4.3.2. Classification by decision tree induction

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flow-chart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each brand represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node. The learning of decision trees from class-labeled training tuples is named decision tree induction. A decision tree can be viewed as a flow-chart-like tree structure, where each internal node (nonleaf node) represents a test on an attribute, each brand represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The root node is the principal node (highest node) in a tree.A typical decision tree is shown below (Figure 6).

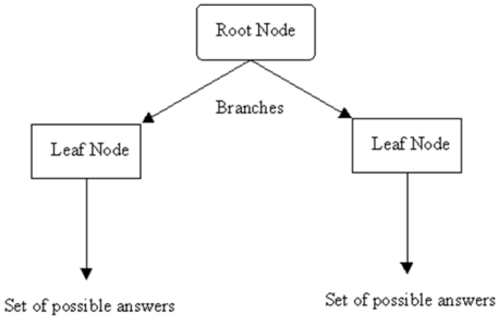


Figure 6. Decision Tree Induction

ID3 is an algorithm which generates a decision tree based on input data by looking at the amount of information contents contained in the various input attributes. At each step in the decision tree, it chooses the attribute which provides the biggest information gain and uses that attribute to classify data further. Its pseudo code is summarized as follows:

```
Input: Set S of positive and negative examples, Set F of features
ID3( F, S )
1. if S contains only positive examples, return “yes”
2. if S contains only negative examples, return “no”
3. else
    choose best feature f in F which maximizes the information gain
    for each value v of f do
    add arc to tree with label v, along with the sub tree for that new branch
```

Like an example, the input and output variables and their domains are specified in the list below:

1. Input variables (from clinical observations):
    a. Extent (Size of Spreading): {E1, E2, E3, E4}
    b. Hypoxia: {H1, H2}
    c. Surface (surface marker): {S1, S2, S3}
    d. LOH: {M1, M2, M3}
2. Final result/outcome:

    Outcome: {P (progressed to cancer), NP (didn’t progress to cancer)}

The ID3 algorithm as implemented and the following decision tree are generated (Figure 7):

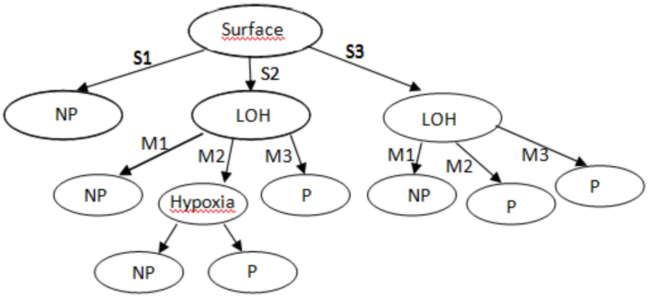


Figure 7. Decision Tree Induction for the medical data.

The decision tree method produces a reasonably good estimate on the outcome based on the inputs. This estimate is about 92% accurate, which is above the acceptable level of accuracy as proposed by the clinical researchers.

4.3.3. Classification by Back-propagation

An artificial neural network (ANN) is a computational model that is inspired by the structure and functional aspects of biological neural networks. They are usually used to model complex relationships between inputs and outputs and find patterns in data. In other words, we wish to infer the mapping implied by the data. The cost function is related to the mismatch between our mapping and the desired outcome. One very commonly used approach to train neural network from input examples is the back-propagation algorithm. Back-propagation algorithm is a common supervised-learning method that teaches an artificial neural network on how to perform a given task. The neural network is modeled as a set of neurons which take inputs, apply certain weights to each input and propagate the result forward into the next layer of units. Each unit in a particular layer is essentially a linear function of the input units from its previous layer. Eventually, the data gets propagated into the output layer where the results are presented.

Another important aspect is this algorithm is able to learn by propagating the errors in the output layer backwards into the inner layers by adjusting the weights between the input and hidden layer and between hidden and output layer in order to reduce the error on the output. The algorithm continues to do this until either the maximum number of epochs is reached or the errors at the output are within an acceptable range. This technique is also referred to as "back-propagation", as denoted in its name. A very typical neural network consists of 3 layers – input, hidden, and output layer. In practice, it is possible to have more than one hidden layers. The back-propagation algorithm used for this project is based on such a 3-layer neural network as illustrated in the figure 8.

The pseudo code for the back-propagation algorithm is as follows:

```
Initialize the weights in the network (randomly between -0.5 and 0.5)
Do
For each example e in the training set
```

```
O = neural-network-output(network, e) ; forward pass
T = desired output for e
Calculate error (T - O) at the output units
Compute delta_wh for all weights from hidden layer to output layer
Compute delta_wi for all weights from input layer to hidden layer
Update the weights in the network to reduce error
Until all examples classified correctly or stopping criterion satisfied
Return the network
```

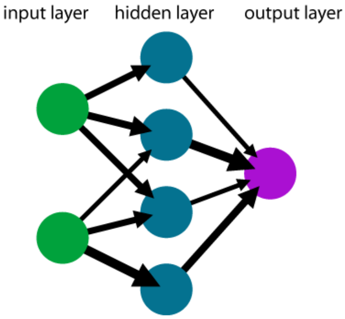


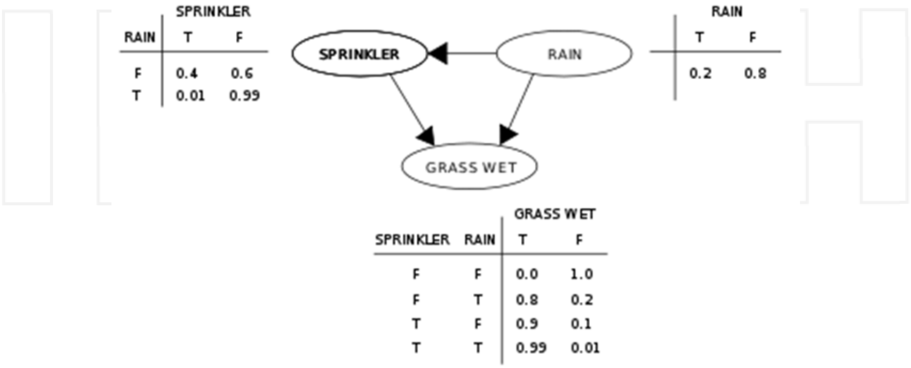
Figure 8. A simple neural network.

The output from the neural network is a simple binary value {0, 1} representing whether or not the patient's tumor progresses into malignant cancer. the classification boundary value to be the half-way point 0.5, so if the neural network's output value turns out to be above 0.5, it is categorized as 1; and values below 0.5 gets categorized as 0. The next important step is to determine the appropriate number of hidden variables in the neural network to avoid both under-fitting and over-fitting. The number of hidden variables should be strictly less than the number of inputs to the neural network, which is 4 in this case.

4.3.4. Classification by Bayesian networks

The naïve Bayesian classifier makes the assumption of class conditional independence, that is, given the class label of tuple, the value of the attributes are assumed to be conditionally independent of one other. This simplifies computation. When the assumption holds true, then the naïve Bayesian classifier is the most accurate in comparison with all other classifiers. However, dependencies can exist between variables. Bayesian networks specify joint conditional probability distributions. They allow class conditional independencies to be defined between subsets of variables. They provide a graphical model of causal relationships, on which learning can be performed. The learning can be perfomed in the graphical model of causal relationships, that they provide. Trained Bayesian belief networks can be used for classification.

A belief networks is defined by two components –a directed acyclic graph and a set of conditional probability tables (e.g., Figure 9). Each node in the directed acyclic graph represents a random variable. The variables may b discrete o continuous-valued. They may correspond to actual attributes given in the data to form a relationship (e.g., in the case of medical data, a hidden variable may indicate a syndrome, representing a number of symptoms that, together, characterize a specific disease). Each arc represents a probabilistic dependence. If an arc is drawn from a node *Y* to a node *Z*, then *Y* is a parent or immediate predecessor of *Z*, *Z* is a descendant of *Y*. Each variable is conditionally independent of its no descendants in the graph, given its parents, as is possible see in Figure 9.



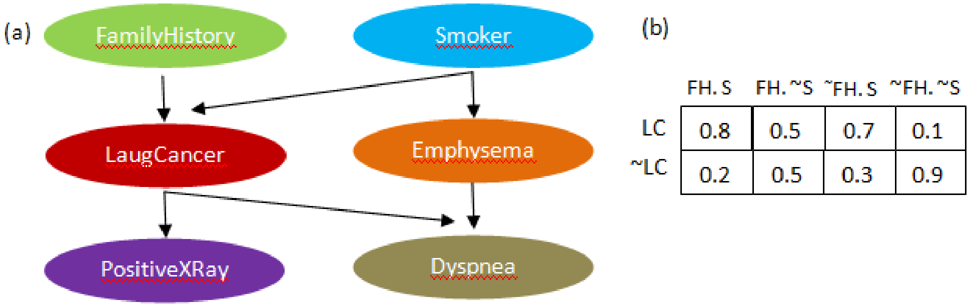
	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Figure 9. A example of Bayesian Network.

The Figure 10 is a simple Bayesian network for six Boolean variables. The arcs in figure 10 (a) allow the representation of causal knowledge. For example, having lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a smoker. The arcs also show that the variable *LungCancer* is conditionally independent of *Emphysema*, given its parents, *FamilyHistory* and *Smoker*.



	FH. S	FH. ~S	~FH. S	~FH. ~S
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

Figure 10. A simple Bayesian network: (a) A proposed casual model, represented by a acyclic graph. (b) The conditional probability table for the value of the variable LungCancer (LC) showing each possible combination of the values of its parents nodes, *FamilyHistory* (*FH*) and *Smoker* (*S*).

A belief network has one conditional probability table (CPT) for each variable. The CPT for a variable Y specifies the conditional distribution *P(Y|Parents(Y))*, where *parents(Y)* are the parents of *Y*. Figure 6 (b) shows a CPT for the variable *LungCancer*. The conditional probability for each knows value of *LungCancer* is given for each possible combination of values of its parents. For instance, form the upper leftmost and bottom rightmost entries, respectively, we see that

P(LungCancer = yes | FamilyHistory = yes, Smoker = yes) = 0.8

P(LungCancer = no | FamilyHistory = no, Smoker = no) = 0.9

A node within the network can be selected as an "output" node, representing a class label attribute. There may be more than one output node. Various algorithms for learning can be applied to the network. Rather than returning a single class label, the classification process can return a probability distribution that gives the probability for each class.

4.3.5. Visual data mining

Visual data mining discovers implicit and useful knowledge from large data Visual data mining have the capacity to find implicit and useful knowledge from great amount of data sets using data and/or knowledge visualization techniques. The human visual system is controlled by the eyes and brain, the latter of which can be thought of as a powerful highly parallel processing and reasoning engine containing a large knowledge base (Figure 11). Visual data mining essentially combines the power of these components, making it a highly attractive and effective tool for the comprehension of data distribution, patterns, clusters, and outliers in data. the eyes and brain, the latter of which can be thought of as a great highly parallel processing and reasoning engine that contain a large knowledge base (Figure 11). Visual data mining combines the power of these components, making it a highly attractive and effective tool for the comprehension of data patterns, clusters, distribution and outliers in data.

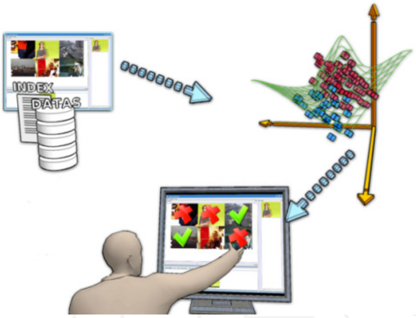


Figure 11. Human interact and processing large knowledge base.

Visual data mining can be viewed as an integration of two disciplines: data visualization and data mining. It is also closely related to computers graphics, multimedia systems, human computer interaction, pattern recognition, and high-performance computing. In general, data visualization and data mining can be integrated in the following ways:

Visual data mining can be viewed as an integration of two disciplines: data visualization and data mining. It is also closely related some disciplines: human computer interaction, pattern recognition, high-performance computing, computers graphics and multimedia systems. Data mining and data visualization can be integrated in the next ways:

- Data visualization: Data in a database or data warehouse can be view at different levels of granularity of abstraction, or as different combination of attributes or dimensions. Data can be presented in various visuals forms, such a boxplot, 3-D cubes, data distribution charts, curves, surfaces, link graphs, and so on. An example represented below:
  Data visualization: Data in a database or data warehouse can be view at different levels of granularity of abstraction, or as different combination of attributes or dimensions. Data can be presented in various visuals forms, such a data distribution charts, boxplot, curves, 3-D cubes, link graphs, surfaces, and so on. An example represented below:

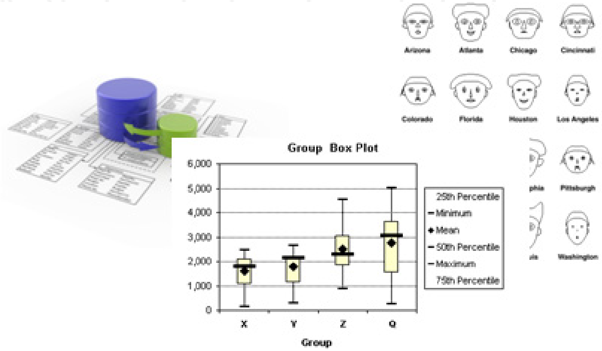


Figure 12. Boxplots showing multiple variable combinations in datasets.

- Data mining result visualization: Visualization of data mining results is the presentation of results or knowledge obtained from data mining in visual forms. Such forms may include scatter plots and boxplots, as well as decision tree, clusters, outliers, generalized rules and so on (Figure 9).
  Data mining result visualization: It means use techniques with which is possible the visual representation of results or knowledge that is obtained from data mining process. Such vicual forms may include scatter plots and boxplots, decision tree, clusters, outliers, generalized rules and so on (Figure 13).

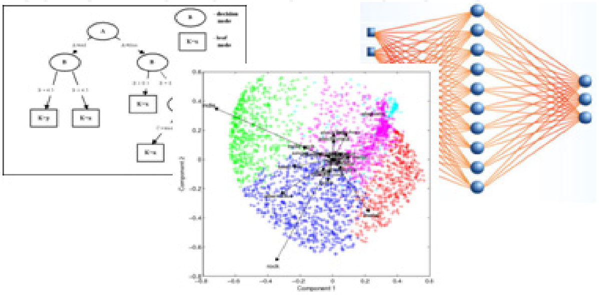


Figure 13. Visualization on data mining results.

- Interactive visual data mining: In visual data mining, visualization tools can be used in the data mining process to help users make smart data mining decisions. For example, the data distribution in a set of attributes can be displayed using colored sectors (where the whole space is representing by a circle). This display helps users determine which sector should first be selected for classification and where a good split point for this sector may be.

  The data mining process can be supported by visualization tools to help users to make smart data mining decisions. For example, in a circle that represents a whole space, the data distribution in a set of attributes can be displayed using colored sectors. With this visual representation the users can determine which sector should first be selected for classification and where a good split point for this sector may be.

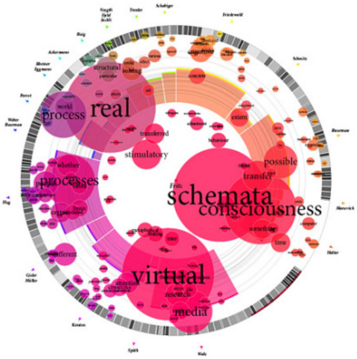


Figure 14. Example for circular data representation.

5. Analyzing people profile

The concept also is used to describe to the set of the characteristics that characterize to somebody or something. In the case of the human beings, the profile is associate to the personality. On the other hand, the word profile also is used very many to designate those particular characteristics that characterize a person and by all means they serve to him to be different itself from others. Your profile is built on other people's impressions and opinions, from the first time they hear your group's name or come into contact with one of its members. To some extent, you can control what people think and feel about your group, building a strong profile that will help you achieve action success. See Figure 15.

Orkut is a system of social networks used in Brazil by 13 million users, many of them, create more of a profile, and generate different relationships from their different profiles, this takes to think that they develop Bipolar Syndrome, to be able to establish communications with people of different life styles, and when they doing to believe other users that they are different people (Zolezzi-Hatsukimi, 2007).

The false profiles are created for: to make a joke, to harass other users, or to see who visualizes its profile. As the profile is false, the friends of this profile are also generally false, making difficult the tracking of the original author (Ochoa et. al., 2011). Using the tool of Data Mining denominated WEKA, it was come to develop a denominated "Ahankara" Model which perits reaize prediction of profiles in users of Orkut, which al-lows to understand the motivations of this type of profile and to determine if it has generated Syndrome Bipolar, to see figure 3 (Ponce et al., 2009). The model obtained Ahankara once used WEKA to look for the relations that us could be of utility to process the data. see Figure 16.

Figure 15. The profile show characteristics of a person or a group of people

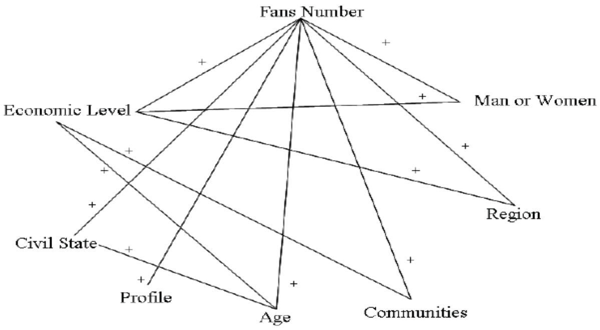


Figure 16. Ahankara Model

Waste. It is something that we produce as part of everyday living, but we do not normally think too much about our waste. Actually many cities generates a waste stream of great complexity, toxicity, and volume (see figure 17). In the management of solid waste have the problem relates to the household waste is the individual decision-making over waste generation and disposal. When the people decide how much to consume and what to consume, they do not take into account how much waste they produce (Ochoa et al., 2011).

E-commerce is the term use to describe the consumers that use the Internet for making purchases, usually refers typically to business to business type activities rather than consumer activity. It maybe more appropriate to refer to consumer activity in relation to purchasing goods and services on the Internet as on-line shopping (see figure 18). E-commerce is the term use to describe the consumers that use the Internet for making purchases, generally refers activities thath involve some business between two or more entities, rather than only consumer activity. This is more related to purchasign goods and services on the Internet rather than on-line shopping (see figure 18).

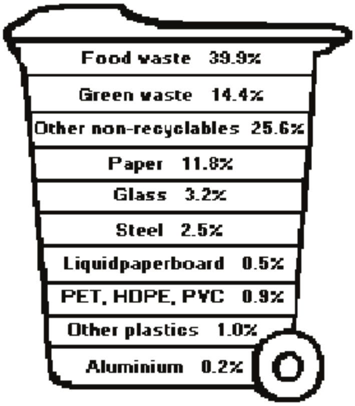


Figure 17. Example of composition by weight of household garbage

One of the important factors in the world of E-commerce is that it is much more than just a change in the way payments are made; E-commerce may not involve money at all. It gives customers the choice of making a wide range of transactions electronically rather than over the telephone, by post or in person. The E-commerce is not only a different way that the people use to pay for any thing, because, it is not simply money; this implies transactions that could be done by telephone, by post or in person, which the costumer can done electronically.

Figure 18. The people can buy services or things online

The major benefits of E-commerce are that it can help organizations to:

- improve working processes and service delivery;
- understand their customers better; and
- reduce costs through elimination of paperwork and bureaucracy.

Some of the most important benefits of E-commerce for the organizations are:

- The service delivery and the working process are improved.
- The organizations can understand their customers.

- They can reduce the costs caused by paperwork and bureaucracy.

In the e-commerce we has two different profile, the buyers and salesman profile, in this case we work with the buyers profile, In (Cocktail Analysis and Google, 2011) is sow a research about the buyers of fashionable clothes, in this work we can see that 42% of the people they have bought some article of clothes by Internet. They describe five different profiles only for the buyers of clothes, also it shows the relation of the purchases online with those of the physical stores. Data mining process can be used to determine the buyers profile, the enterprise can use this information to realize market studies in order to offer to the people specific products to them on the basis of its profile of purchases. Also the analysis of profiles is very important like dominion application of the data mining, can help to determine landlords us of conducts, habits, or of a single person or of a group, these data allow us to make predictions and can be used of diverse ways.

6. Data mining for E-comerce

The e-commerce is one of the profound changes that internet has induced in the people's lifestyle and in the way of doing business and transactions. The way that the consumers buy has been modified, appearing trends, patterns and preferences in specific groups. Some characteristics that can affect the consumerism by internet are: gender, age, social status, economic status, financial status, studies, culture, technology, knowledge of technology, geographic location, politics and others. In the early years of e-commerce, buying online was an erudite activity strictly dominated by "techies" and semi-technology literate individuals. These individuals were mostly made up of 20 to 35 year old males. This demographic were more comfortable and in tune with Internet's capabilities. But in recent years, the numbers of females making the technology leap to shop online is surging. Females are starting to harness Internet to make their lives easier and efficient (Christopher, 2004) . In the early years of e-commerce, buying online was an erudite activity. The individuals were mostly made up of 20 to 35 year old males. In recent years, the numbers of females making the technology leap to shop online is surging. (Christopher, 2004) . Data Mining (DM) has been applied successfully to find the patterns that the consumers create in the navigation trough the different web sites giving the opportunity to the enterprises to offer a better service.

6.1. Trends in E-commerce

In the e-commerce, the behavior of the consumers creates trends that change in the time for different variables. (Audette, 2010), mention three important trends in 2010 that should be considered by the people involved in the e-commerce (brands, retailers, and others).

6.1.1. Consumer focus is on price

The consumer always is looking for the lowest prices, it means, the best product for the best price or sometimes only the best price.

The consumer also looks for special offers that can balance that price was not the lowest. The offers can be the free shipping.

6.1.2. Riding the next wave: Video and visual search

It's very important now a day, the visual experience in the e-commerce because it is more attractive for the consumer and can be a reason to decide to buy something. The people spend a lot of time watching videos. A case is Mexico where the viewers watched 5 hours of video in YouTube in September 2011, and the audience has grown 17% to reach 20.5 million viewers, representing 85% of the total online population, according to a study by comScore. The next graphic shows video properties that prefer the viewers in Mexico.

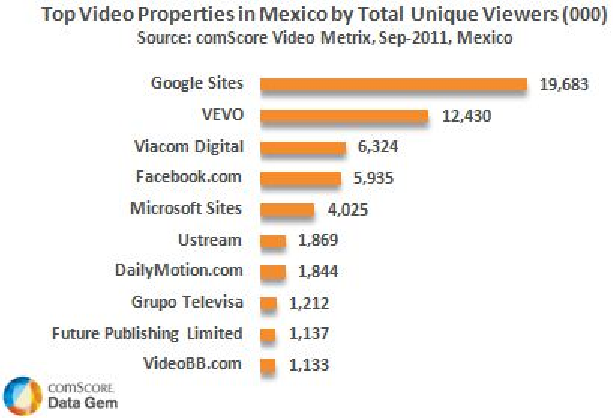


Figure 19. Top Video Properties in Mexico by Total Unique Viewers 2011

The video experience can improve the process information in a 30%, according Bing, and this can be explained because 65% are visual learners.

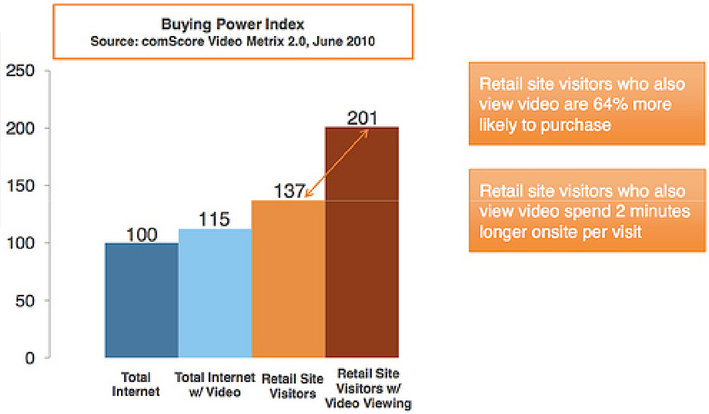


Figure 20. Buying Power Index 2010

6.1.3. Trend in technical SEO

SEO (Search Engine Optimization) is a technique which helps search engines find and rank your site higher than the others millions in response to a search query. This is based primarily in text. Google Instant had a little noticeable effect in the ecommerce clients.

6.2. Data mining and E-commerce

Data Mining (DM) have been applied to study the behavior of the users of different services (entertainment, mail, e-commerce, social network, among others) that internet provides. Many enterprises like Amazon and eBay have invested many resources to understand the consumers. Authors like (Sankar et al., 2002) explain why Web Mining, concept used for the first time by (Etzioni, 1996), is considered like sub-field of Data Mining. They say that Web Mining can be defined as "the discovery and analysis of useful information from the World Wide Web". The source of data can be the server, client, proxy server, or data bases of some enterprise. The web mining is divided in: Web content mining, Web structure mining, Web usage mining (Sankar et al., 2002). The principal tasks/phases of Web mining are: Information retrieval (resource discovery), information extraction (selection/preprocessing), Generalization (pattern recognition/machine learning), Analysis (validation/interpretation).

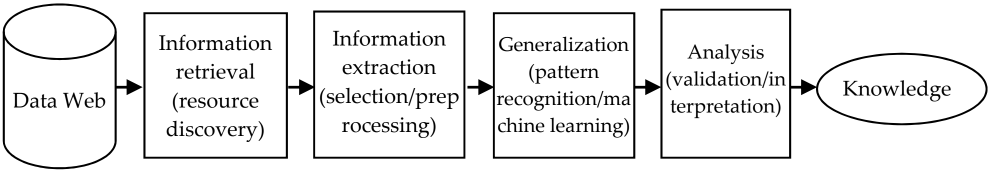

Data Web
Information retrieval (resource discovery)
Information extraction (selection/preprocessing)
Generalization (pattern recognition/machine learning)
Analysis (validation/interpretation)
Knowledge

Figure 21. Tasks of Web Mining

The Data Mining, or in this case Web mining, which is known, needs some problems with certain characteristics to obtain the major benefits. Those characteristics are (Ansari, Suhail, 2000):

- Large amount of data
- Rich data with many attributes
- Clean data collection
- Actionable domain
- Measurable return-on-investment

The e-commerce has every characteristics being a "Killer Domain" of Data Mining (Ansari, Suhail, 2000). The attributes more important in the e-commerce are RFM (Recency, Frequency and Monetary). Examples of these attributes are date, time, duration session, quantity, purchase (Ansari, Suhail, 2000). Other attributes are IP address, URL, error code, among others; however these are common in logs that are not created for analysis (De Gyves Camacho, 2009). The attributes (columns) related with time and date are used to find important hidden patterns. One of the applications of Web mining is the learning of Navigation patterns (Web usage Mining).

6.3. Artificial immune system applied to web mining

The Artificial Immune System (AIS) is a bio-heuristic based in the Natural Immune System (NIS). One of the characteristics that make interesting the NIS are: highly distributed, highly adaptive, self-organising, maintain a memory of past encounters, and learn of new encounters. Some algorithms that have been proposed to use in data mining, are based in theories like negative selection, clonal selection and immune network. However new algorithms have been created inspired in other characteristic or theories. In order to approximate a solution of the learning of navigation patterns an immune-inspired algorithm is proposed which is based in the immune network and was developed by (Timmis et al., 2000). The AIS has some characteristics that can be improved but is good for a first approximation. This algorithm is proposed to clustering the similarities of the users' behavior and according of the pattern, in a next step, suggest the best structure of the web site to the consumers to improve their experience. In this way, the companies con offer a better service, adapted to the consumers' necessities and finally increase sales.

7. Data mining to mobile ad hoc networks security

Mobile radio technologies, for both voice and data communication, has experienced a rapid growth and diverse concepts have been introduced in networking. However the concept of ad hoc network is not new, the paradigm started from the beginning of late 90's and gradually became popular with the wide range of deployments of IEEE 802.11x based WLAN, despite regularly ad hoc networks are based on single-hop peer-to-peer networking between several wireless devices, in different specialized scenarios such as control applications, logistics and automation, surveillance and security, transportation management, battlefields, environmental monitoring, unexplored and hazardous conditions, home networking, etc. multi-hop wireless networks are used. Multi-hop wireless ad hoc network consists of a number of self-configurable nodes (e.g. IEEE 802.11-based WLAN, 802.16-based WiMAX, ZigBee, Bluetooth, etc.) to establish an on-demand network using multiple hops paths if required where no network infrastructures pre-exist. The basic block of multi-hop ad hoc networking can be divided into four major specialized categories – Mobile Ad hoc Networks (MANET), Wireless Mesh and Hybrid Networks (WMN), Vehicular Ad hoc Networks (VANET) and Wireless Sensor Networks (WSN) (Kamal, 2010). MANET is the most theoretically researched arena of ad hoc networking which is a collection of autonomous and mobile network objects of any kind with truly dynamic and uncertain mobility that communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. Nowadays, MANET has become a practical platform for pervasive services, i.e., the services that are requested and provided anywhere and anytime in an instant way. This kind of service is very valuable for mobile users, especially when fixed networks (e.g. Internet) or mobile networks are temporarily unavailable or costly to access. A generic concept of the general-purpose pure MANET is shown in Figure 22.

In Figure 22, let's suppose that node A wants to send data to node C but node C is not in the range of node A. Then in this case, node A may use the services of node B to transfer data since node B's range overlaps with both the node A and node B. In MANET, no fixed infrastructure, like base station or, mobile switching center is required. Instead, every possible wireless mobile host within the perimeter of radio link acts as an intermediate switch and participates in setting up the network topology in a self organized way.

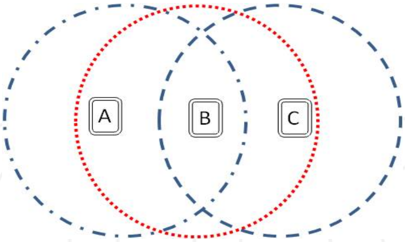


Figure 22. A Simple MANET

7.1. Data mining to deal with vulnerabilities of MANET

Despite the advantages, accord to Nakkeeran, the nature of mobility creates new vulnerabilities due to the open medium, dynamically changing network topology, cooperative algorithms, lack of centralized monitoring and management points and yet many of the proven security measures turn out to be ineffective (Nakkeeran et al. , 2010). Despite the advantages, accord to Nakkeeran, the nature of mobility creates new vulnerabilities due to the open medium, dynamically changing network topology, cooperative algorithms, lack of centralized monitoring and management points and yet many of the proven security measures turn out to be ineffective (Nakkeeran et al. , 2010). All these mean that a wireless ad-hoc network will not have a clear line of defense, and every node must be prepared for encounters with an adversary directly or indirectly. In order to avoid such circumstances requires the development of novel architectures and mechanisms that protect wireless networks and computer applications. Hence diverse research scopes do exist. Unfortunately, investigations are principally targeted towards routing, scheduling, address assignment, developing protocol stack etc. These are mainly functional properties. However, as nomadic and ubiquitous computing reaches its full potential, semantics and security will play the leading role, because the flexibility in space and time induces new challenges towards the security infrastructure. Due to in the case of the securtiy infraestructure, the flexibility in space and time will generate new challenges. Therefore, the traditional way of protecting wired/wireless networks with firewalls and encryption software is no longer sufficient. A very recurrent solution are Intrusion Detection Systems (IDS) (Mishra, 2004). Generally IDS can be defined as the detection of intrusions or intrusions attempts either manually or via software, through the use of schemes that collects the information and analyzing it for uncommon or unexpected events. Intrusion Detection (ID) is the process of monitoring and analyzing the events which occurred in a digital network in order to detect signs of security problems (Shirbhate, 2011). Then the ID is data analysis process, for this reason, as well as the growth of volume of existing data and insufficiency of data storage capacity leads us to the dynamic processing data and extracting knowledge. So the nature solution is utilizing data mining techniques, for example, anomaly detection techniques could be used to detect unusual patterns and behaviors, link analysis may be used to trace the viruses to the perpetrators, classification may be used to group various cyber attacks and then use the profiles to detect an attack when it occurs, prediction may be used to determine potential future attacks depending in a way on information learnt about terrorists through email and phone conversations (Khalilian, 2011). Data mining can improve variant detection rate, control false alarm rate and reduce false dismissals (Jianliang, 2009).

7.2. Intrusion detection methodologies

If we want to categorize intrusion detection methods, we will recognize two main aspects for grouping approaches, which one group refers to type of attack according to the kind of input information the analyze includes host based, network based, wireless and Network Behavior Analysis (NBA). Another group of approaches refers to solutions techniques which are misuse detection, anomaly detection methods and hybrid methods (Khalilian, 2011).

a. Host based methods.
This methods are based on data source category; consequently, its data comes from the records of various activities of hosts, including system logs, audit operation system information, etc. the main architecture for this kind of methods is similar to network based.

b. Network based methods.
These systems analyze network packets that are captured on a network. Network packet is the data source for network intrusion detection system.

c. Wireless methods.
Wireless intrusion detection system monitors wireless network traffic and analyzes its wireless networking protocols to identify suspicious activity involving the protocols themselves. It cannot identify suspicious activity in the application or higher-layer network protocols such as TCP, UDP that the wireless network traffic is transferring. So each node is responsible for detecting signs of intrusion locally and independently, but neighboring nodes can collaboratively investigate in a broader range.

d. Network Behavior Analysis.
NBA which examines network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDoS) attacks, certain forms of malware such as worms, backdoors, and policy violations. NBA systems are also deployed to monitor flows on an organization's internal Networks, and are also sometimes deployed where they can monitor flows between an organization's Networks and external networks such as the Internet.

Figure 23 shows the basic architecture for NIDS in data mining, which is very similar to the other detection methods

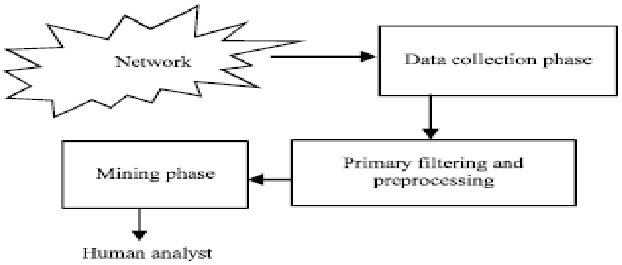


Figure 23. Basic NIDS architecture

e. Misuse based methods.

Misuse detection which the main study is the classification algorithms relies on the use of specifically known patterns of unauthorized behavior. In misuse detection related problems, standard data mining techniques are not applicable due to several specific details that include dealing with skewed class distribution, learning from data streams and labeling network connections. The problem of skewed class distribution in the network intrusion detection is very apparent since intrusion as a class of interest is much smaller i.e. rarer than the class representing normal network behavior. In such scenarios when the normal behavior may typically represent 98-99% of the entire population a trivial classifier that labels everything with the majority class can achieve 98-99% accuracy (Dokas, 2002). It is apparent that in this case classification accuracy is not sufficient as a standard performance measure. ROC analysis and metrics such as precision, recall and F-value have been used to understand the performance of the learning algorithm on the minority class. A confusion matrix as shown in Table 1 is typically used to evaluate performance of a machine learning algorithm.

Confusion matrix (Standard metrics)		Predicted connection label	
		Normal	Intrusions (Attacks)
Actual connection label	Normal	True Negative (TN)	False Alarm (False Positive)
	Intrusions (Attacks)	False Negative (FN)	Correctly Detected Attacks (True Positive)

Table 10. Standards metrics for evaluations of intrusions (attacks)

In addition, intrusions very often represent sequence of events and therefore are more suitable to be addressed by some temporal data mining algorithms. Finally, misuse detection algorithms require all data to be labeled, but labeling network connections as normal or intrusive re-quires enormous amount of time for many human experts. All these issues cause building misuse detection models very complex.

f. Anomaly based methods.

Misuse detection system unable to detect new or previously unknown intrusions occurred in computer system or digital network. Novel intrusions can be found by anomaly detection which the main study is the pattern comparison and the cluster algorithms ((Khalilian, 2011). The basic idea of clustering analysis originates in the difference between intrusion and normal pattern; consequently, we can put data sets into different categories and detect intrusion by distinguish normal and abnormal behaviors. Clustering intrusion detection is detection for anomaly with no supervision, and it detects intrusion by training the unmarked data.

Most anomaly detection algorithms require a set of purely normal data to train the model, and they implicitly assume that anomalies can be treated as patterns not observed before. Since an outlier may be defined as a data point which is very different from the rest of the data, based on some measure. In statistics-based outlier detection techniques the data points are modeled using a stochastic distribution and points are determined to be outliers depending upon their relationship with this model. However, with increasing dimensionality, it becomes increasingly difficult and inaccurate to estimate the multidimensional distributions of the data points. However, recent outlier detection algorithms are based on computing the full dimensional distances of the points from one another as well as on computing the densities of local neighborhoods. Nearest Neighbor (NN), Mahalanobis-distance Based Outlier Detection and Density Based Local Outliers (LOF) are approaches for recent outlier detection algorithms.

g. Hybrid methods.

Through analyzing the advantages and disadvantages between anomaly detection and misuse detection, a mixed intrusion detection system (IDS) model is designed. First, data is examined by the misuse detection module, and then abnormal data detection is examined by anomaly detection module. The intrusion detection system (IDS) is designed based in the advantages and disadvantages of the models of intrusion detection that are: anomaly detection and misuse detection. The first step is examine the data with de misuse detection module and in the second step, the anomaly detection module analyze the atypical data detected.

7.3. New trends in safety MANET

The ultimate goal of the security solutions for wireless networks is to provide security services, such as authentication, confidentiality, integrity, anonymity, and availability, to mobile users. The final goal of the security solutions for wireless networks is to offer security services to mobile users. This services are authentication, integrity, confidentiality, anonymity, and availability.This kind of schemes depend on cooperation amongst the nodes in a MANET for identifying nodes that are exhibiting malicious behaviors such as packet dropping, packet modification, and packet misrouting, so most of this methods assume that this problem can be viewed as an instance of detecting nodes whose behavior is an outlier when compared to others. Some novel solutions incorporate mobile agents (Nakkeeran et al., 2010) to provide solution against security issues in MANET networks. With the help of home agent and mobile agents, it gathers information from its own system and neighboring system to identify any attack and through data mining techniques to find out the attacks has been made in that networks.With the help of home agent and mobile agents, it is possible to extract information from both, own system and neighbor system, to identify any attack and with data mining techniques try to find the attacks that has been perpetrated in such networks. Home agent is present in each system and it gathers information about its system from application layer to routing layer.

Each system have a home agent, which should obtain iformation about the system from application layer to routing layer.

Mobile agents are a special type of agents defined as "processes capable of roaming through large networks such as the ad hoc wireless network, interacting with machines, collecting information and returning after executing the tasks adjusted by the user". Mobile agents are a special type of agents defined as "processes capable of roaming through large networks such as the ad hoc wireless network, interacting with machines, which return the colected information when finishing the execution of the tasks of the user". Often such proposals provide the three different techniques to provide suffice security solution to current node, Neighboring Node and Global networks.

Frequently, the proposals afford the three differents techniques that are used to offer security solution to current node, Neighboring and Global networks.

The trust together cooperation are another kind of novelty solution (Li, 2009); the idea of them consists in an algorithm to can help us identify the outliers, which are generally the nodes that have exhibited some kind of abnormal behaviors. Given the fact that benign nodes rarely behave abnormally, it is highly likely that the outliers are malicious nodes. Moreover, a multi-dimensional trust management scheme is proposed to evaluate the trustworthiness of the nodes from multiple perspectives. There are many techniques that have been discussed to prevent attacks in wireless ad hoc networks but most of them have in common that are based on cooperation and on methods based on the principle of anomalies.

8. Conclusions and another specific application domains improved with data mining

Diverse applications based on Data Mining have the objective to learn the patterns of the users' interaction with the Web or Data Repository. The data includes user profiles, registration profiles, user queries, and any data generated by the users' interaction with the web. This is useful, for example, to restructure the web page according the preferences of the users. This means that the web site is going to provides information, special offers, and others, that can be interesting for the consumers according their patterns of interaction or according the hour of the day. Also, this can be used to design offline strategies. One process to make projects of web usage mining is described in (DAEDALUS, 2002). To find the patterns in Web usage mining, the techniques used are: Clustering and Classification, Association rule detection, Path analysis, Sequential patterns detection. The use of any technique mention above to analyze automatically the data implies difficulties by the complexity of the problem (heterogeneous data, and others) and the limitations of the existing methodologies. To overcome that difficulties and limitations is necessary to use other techniques and methodologies like soft computing. Some algorithms developed to address Data Mining using techniques of Soft Computing are revised by (Mishra el al., 2004). Other application of Data Mining using evolutionary algorithms were proposed by (Ochoa et al., 2011) obtaining good results. In addition, we described another specfic applications domains such as: Deterining Euskadi ancesters based on family names and compare the anthropmetry of the individuals to found patterns of their ancesters; Organizational Models to supporting little and medium business related with Regional Development; Organizational Climate to identify cases of Burnout Syndrome characterized by high expectatives of productivty and Organizational Culture (Hernández et al, 2011); Identification of the use of new languages related with the songs from Eurovision for exameple Udmurt language in the entry from Russia to Eurovision Song Contest'2012; Analysis of Pygmalion Effect on people from Pondichérry in India whom be considering more closely culturally of Francophonie because the French influence in their past lifes; Zoo applications to classify more vulnerable species in an interactive map (see figure 24) or identify the adequate kind of avatars on a roll multigame players associated with cultural aspects in this case Brazilian people and their selections of features related with spcific skills (see figure 25).

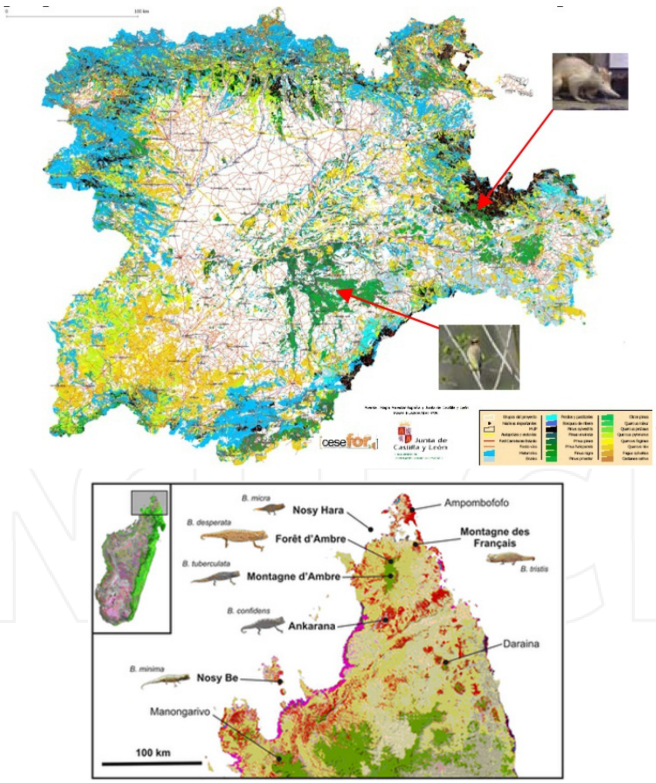


Figure 24. Interactive map based on data mining, locating the habitats of species of reptils, birds and mammals specifying the ubiquity of their behavior –changes provoked by the human- during the time.

Figure 25. Cultural Avatars related with traditional aspects and antropometry from Brazlian people used in Multi player games according of specific skills used on the online game.

In addition Tatebanko traditional Japanese Dyoram is using new ideas based with a Hybrid Algorithm conformed by the use of Data Mining and a Bioinspired Algorithm to built 3D scenario (Ochoa et al., 2012) including issues of specific time and location by each one.

Author details

Alberto Ochoa, Daniel Azpeitia, Petra Salazar, Emmanuel García and Miguel Maldonado
Juarez City University, México

Rubén Jaramillo and Jöns Sánchez
LAPEM, México

Javier González and Claudia Gómez
ITCM, México

Julio Ponce, Sayuri Quezada, Francisco Ornelas and Arturo Elías
UAA, México

Edgar Conde and Víctor Cruz
Veracruzana University, México

Lourdes Margain
Universidad Politécnica de Aguascalientes, México

9. References

Ansari, Suhail; Kohavi, Ron; Mason, Llew and Zheng, Zijian. Integrating E-Commerce and Data Mining: Architecture and Challenges. WEBKDD’2000 workshop: Web Mining for E-Commerce -- Challenges and Opportunities, 2000.

Audette Adam. Founder and Presidente of AudetteMedia., Nov 29, 2010 http://searchengineland.com/3-important-trends-to-watch-in-ecommerce-56890

Cocktail Analysis and Google (2011). El comportamiento del Comprador de Moda OnLine. http://tcanalysis.com/

Cheng, Ri; Kai, L.; Chun, B.; Shao-Yu, D.; Gou-Zheng, X. Study on Partial Discharge Localization by Ultrasonic Measuring in Power Transformer Based on Particle Swarm Optimization. *International Conference on High Voltage Engineering and Application*. (2008). 600-603.

Christopher, James. E-Commerce: Comparison of On-line Shopping Trends, Patterns and Preferences against a Selected Survey of Women. Kingston University. MSC Business Information Technology Program. November 2004.

DAEDALUS – Data, Decisions and Language, S.A.: Minería Web: Documentos básico DAEDALUS. White Paper, C-26-AB-6002-010, Noviembre 2002. http://www.daedalus.es

De Gyves Camacho, Francisco Manuel. Web Mining: Fundamentos Básicos Doctorado en informática y automática Universidad de Salamanca. Informe Técnico, DPTOIA-IT-2006-003. Mayo 2009.

Dokas, P., et al. *Data mining for network intrusion detection*. in *In Proceedings of the NSF Workshop on Next Generation Data Mining*. 2002. Baltimore, MA.

Etzioni, O. The world-wide web: Quagmire or goldmine?, Communications of the ACM, vol. 39, pp. 65-68, 1996.

Hernández, Alberto et al. Aplicación de la minería de datos para la toma de decisiones: El Caso de la cultura organizacional en una tienda del IMSS, XVI Congreso Internacional de Contaduría, Administración e Informática, 2011.

Jianliang, M., S. Haikun, and B. Ling. *The Application on Intrusion Detection Based on K-means Cluster Algorithm*. in *Information Technology and Applications, 2009. IFITA '09. International Forum on* 2009. Chengdu IEEE, Press.

Kamal, J.M.M., *A Comprehensive Study on Multi-Hop Ad hoc Networking and Applications: MANET and VANET*, in *Faculty of Computing, Engineering and Technology*. 2010, Staffordshire University: Stafford. p. 155.

Khalilian, M., et al., *Intrusion Detection System with Data Mining Approach: A Review.* Global Journal of Computer Science and Technology (GJCST), 2011. 11(5): p. 29-34.

Kohonen T. Engineering Applications of Self Organizing Map. *Proceedings of the IEEE*. (1996).

Li, W., J. Parker, and A. Joshi, *Security through Collaboration in MANETs Collaborative Computing: Networking, Applications and Worksharing*, E. Bertino and J.B.D. Joshi, Editors. 2009, Springer Berlin Heidelberg. p. 696-714.

Mishra, A., K. Nadkarni, and A. Patcha, *Intrusion detection in wireless ad hoc networks* Wireless Communications, IEEE 2004. 11(1): p. 48-60.

Nakkeeran, R., T. Aruldoss Albert , and R. Ezumalai, *Agent Based Efficient Anomaly Intrusion Detection System in Adhoc networks.* IACSIT International Journal of Engineering and Technology, 2010. 2(1): p. 52-56.

Ochoa, Alberto; Castillo, Nemesio; Yeongene, Tasha & Bustillos, Sandra. Logistics using a new Paradigm: Cultural Algorithms. Programación Matemática y Software, Vol. 1. No 1. Dirección de Reservas de Derecho: 04-2009-011611475800-102. 2011.

Ochoa, Alberto et al. New Implementations of Data Mining in a Plethora of Human Activities. In Knowledge-Oriented Applications in Data Mining, ISBN 978-953-307-154-1, 2011.

Ochoa, Alberto et al. Developing a Traditional Tatebanko Dyoram using Cultural Algorithms V Workshop Hybrid Intelligent Systems at MICAI'2012 to publish.

Orihuela, José Luis. Nuevos Paradigmas de la Comunicación. Retrieved March. Vol. 12, España (2002).

Ponce, Julio; Hernández, Alberto; Ochoa, Alerto et al. Data Mining in Web Applications. In Data Mining and Knowledge Discovery in Real Life Applications, ISBN 978-3-902613-53-0, 2009.

Rubio-Sánchez, M. *Nuevos Métodos para Análisis Visual de Mapas Auto-organizativos*. PhD Thesis. Madrid Politechnic University. (2004).

Salaveria, Ramón (2009). El Impacto de Internet en los Medios de Comunicación en España. Comunicación Social Ediciones y Publicaciones. Pp. 11-15, 2009.

Sandoval, Rodrigo, Saucedo Nancy Karina. Grupos de Interés en las Redes Sociales: El caso de Hi 5 y Facebook en México. Tecnociencia Chihuahua. Vol. IV, No. 3, 2010.

Sankar K. Pak, Varun Talwar, Pabitra Mitra. Web Mining in Soft Computing FrameWork: Relevance, State of the Art and Future Directions. IEEE Transactions on Neural Networks Vol. 13, No. 5 pp 1163-1177, September 2002.

Shirbhate, S.V., V.M. Thakare, and S.S. Sherekar, *Data Mining Approaches For Network Intrusion Detection System.* International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2011. 2(2): p. 41-44.

Timmis, J, Neal, M and Hunt, J. An Artificial Immune System for Data Analysis.*Biosystems. 55(1/3)*, pp. 143-150. 2000.

Zolezzi-Hatsukimi, Z. Implement social nets using Orkut, Proceedings of CHI'07, Nagoya, Japan, 2007.

Handling of Synergy into an Algorithm for Project Portfolio Selection

Gilberto Rivera[1], Claudia G. Gómez[1], Eduardo R. Fernández[2], Laura Cruz[1], Oscar Castillo[3], and Samantha S. Bastiani[1]

[1] Madero Institute of Technology, México
riveragil@gmail.com, {cggs71,b_shulamith}@hotmail.com,
lcruzr@prodigy.net.mx
[2] Sinaloa Autonomous University, México
eddyf@uas.uasnet.mx
[3] Tijuana Institute of Technology, México
ocastillo@tectijuana.mx

Abstract. Public and private organizations continuously invest on projects. With a number of candidate projects bigger than those ones that can be funded, the organization faces the problem of selecting a portfolio of projects that maximizes the expected benefits. The selection is made on the evaluation of project groups and not on the evaluation of single projects. However, there is a factor that must be taken account, since it can significantly change the evaluation of groups: synergy. This is that two or more projects are complemented in a way that generates an additional benefit to they already own individually. Redundancy, a special case of synergy, occurs when two or more projects cannot be financed simultaneously. Both features add complexity to the evaluation of project groups. This article presents an evaluation of the two most used alternatives for handling synergy, in order to incorporate it into an ant-colony metaheuristic for solving project portfolio selection.

1 Introduction

One of the most important decisions in organizations is to select which projects will be funded for ensuring their growth [2]. This decision is made by one or several people, commonly called the *Decision Maker* (DM). In resources allocating for social projects, the monetary benefits are not the main criteria for selecting projects. Other objectives that measure the social profit are more important. DMs are responsible for selecting a portfolio depending on:

1. Values in the portfolio objectives which have to assure a minimal acceptability level.
2. His/her criteria that shall depend on several personal subjective issues, among which can be cited: his/her belief, experience, policies to follow according to the organization and personal ethics.

O. Castillo et al. (Eds.): Recent Advances on Hybrid Intelligent Systems, SCI 451, pp. 417–430.
springerlink.com

Typically the *Social Portfolio Problem* (SPP) is modeled as a multicriteria problem [3, 15, 20], and has been solved by means of algorithms that commonly search a Pareto frontier approximation. This guarantees a good level of acceptability (in terms of objective values) in the final solution set. However, the cardinal of this set is often too long; this complicates the selecting process made by the DM. A typical DM will have problems for processing more than 5-9 pieces of knowledge [19], so he/she is unable to select satisfactorily one portfolio even from a relatively-short set of solutions with more than five objectives. To ease the decision making process, the reducing the final solution set is critical. But, the reduced set presented to the DM has to match the preferences of him/her.

Three approaches are commonly used for identifying solutions that match the DM preferences [17]:

1. *Including the DM's preferences after optimization process*, searching a uniform Pareto frontier but presenting only the solutions that better match his/her preferences.
2. *Including the DM's preferences before optimization process*, guiding the search to privileged Pareto frontier zones that better match the DM's preferences.
3. *Interacting with the DM progressively during optimization process.*

In this paper we use the Fernández's model for identifying the DM's preferences about the portfolios. It is a priori articulation of preferences by creating fuzzy outranking relations that allow identifying which solutions match better the DM's preferences (these solutions are called the best compromise).

One feature commonly presents in SPP is synergy, which involves that there are interrelated groups of projects. Such interrelation provokes an additional benefit to organization objectives. So, when the whole synergetic group is supported the benefits are bigger than the sum of benefits of the same projects taken independently.

Basically, there exist two forms when the synergy may be neglected without major problems, when:

1. the interaction is too weak.
2. the interactions affect all the projects in a similar way and constantly.

But in other cases, when no previous characteristic is present, then synergy and redundancy among projects become relevant matters for decision making on project portfolio.

When the DM identifies important synergetic effects, finding an appropriate way for handling synergy and redundancy is not a simple task. Commonly, it is done by means of inclusion of an artificial project, that groups the synergetic set and whose benefits already include the synergy gain. But, extra redundancy relations must be included, between artificial project and each one of the original projects with synergy.

Although previous works on the formation of project portfolio with synergy have been proposed [3, 6, 18, 20], it has made evident the lack of methods that also include the decision maker's preferences. This work presents an Ant Colony

Optimization (ACO) Algorithm that searches a Pareto frontier subset that matches the DM's criteria for selecting a portfolio.

This paper is structured as follows: we formalize the problem in Section 2; in Section 3 we analyze the metaheuristic procedures used to solve it; Section 4 develops our proposal; Section 5 presents the instances and discusses the results; and final section offers the conclusions.

2 Background

In order to establish the theoretical elements needed to pose our contribution, we begin with a formal definition for social portfolio problem, afterward we shall focus on the elements related to the synergy among projects, types and issues inherent, and we conclude with a description of the preference model used in this proposal.

2.1 Social Portfolio Problem

From a set of all proposed projects competing for resources, denoted as X, a portfolio may be defined as a subset of them. Typically is modeled as a binary vector $x = \{x_1, x_2, x_3, \ldots, x_N\}$, where N is the total of project proposals and the variables xi indicate whether the project i is included in the portfolio (if $x_i = 1$) or not (if $x_i = 0$).

There is a total budget that the organization is willing to invest, which is denoted as B, and each project proposal i has an associated cost to carry it out, denoted as c_i, it is understood that the formation of any portfolio x is subject to the constraint:

$$\left(\sum_{i=1}^{N} x_i c_i \right) \leq B \tag{1}$$

It is also common that projects are associated with some form of grouping that influences the decision of the DM. For example in a private company, projects may be associated to departments (e.g. marketing, sales, production or human resources), in this case a balanced DM would try to ensure a minimum of the total budget for each, and would prevent either of them unjustifiably monopolizes most of the budget. For public projects, such groups can be associated with geographical divisions, social groups or public interest areas (e.g. social security, education or public health). In general terms, this form for grouping will be called "areas".

Let L_i and U_i be respectively the minimum and maximum budget that an area i can get ($L_i \leq U_i \leq B$). The area for a project i may be defined as a_i. For each area j, any portfolio x must satisfy the constraint:

$$L_i \leq \sum_{i=1}^{N} x_i g_{i,j} \leq L_u \tag{2}$$

where $g_{i;j}$ may be defined as follows:

$$g_{i,j} = \begin{cases} c_i & \text{if } a_i = j \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

Suitable values for L and U and the total of areas depend on the problem characteristics, the DM criteria and the organizational policies.

In private organizations is more natural to represent the benefits of projects in monetary units than in public or social organizations. There are also ethical and moral reasons preventing the realization of a monetary equivalent of objectives of public organizations, for example, can raise questions such as: what is the value of a person's health?, and for a child's education?, and for a young man's postgraduate studies?. Due to these reasons, the selection of social projects is done on the multiobjective evaluation of portfolios, and not on a single monetary equivalent of objectives.

So, the benefits for a project i evaluated on p objectives may be modeled as $f(i) = \{f_1(i), f_2(i), f_3(i), \ldots, f_p(i)\}$. And therefore, the quality of a portfolio x is expressed as

$$z(x) = \{z_1(x), z_2(x), z_3(x), \ldots\ldots, z_p(x)\} \tag{4}$$

where $z_j(x)$ is defined as

$$z_j(x) = \sum_{i}^{N} x_i f_j(i) \tag{5}$$

Social Portfolio Problem (SPP) consists in identifying one or more portfolios that solve

$$\arg\max_{x \in X} \{z(x)\}, \tag{6}$$

subject to the constraints expressed in Equations 1 and 2. In this case, the maximization concept is based on Pareto efficiency. In this case, the dimension of its solution space is 2^N, and if taken into account additional considerations such as synergy, partial support, project scheduling or risky conditions, the problem complexity and the number of possible solutions tend to increase.

2.2 Synergy Among Projects

Interactions between projects may affect the evaluation of portfolios. Among synergy types to be found are:

1. *Synergy positive on objective values.* When two or more projects are complemented for increasing profits. For example, a Project A: creating a recreational park, and a Project B: Paving a road. Each one individually benefits certain amount of people, but if the recreation park (project A) is on the road to be

paved (project B) both increase their value. The park becomes more accessible to people, and increase the amount of persons that use the road.

2. *Negative synergy on objective values:* The benefit of two or more projects decreases when are together. For example, a project A that benefits to 100 people and a Project B benefiting 200 people. But if there are 50 people in common between both projects, the benefit of supporting both projects is 250 people instead of 300 as expected.
3. *Redundancy:* Two projects can not be supported simultaneously. For example, a Project A: Building a hospital, and a Project B: building a school. But if the hospital needs to be built on the same ground that the school, only one of them can receive budget.
4. *Decrease in the cost:* Two projects decrease the costs if both are supported simultaneously. For example, a project A: has a total cost of 250 monetary units, of which uses 100 to buy expensive computer equipment, and a Project B: it has a cost of 200 monetary units and also need to use similar equipment. If possible, and both projects have no problem sharing the common resource, the cost of financing both resources is 350 monetary units instead of 450 as expected.

Addressing the synergy among projects is a task that is not considered trivial in the forming of social portfolios [14]. Handling of redundancy involves modifying the feasibility conditions of a portfolio. While the handling of synergy that changes the objective values has been handled typically in two forms:

1. Doing an adaptation in the multiobjective function (Equation 4).
2. Adding an artificial project that represents the synergetic set and adding all necessary redundancy relations to prevent the artificial project and any original synergetic projects in this set appear together.

Synergy have been addressed in the social portfolio problem [3, 6, 20], but few research works have also considered the use of a model of preferences [18].

2.3 Preference Model

Although finding Pareto front is important [4, 9], the problem does not is completely solved [5, 11, 12, 13, 23]. Now the DM will choose which portfolio will be selected to receive the budget. Presenting a large set of solutions will complicate the decision process of the DM, even if the solutions found belong to the Pareto front, moreover finding a single function that matches the DM's preference, such as a weighted sum, is not simple and nor guaranteed for real problems [21, 22].

Fernández [10] proposed an *a priori* articulation to guide the search process to areas of the Pareto front that best match the preferences of the DM. Thereby reducing the amount of solutions presented to the DM. The preference model is based on the value of $\sigma(x,y)$, which measures the degree of credibility of the statement "x is at least as good as y", where x and y are portfolios in SPP. To calculate the value of $\sigma(x,y)$ can be used several proven methods, including the ELECTRE [1] and PROMETHEE [8] methods.

Considering the parameters λ, β, and ε ($0 \le \varepsilon \le \beta \le \lambda$), the model identifies one of the following relationships for each pair of portfolios (x, y):

1. $x\mathbf{I}y$, *Indifference:* Corresponds to existence of reasons that justifies equivalence between x and y. $x\mathbf{I}y$ is justified if both conditions are held:
   a. $\sigma(x,y) \ge \lambda \wedge \sigma(y,x) \ge \lambda$.
   b. $|\sigma(x,y) - \sigma(y,x)| \le \varepsilon$.
2. $x\mathbf{P}y$, *Strict preference*: Represents the existence of reasons that justify significant preference in favor of x. The statement "x is strictly preferred to y" is hold if at least one of following sentences is true:
   a. x dominates y.
   b. $\sigma(x,y) \ge \lambda \wedge \sigma(y,x) < 0.5$.
   c. $\sigma(x,y) \ge \lambda \wedge (0.5 \le \sigma(y,x) \le \lambda) \wedge (\sigma(x,y) - \sigma(y,x)) \ge \beta$.
3. $x\mathbf{Q}y$, *Weak preference*: Corresponds to the existence of reasons in favor of x over y, but that are not sufficient to justify strict preference. $x\mathbf{Q}y$ is modeled by the conjunction of the following propositions:
   a. $\sigma(x,y) \ge \lambda \wedge \sigma(x,y) \ge \sigma(y,x)$.
   b. $\neg x\mathbf{P}y$.
   c. $\neg x\mathbf{I}y$.
4. $x\mathbf{R}y$, *Incomparability*: There is no reason that justifies preference or equivalence in favor of x or y. $x\mathbf{R}y$ is the conjunction of
   a. $\sigma(x,y) < 0.5$.
   b. $\sigma(y,x) < 0.5$.
5. $x\mathbf{K}y$, *K-Preference*: There exist reasons for justifying preference in favor of x, but with no significant division established between $x\mathbf{P}y$ and $x\mathbf{R}y$. $x\mathbf{K}y$ is held if the conjunction of following propositions is true:
   a. $0.5 \le \sigma(x,y) < \lambda$.
   b. $\sigma(y,x) < 0.5$.
   c. $\sigma(x,y)- \sigma(y,x) > \beta/2$

Let O be the set of alternative portfolios found by an algorithm, we can calculate the number of portfolios that are strictly preferred to each $x \in O$ as follows:

$$\mathrm{S}(O,x) = \{y \in O \mid y\mathbf{P}x\}, \tag{7}$$

and, from this, the *non-strictly outranked frontier* in O is defined as $\mathrm{NS}(O) = \{x \in O \mid \mathrm{S}(O,x) = \emptyset\}$. The best compromise is in NS(O). However, it is possible to identify more "weak" preference relations in the set, defining:

$$\mathrm{W}(O,x) = \{y \in O \mid y\mathbf{Q}x \vee y\mathbf{K}x\}. \tag{8}$$

Using Equation 8, the *non-weakly outranked frontier* in O is defined as $\mathrm{NW}(O) = \{x \in O \mid \mathrm{W}(O,x) = \emptyset\}$. The best compromise is in NS(O) and NW(O). Even within the set NW(O), a third measure of preference can be established. Consider the preference flow for an alternative x as:

$$\mathrm{F}_n(O,x) = \sum_{y \in O - \{x\}} \sigma(x,y) - \sigma(y,x), \tag{9}$$

note that $F_n(x) > F_n(y)$ denotes a kind of preference relation x on y. Similar to $W(O,x)$ and $S(O,x)$ sets, $F(O,x)$ may be defined as:

$$F(O,x) = \{y \in O \mid F_n(y) > F_n(x)\}, \tag{10}$$

and the *non-net-flow outranked frontier* in O is defined as $NF(O) = \{x \in O \mid F(O,x) = \emptyset\}$. A solution x with $|F(O,x)|=0$ should match with the preferences better than another with a higher value. Thus, identification of the best compromise can be expressed as:

$$x^* = \arg\min_{x \in O} \{|S(O,x)|, |W(O,x)|, |F(O,x)|\}. \tag{11}$$

In Equation 11, the minimization is lexicographical; this involves the minimization of the three objectives in order.

3 Related Works

Several algorithms have been developed to solve SPP with synergy, which range from deterministic heuristic algorithms to more sophisticated metaheuristic techniques. The options in the handling of synergy have been basically two: 1) Modifying the multiobjective evaluation function and 2) Adding artificial projects representing the synergic group, and then add redundancy relations that prevent the artificial project appears at the same portfolio that some project of that group.

P-ACO (*Pareto Ant Colony Optimization*) [6] is an algorithm based on the known metaheuristic of Ant Colony to generate the Pareto front with the most efficient portfolios. Each ant generates a candidate portfolio, and the amount of pheromone deposited by such ant is inversely proportional to the number of solutions that dominate it. The algorithm stores the solutions that have never been dominated, which form an approximation of the Pareto front. P-ACO is able to deal with the synergy between projects, salient characteristic in the algorithm. The algorithm only considers the handling of synergy between groups of two projects. No consideration is made about the DM's preferences and the synergy is handling by modifying the multiobjective function.

APS (*Adaptive Sampling Pareto*) [20] is an algorithm that uses a sampling approach for estimation of the Pareto front, based on Monte Carlo simulation method. APS allows the portfolios selection considering synergy between projects. APS is compared with P-ACO over a hundred instances of test, outperforms it. No preference model is considered and the synergy is handling by modifying the multiobjective function.

RPM (*Robust Portfolio Modeling*) [18] is a support tool for multiobjective decision making for project selection. RPM has an algorithm that develops a comprehensive search with dynamic programming to find all optimal portfolios, which then are indexed according to the DM's criteria through a weighted sum of the objectives. The synergy is handling by means of adding artificial projects and extra redundancy relations.

SS-PPS (*Scatter Search for Project Portfolio Selection*) [3] is an algorithm based on scatter search that addresses the problem of the portfolio selection.

SS-PPS has basically two stages, the first generates a set of efficient starting points using Tabu Search, and the second stage improves the initial set by Scatter Search. No consideration is made about the DM's preferences and the synergy is handling by modifying the multiobjective function.

4 Proposed Algorithm

Our algorithm, ACOS-SPP (*Ant-Colony Outranking System for SPP*) is an ant colony algorithm for solve social portfolio problem. It takes ideas from Dorigo's ACS [7], but incorporates changes in the constructing and updating phases for taking into account the preference model. As most ACO algorithms, ACOS-SPP has two phases: constructing and updating.

During ACOS-SPP constructing phase, each ant selects a portfolio, choosing project by project until the budget is over. The way for selecting next project to portfolio is named *selecting rule*.

Afterwards, the updating phase is performed. All portfolios are evaluated by the preference model (Section 2.2), obtaining the three non-outranked fronts: NS, NW and NF, then pheromone is evaporated and each ant drops pheromone according to non-outranked front of its portfolio.

In this section is described the pheromone structure, and constructing and updating phases for ACOS-SPP.

4.1 Pheromone Structure

The pheromone is modeled as a two-dimensional matrix with a size *N*×*N*, where *N* is the total of projects, so τ has two entries. $\tau_{i,j}$ is the preference of having the projects *i* and j in the same portfolio. The values of τ is in (0, 1], being one the initial value. The pheromone is decreased with a constant factor, ρ, once at the end of each iteration.

4.2 Constructing Phase

The next project selection depends on the selection rule:

$$j_x = \begin{cases} \arg\max_{i\in X}\{\Omega(x,i)\} & \text{if } \wp \le \alpha_1, \\ \mathcal{L}_{i\in X}\{\Omega(x,i)\} & \text{if } \alpha_1 < \wp \le \alpha_2, \\ \ell_{i\in X} & \text{Otherwise,} \end{cases} \tag{12}$$

where j_x is the next project to be incorporated in portfolio *x*, *X* is the project list, Ω(x,*i*) is a function that evaluates the expected benefit to incorporate *i* to portfolio *x*, $\mathcal{L}$ is the roulette selection function based on Ω(*x*,*i*), ℓ is a function that randomly selects an available project, and $\wp$ is a pseudo-random number between zero and one with uniform distribution. The benefit function Ω(*x*,*i*) may be defined as

$$\Omega(x,i) = w \bullet \eta_i + (1-w)\left(\frac{\sum_{j\in x} \tau_{i,j}}{|x|}\right), \tag{13}$$

where x is the current portfolio, i is the candidate project for being added to x, η_i is known as local knowledge, and is a measure of benefit of the project i, $\tau_{i,j}$ is the pheromone between projects i and j, and w is a weight factor between local knowledge and the pheromone. The local knowledge η_i may defined as

$$\eta_i = \frac{\left(\frac{1}{c_i}\right)\sum_{j=0}^{p} \mathrm{f}_j(i)}{\max_{k\in X}\left\{\left(\frac{1}{c_k}\right)\sum_{j=0}^{p} \mathrm{f}_j(k)\right\}}, \tag{14}$$

where c_i is the cost of project i, p is the total of objectives, $\mathrm{f}_j(i)$ is the objective function of project i in the j-th objective, and X is the candidate project list. A high value η_i indicates that the project i has high objectives values with a low cost.

According to Equation 12, the selection rule has three phases: 1) Exploitation (if $\wp \leq \alpha_1$), the project with highest value of Ω is selected, 2)Exploitation-Exploration ($\alpha_1 < \wp \leq \alpha_2$), a roulette selection function is used to promote the best projects but not totally ignore those with low values in Ω, and 3) Exploration, where a project is randomly selected to promote constructing of new portfolios.

The first term in Equation 13 favors projects with high values in η, and the second term favors projects with high pheromone with the current portfolio which is being constructed. The last term allows sensing the performance of the portfolio when completed.

4.3 Updating Phase

An ant lays pheromone depend on the portfolio created, according to

$$\Delta\tau_{i,j} = \begin{cases} 0.25\left(1-\tau_{i,j}\right) & \text{If } c\in \text{NS}, \\ 0.50\left(1-\tau_{i,j}\right) & \text{If } c\in \text{NW}, \\ 1-\tau_{i,j} & \text{If } c\in \text{NF}, \\ 0 & \text{Otherwise.} \end{cases} \tag{15}$$

where i and j are projects in the portfolio c, NS is the non-strictly outranked front, NW is the non-weakly outranked front and NF is the non-net-flow outranked front.

5 Experimental Results

In this section, tests to verify the quality of the results are reported. This section 1) provides the experimental conditions of the tests, and 2) verifies the quality of the results obtained for each type of synergy handling.

5.1 Experimental Conditions

The following configuration corresponds to the experimental conditions that are common to the tests described in this section:

1. *Software*. Operating system, Microsoft Windows 7; programming language, Java; compiler, JDK 1.6
2. *Hardware*. Computer equipment dual-processor Xeon (TM) CPU 3.06 GHz in parallel and 4 GB RAM.
3. *Instances*. Two groups of 30 instances and nine objectives each one. One group with 25 projects, and the second one with 100 projects.
4. *Addressed synergy types*: Redundancy and synergy in objective functions (negative and positive). For instances of 25 projects, there are between three and six synergetic relations; for instances of 100 projects, between 12 and 24.
5. *Performance measurement*. Performance is measured according to the number of Non-Outranked Solutions(NOS) found by the algorithm. In this work, a portfolio x is considered as NOS if has S(O,x) = 0, W(O,x) = 0 and F(O,x) = 0.
6. *Parameters values*: The values of the preference model parameters are λ = 0.67, β = 0.1 and $\epsilon\varepsilon$ = 0.05. The values of ACOS-SPP parameters are w = 0.35, α_1 = 0.7, α_2 = 0.2 and ρ = 0.05.
7. *ACOS-SPP stop criteria*: ACOS-SPP finishes if any of following conditions occurs:
   a) 1000 iterations are reached.
   b) NS, NW, NF fronts remain unchanged for ten iterations.

5.2 Comparison between Synergy Handling Techniques

For instances of 25 projects, an enumerative search was performed for knowing the NOS. Moreover, two versions of ACOS-SPP were created; the first handles the synergy into the multiobjective function, and the second one adding artificial projects. Figure 1 presents a performance comparison chart between both synergy techniques. As can be observed, except for three instances, both methods find the optimal solution. No relevant difference on execution time was observed.

For instances of 100 projects, it was not possible to perform an enumerative search. The preference model was applied to the results provided by each version of the algorithm, and the amount of NOS found for each was counted. Figure 2 presents these results. The performance difference in favor of "synergy in the objective function" was 15% on average.

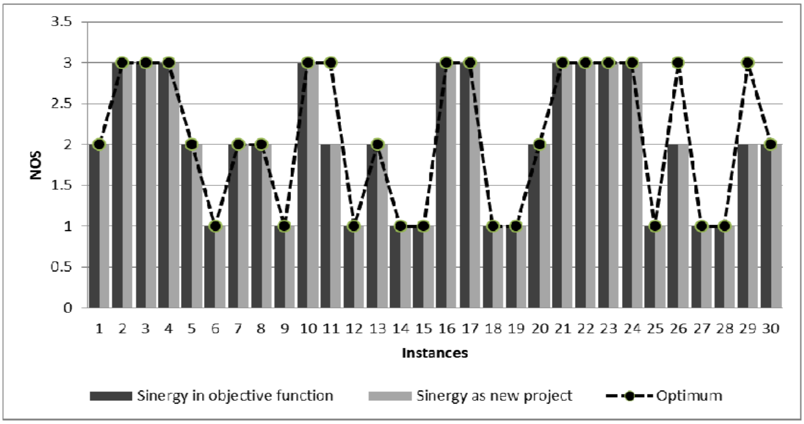


Fig. 1. Performance on instances of 25 projects

Furthermore, significant differences in the execution time were observed. In Figure 3 shows the time run consumed by each version of the algorithm. The time reduction averaged 11% by using "synergy in the objective function".

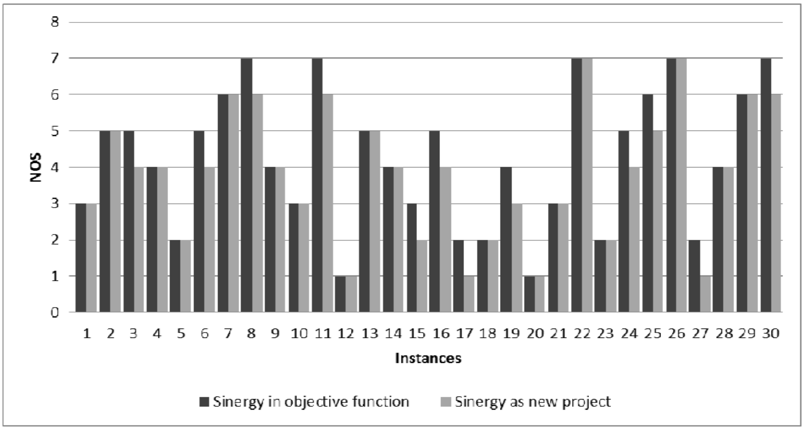


Fig. 2. Performance on instances of 100 projects

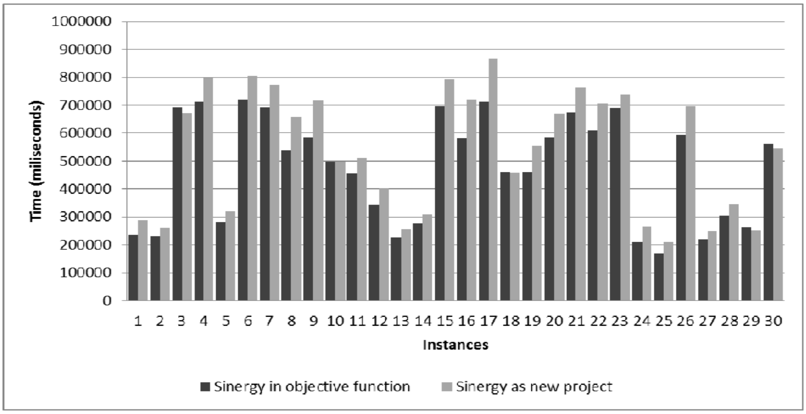


Fig. 3. Execution time on instances of 100 projects

6 Conclusions and Future Work

This article was elaborated to select one of the two most used forms to handle the synergy, and add it into an ant colony algorithm to solve SPP, called ACOS-SPP. The choice had to be made based on experimental evidence. According to Section 5.1, handling of synergy as an adaptation of multiobjective function offers a better performance, in terms of the solution quality and execution time, at least in the instance sets used.

It is important to note that adding artificial projects and redundancy rules is more flexible. In this way, all synergy type presented in Section 2.1 can be addressed, whereas the other needs more specific changes for each synergy type, either in the multiobjective function or in feasibility conditions. Despite its flexibility, it increases the number of unfeasible combinations by adding artificial projects, which can affect the performance of search algorithms, as was the case of ACOS-SPP.

On the test instances used, there was an average increase of 15% in performance, and a 11% reduction of consumed time, by adapting the objective function instead of adding artificial projects.

As future work, we plan to include all synergy types of the Section 2.1. Moreover, to solve other SPP cases, for example partial supporting and scheduling.

References

1. Brans, J., Mareschal, B.: PROMETHEE Methods. In: Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 163–190. Springer, New York (2005)
2. Castro, M.: Development and implementation of a framework for the forming of R & D portfolios in public organizations. Masters Thesis, Nuevo Leon Autonomous University (2007)

3. Carazo, A.F., Gómez, T., Molina, J., Hernández-Díaz, A.G., Guerreo, F.M., Caballero, R.: Solving a comprehensive model for multiobjective project portfolio selection. Computers & Operations Research 37(4), 630–639 (2010)
4. Coello, C., Van Veldhuizen, D.A.C.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002)
5. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. In: Genetic and Evolutionary Computation, 2nd edn. Springer (2007)
6. Doerner, K., Gutjahr, W.J., Hartl, R., Strauss, C., Stummer, C.: Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Annals OR 131, 79–99 (2004)
7. Dorigo, M., Gambardella, L.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
8. Doumpos, M., Marinakis, M., Marimaki, Y., Zopounidis, M.: An evolutionary approach to construction of outranking models for multicriteria classification: The case of ELECTRE TRI method. European Journal of Operational Research 199(2), 496–505 (2009)
9. Durillo, J.J., Nebro, A.J., Coello Coello, C.A., García-Nieto, J., Luna, F., Alba, E.: A study of multiobjective metaheuristics when solving parameter scalable problems. IEEE Transactions on Evolutionary Computation 14(4), 618–635 (2010)
10. Fernández, E., López, E., López, F., Coello Coello, C.A.: Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. Information Sciences 181(1), 44–56 (2011)
11. Fernández, E., López, E., Bernal, S., Coello Coello, C.A., Navarro, J.: Evolutionary multiobjective optimization using an outrankingbased dominance generalization. Computers & Operations Research 37(2), 390–395 (2010)
12. Fernández, E., Navarro, J.: A genetic search for exploiting a fuzzy preference model of portfolio problems with public projects. Annals OR 117, 191–213 (2002)
13. Fernández, E., Navarro, J., Bernal, S.: Multicriteria sorting using a valued indifference relation under a preference disaggregation paradigm. European Journal of Operational Research 198(2), 602–609 (2009)
14. Fernández, E., Flerida, L., Mazcorro, G.: Multi-objective optimisation of an outranking model for public resources allocation on competing projects Int. J. Operational Research 5(2) (2009)
15. García, R.: Hyper-heuristic to solve the problem of social portfolio. Master's Thesis, Madero Institute of Technology (2010)
16. Ghasemzadeh, F., Archer, N., Iyogun, P.: A zero-one model for project portfolio selection and scheduling. Journal of the Operational Research Society 50(7), 745–755 (1999)
17. Hakanan, J., Miettinen, K., Sahlstedt, K.: Simulation-based interactive multiobjective optimization in wastewater. In: International Conference on Engineering Optimization, ENGOPT 2008, Río de Janeiro (2008)
18. Liessio, J., Mild, P., Salo, A.: Preference programming for robust portfolio modeling and project selection. European Journal of Operation Research 181, 1488–1505 (2007)
19. Marakas, G.: Decision Support Systems and Megaputer, 2nd edn. Prentice Hall, Upper Saddle River
20. Reiter, P.: Metaheuristic Algorithms for Solving Multiobjective/ Stochastic Scheduling and Routing Problems. Ph.D. Thesis. University of Wien (2010)

21. Roy, B.: The Outranking Approach and the Foundations of ELECTRE methods. In: Reading in Multiple Criteria Decision Aid, pp. 155–183. Spinger (1990)
22. Roy, B.: Multicriteria Methodology for Decision Aiding. Kluwer, Dordrecht (1996)
23. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A comparative case study and the Strength Pareto Evolutionary Algorithm. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)

Improving the Performance of Heuristic Algorithms Based on Exploratory Data Analysis

Marcela Quiroz C.[1], Laura Cruz-Reyes[1], José Torres-Jiménez[2],
Claudia G. Gómez S.[1], Héctor J. Fraire H.[1], and Patricia Melin[3]

[1] Instituto Tecnológico de Ciudad Madero, México
qc.marcela@gmail.com, {lcruzr,hfraire}@prodigy.net.mx,
cggs71@hotmail.com
[2] CINVESTAV-TAMAULIPAS, México
jtj@cinvestav.mx
[3] Tijuana Institute of Technology, México
pmelin@tectijuana.mx

Abstract. This paper promotes the application of empirical techniques of analysis within computer science in order to construct models that explain the performance of heuristic algorithms for NP-hard problems. We show the application of an experimental approach that combines exploratory data analysis and causal inference with the goal of explaining the algorithmic optimization process. The knowledge gained about problem structure, the heuristic algorithm behavior and the relations among the characteristics that define them, can be used to: a) classify instances of the problem by degree of difficulty, b) explain the performance of the algorithm for different instances c) predict the performance of the algorithm for a new instance, and d) develop new strategies of solution. As a case study we present an analysis of a state of the art genetic algorithm for the Bin Packing Problem (BPP), explaining its behavior and correcting its effectiveness of 84.89% to 95.44%.

1 Introduction

Most optimization problems in the real world belong to a special class of problems called NP-hard, which means that there are no known efficient algorithms to find the optimal solution in the worst case. To solve these problems, the efforts of many researchers have resulted in a variety of heuristic algorithms that have shown satisfactory performance. However, to date there is no algorithm that is best for all possible situations. One of the main challenges of behavioral analysis of heuristic algorithms is to identify what strategies make an algorithm to show an improved performance and under what conditions they get it.

In this paper we formulate an experimental approach for a comprehensive study of the optimization process in order to identify inherent relationships among the factors that affect the algorithmic performance. The proposed approach combines methods of exploratory data analysis and causal inference in three stages. In the

O. Castillo et al. (Eds.): Recent Advances on Hybrid Intelligent Systems, SCI 451, pp. 361–375.
springerlink.com

characterization phase factors that influence performance are identified and quantified by indexes. In the characteristics refining stage incorrect and redundant indexes are discarded. In the study of relations stage an analysis of the characteristics of the optimization process is made in order to obtain performance relations that eventually will become algorithm behavior explanations.

To evaluate the contribution of the proposed approach we performed a study of the optimization process for the Bin Packing Problem (BPP). BPP is considered NP-hard [1, 2] and consists in packing a set of n items of different sizes $W=\{w_1,\ldots,w_n\}$ in the minor number of fixed size bins without violating the capacity c of any bin. BPP has an extensive number of industrial and logistic applications and frequently happens as a sub-problem in several practical problems [3, 13]. The case study confirms the importance of applying exploratory data analysis as a guide for understanding the performance of algorithms.

2 Performance Analysis of Heuristics Algorithms

The so-called NP-hard problems are of great interest in computer science. One feature of these problems is that the exact algorithms used to solve require an exponential amount of time in the worst case. In other words, these problems are very difficult to solve [1]. In these conditions it is necessary to use heuristic algorithms that provide approximate solutions in a reasonable time, but do not guarantee the optimal solution. Heuristic algorithms are procedures that used strategies based on common sense in order to obtain high quality solutions (not necessarily optimal) to complex problems efficiently.

The criteria for measuring the performance of heuristic algorithms depend on the methods chosen for characterization, which can be theoretical or experimental. In the first, for each algorithm, mathematical analysis are used to determine the amount of resources required as a function of the size of the better, worse or average case. The latter is based on experimentation for the characterization and, unlike the theoretical methods, allows describing the behavior of specific cases.

The theoretical study of heuristics algorithms performance is unusual, because the randomness in some of these algorithms and the complexity of the optimization problems that impede a proper mathematical analysis. Moreover, the applicability of the theoretical results is very limited, because these are obtained based on idealized conditions that do not occur in practical situations [4, 5].

Despite the importance of the experimental analysis of performance, this has not been sufficiently exploited in the study of heuristics algorithms. One of the reasons which make difficult this study is that, generally, the algorithms are considered as black boxes whose inner workings are unknown.

Recent works propose tools that can be taken into account to assist the analysis of the factors that influence the algorithmic performance and thus the explanation of performance [4, 6, 7, 8, 9]. In general, tools for data analysis (factors) proposed in these works can be grouped into two categories: exploratory data analysis techniques and confirmatory data analysis techniques.

In the exploratory data analysis (EDA) the aim is to obtain knowledge of the data set and its underlying structure. Includes statistical methods, tabular comparisons, graphical analysis, causal inference and multivariate analysis that make possible the construction of a model that describes the set of relations of the factors under study [10, 11, 12].

Confirmatory data analysis (better known as statistical hypothesis testing), begins with assumptions (models) about functional relationships between variables (factors) of the data. Includes estimations of parameters of the models and statistical hypothesis testing to complement and validate proposed models [8, 11, 12].

The identification of appropriate data analysis techniques for the types of problems and opportunities that occur when analyzing the algorithmic performance is a research area still in development. The selection of tools to use depends on the characteristics of the data under study (problem and algorithm). For example, graphical methods of exploratory data analysis are most appropriate to analyze trends in large multivariate data sets. Also, methods of confirmatory data analysis can be problematic, depending on how much knowledge exists about the function (model), therefore, methods of data analysis, with exploratory bases, that do not start assuming functional relationships between the factors studied, are most appropriate when the objective is to discover relationships and evaluate various models [4].

3 Characterization of the Optimization Process

The characterization of the problem and the solution process is an essential part in the performance analysis of algorithms, and allows identifying the factors that influence the algorithmic behavior. A quality performance analysis requires the definition of appropriate indexes to quantify the features that impact final performance.

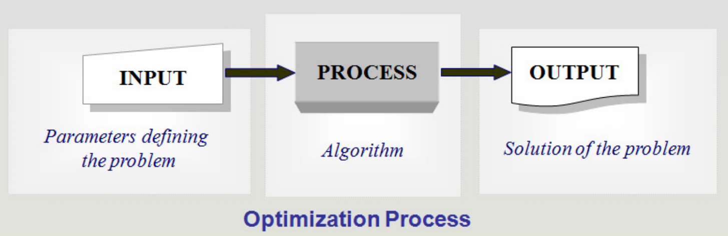


Fig. 1. The optimization process of a problem

To gain insight into the performance of a heuristic algorithm on an optimization problem, it is necessary to make a comprehensive study of the entire solution process. The optimization process can be understood as the act of solving an optimization problem (input) using an algorithm (process), obtaining a final solution (output). This process is illustrated in Figure 1.

The input consists of an instance of the optimization problem to solve, composed of a set of specific parameters that define it. The process includes the set of strategies used to solve the problem, like the functions and parameters used by the algorithm. The output provides the solution of the problem and some important measures of performance, like number of iterations and execution time.

4 An Experimental Approach to Study the Optimization Process

Figure 2 shows the proposed approach for the experimental analysis of heuristics algorithms in optimization problems. The main objective of the *characterization* stage is to identify, in each phase of the optimization process, relevant and measurement feasible performance factors. These factors are characterized through characterization functions (indexes) that provide useful information to describe the algorithmic performance. In the second stage, *characteristics refining*, the indexes defined in the characterization stage are analyzed using exploratory techniques in order to rule out incorrect, redundant or irrelevant indexes. If necessary, new indexes are defined using multivariate analysis techniques.

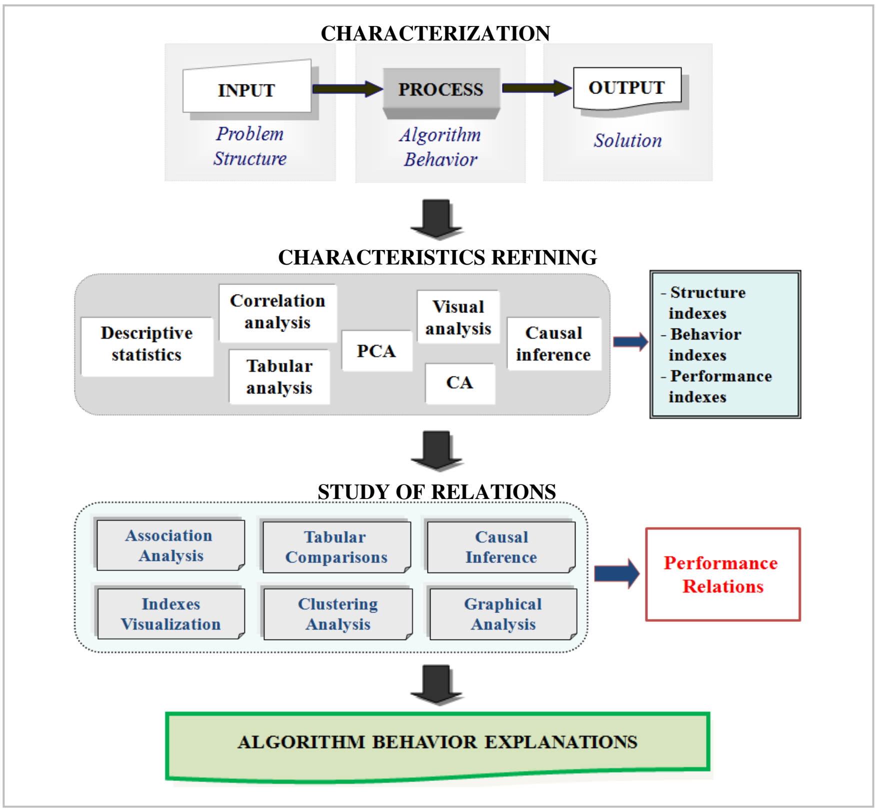


Fig. 2. Experimental approach to study the performance of heuristic algorithms

In the third stage, *study of relations*, an exploratory analysis of the characteristics of the optimization process is done in order to obtain performance relations that explain the behavior of the heuristic algorithm under study. For this purpose we use multivariate statistical methods, tabular and graphical analysis, data visualization techniques and causal analysis. The knowledge gained as a result of the study of relations allows understanding the behavior of the heuristic algorithm, explaining how its final performance is affected by several factors that cause it, visualizing possible improvements in its structure.

5 Case Study: Bin Packing Optimization Process

In this section we present the application of the proposed experimental approach to the characterization of the bin packing problem (BPP) by analyzing the performance of a state of the art genetic algorithm named HGGA-BP [13]. We introduce a new set of indexes for BPP and the algorithm behavior characterization; in addition, we show the redesign of HGGA-BP algorithm structure and present new experimental results.

5.1 Phase 1: Characterization

This stage is carried out to characterize the optimization problem, the behavior of the algorithm of interest and the final performance. Having identified the factors that influence the optimization process, it is analyzed which aspects can be measured in each category, and indexes that characterize these factors are defined.

Bin Packing Characterization

Characterize the structure of an instance of BPP is a key to predict the behavior that will have a heuristic algorithm at the time of the solution. It is known that factors like the number of items, the central tendency of the weights and their distribution, impact the degree of difficulty that an instance can have on a solution algorithm. The challenge is to formulate indicators that quantify these factors.

We carried out the compilation of 1668 benchmark instances recognized by the scientific community. These test instances were taken from Internet sites [14, 15, 16, 17]. Descriptive information for each instance of the problem is characterized by indexes which use information from the parameters of the problem in that instance. Different authors have proposed set of indexes for the characterization of BPP [18, 19].

With the goal of modeling the structure of a instance, Pérez and Cruz [18] formulated a specific-purpose indexes set, formed by five difficulty indexes: instance size (*p*), occupied capacity (*t*), dispersion (*d*), factors (*f*), and bin usage (*b*). In other work, Álvarez [19] proposed 21 indexes based on descriptive statistics, these indexes characterize the weight distribution of the BPP items and can be grouped into four categories of statistical measures: centralization, dispersion, position and form of the frequency distribution of the weights. We studied all these indexes by analyzing if they allowed discriminating between different BPP instances. To assist in this study and contribute with the characterization of BPP, each instance was plotted showing the distribution of the weights of items in relation to the bin capacity.

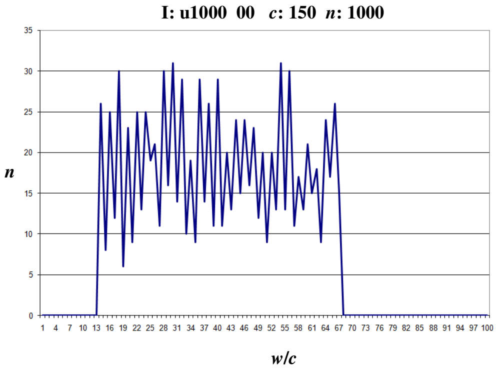


Fig. 3. Weight distribution graph for a BPP instance

Figure 3 shows the distribution plot of a BPP instance. At the top of the plot one can see the name of the instance (I: u1000_00), the bin capacity (*c*: 150) and the number of items (*n*: 1000). The horizontal axis represents the weight of the items as a percentage of the bin capacity ($0 < w_i/c \leq 1$). The vertical axis counts the number of items in each weights percentage. As it can be observed, the weights of the items are uniformly distributed between 13% and 67% of the bin capacity. The chart also shows that the number of items with each weight varies between 6 and 31.

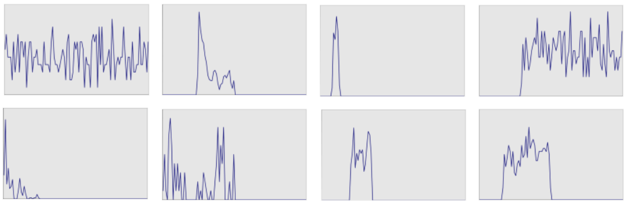


Fig. 4. Different weight distribution graphs for a BPP instances

Figure 4 shows representative plots of the set of instances. It can be observed the variety of forms of the frequency distribution of the weights of the items, as well as the ranges of weights, suggesting the importance of a correct characterization of these factors. The analysis of the distribution plots allows identifying differences between different classes of instances and helps to discover characterization indexes.

The study of the weight distribution graphs revealed that the indexes proposed in previous works for BPP characterization are able describe much of the structure of an instance of BPP. However it seems that there are certain aspects that were not taken into account by the authors, factors that are important to point out differences between instances. In this work we propose five new indexes of characterization. These new indexes can: locate the distribution range of the set of weights, identify important trends in the weights of items, measure the frequency of repetition of the weights, and contribute to the characterization of the form of the distribution of the weights. These indexes are defined below.

Lower_Weight and *Upper_Weight* indexes, respectively, represent the weight of the smallest and the biggest item, in relation to the bin capacity, and allow locating the beginning and the end of the weights distribution.

Multiplicity characterizes the average number of repetitions of each weight and helps to identify trends or peaks in the weights frequency distribution. Equation 1 defines this index, m_i is the number of items with weight $s_i \in S$, where S include the $\hat{n}$ different weights, without repetitions ($1 \leq \hat{n} \leq n$).

$$Multiplicity = \frac{\sum_{i=1}^{\hat{n}} m_i}{c} \tag{1}$$

Max_Repe index represents the maximum frequency of repetition of a weight in the set of items. Comparing this measure values with those of *Multiplicity* index, may suggest the existence of points of the weights frequency distribution having a greater accumulation items.

Uniformity measures the degree of uniformity of the weights distribution, from the division of the range of the weights into four segments of equal magnitude and the differences between the expected number and the actual number of items in each subrange. Given: n (number of items), $W = \{w_i : w_i < w_{i+1}\}$, $1 \leq \forall\, i \leq n$ (array of weights in increasing order), $R = \max(W) - \min(W)$ (range of the weights), $r_i = \min(W) + iR/4$, $0 \leq i \leq 4$ (division of subranges). The set of items contained in each subrange j is: $B_j = \{w \in W : r_{j-1} < \mathrm{w} \leq r_j\}$, $1 \leq j \leq 4$. Equation 2 measures the degree of uniformity of distribution of the set of weights. A value of *Uniformity* close to one represents a uniform distribution.

$$Uniformity = 1 - \frac{\sum_{j=1}^{4} \left| \left(\frac{n}{4} - \left| B_j \right| \right) \right|}{n} \tag{2}$$

Algorithm Characterization

The behavior of the algorithm was measured by three indexes: a) *New_individuals* which is the average number of new individuals added to the population in each generation; b) *Deviation_Fitness* characterizes the average deviation in the fitness of the solutions of the population; c) *Deviation_Best* represents the deviation in the fitness of the best solutions of each generation [13].

Final Performance Characterization

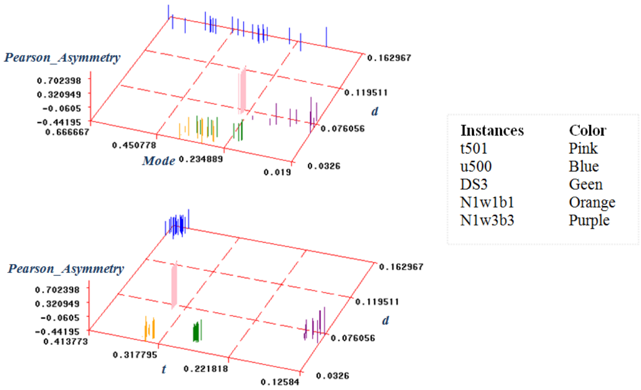
The final performance of the algorithm was measured by means of three indexes: a) *Theoretical_Ratio*, which is the ratio between, the number of bins of the obtained solution, and the number of bins of the optimal solution; b) *Best_Fitness*, the fitness of the best solution obtained, for an instance of BPP, in *r* runs of the algorithm; c) *Generation*, represents the generation in which the algorithm finds the best solution for an instance.

5.2 Phase 2: Characteristics Refining

After an initial review of BPP indexes, it appears that several features of the structure of the problem (BPP) are characterized by a large number of indexes, so it is possible that some of them are redundant or unnecessary. The exploratory analysis of the measures helps to verify this fact. The correlation matrix, revealed the existence of association between certain variables that characterize the centralization, dispersion and form of the weights distribution. If there are lineal associations between pair of variables it is possible that statistical analysis based in the correlation matrix does not make sense or produce incorrect results. To avoid this, some indexes were discarded to eliminate redundancy and possible problems in future analyzes. Overall, it was decided to keep indexes requiring fewer calculations.

Having identified the new set of indexes, it is necessary to analyze the contribution and consistency of each measure. The aim of this study is to identify and eliminate those indexes that do not provide meaningful information for the characterization of the structure of the BPP instances. Inconsistent or incorrect indexes only add wrong information to the description of an instance and do not allow discriminating between instances of different nature.

An example of inconsistent indexes can be seen in Figure 5 that shows two scatter plots for characterization indexes of central tendency (*Mode* and *t*),



Instances	Color
t501	Pink
u500	Blue
DS3	Geen
N1w1b1	Orange
N1w3b3	Purple

Fig. 5. Scatter plots for *Mode*, *t*, *d* and *Pearson_Asymmetry* for BPP instances

dispersion (*d*) and form (*Pearson_Asymmetry*). These indexes were applied to the characterization of five sets of instances with different structures. In the first plot, we use the index *Mode* proposed by Álvarez [19] as a measure of central tendency of the set of weights of items and it can be seen that this measure cannot discriminate between different instances. In the second plot, substituting *Mode* for *t* proposed by Pérez and Cruz [18], it can be clearly seen the importance of the information provided by the latter, by interacting with *d* and *Pearson_Asymmetry*; the different instances sets are better identified in groups.

The exploratory data analysis suggested that indexes *Mode*, *Quartiles*, *Deciles*, *Percentiles*, *Pearson_Coefficient_Mode*, *Bowley_Skewness* and *Standard_Error* adapted for BPP by Álvarez [19] did not offer significant information for the characterization and some of their values are inconsistent with the structure of the BPP instances.

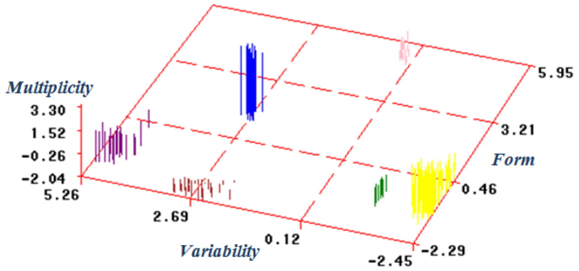
The exploratory analysis of the indexes for BPP showed the contribution and importance of each, in the characterization of the problem. Starting from an initial set of 31 indexes, 10 were discarded due to redundancy and other 7 were eliminated by their inconsistency or no contribution. The final set of characterization indexes is shown in Table 1.

Table 1. Final set of BPP characterization indexes

Type	Indexes
Size	*p* [18]
Centralization	*t* [18]
Dispersion	*Range* [19] *Variation_Coefficient* [19] *d* [18]
Form	*Pearson_Coefficient* [19] *Kurtosis* [19] *Uniformity* [This work]
Location	*Lower_weight* [This work] *Upper_weight* [This work]
Relations items/Bins	*f* [18] *b* [18]
Multiplicity	*Multiplicity* [This work] *Max_Repe* [This work]

The first component, called *Variability*, measures the variation of the weights of items and groups the measures of central tendency, dispersion and location. The second, called *Form*, describes the weight distribution, grouping measures of form, location and size of the problem. The last component, called *Multiplicity*, measures the frequency of occurrence of the weights of items.

The principal components allowed plotting the characteristics of the six groups of instances, in Figure 6 is possible to distinguish clearly between six different classes of instances analyzed.



Instances	Color
t501	Pink
u500	Blue
DS3	Green
N3c1w1	Purple
Hard28	Brown
Was 1	Yellow

Fig. 6. Scatter plot for the first three principal components of the set of indexes

The analysis of the scatter plot of the weights distribution (defined in Section 5.1) proved the accuracy and discriminatory power of the proposed indexes, because the similarities and differences between the characteristics of the instances belonging to the six sets agree with those observed in the plots.

The analysis of algorithm and final performance measures showed that there were no incorrect, redundant or inconsistent indexes, as each captures important aspects of HGGA-BP algorithm and its performance.

5.3 Phase 3: Study of Relations

In this stage, the final set of indexes of characterization is studied in order to discover relationships of performance detailing the structure of the optimization process. The main objective of this study is to gain knowledge that could explain the performance of the algorithm.

To explain the optimization process is necessary to analyze the relation between the characterization indexes that define: the structure of BPP instances, the behavior shown by the algorithm and the final performance. These relationships were obtained by means of the association analysis, scatter plots and causal inference. Initially all the analyses were performed on the entire set of instances, for an overview of the performance relationships that define the behavior of the algorithm and the difficulty of the BPP instances.

In order to discover the underlying relation structure between BPP characteristics and the final performance of the algorithm we use causal inference. The PC learning algorithm of the TETRAD tool [20] was applied for automatically learning a causal model from the measures of the principal components of BPP (*Variability*, *Form* and *Multiplicity*) and the three performance indexes (*Theoretical_Ratio*, *Best_Fitness* and *Generation*). The causal graph obtained through the analysis is presented in Figure 7, which clearly shows that BPP instances characteristics have a direct influence on the final performance of the algorithm.

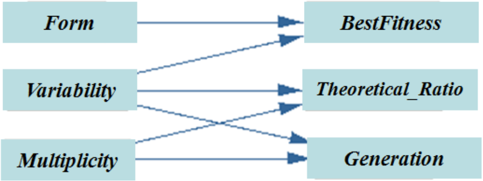


Fig. 7. Causal relations between problem characteristics and final performance

Figure 8 shows the causal graph for the algorithm characteristics (*Deviation_Fitness*, *New_individuals* and *Deviation_Best*) and the final performance. It seems that *New_individuals* measure (highlighted in yellow) have no relation to the final performance, it seems that the procedures used to create new individuals are not helping to escape from local optima.

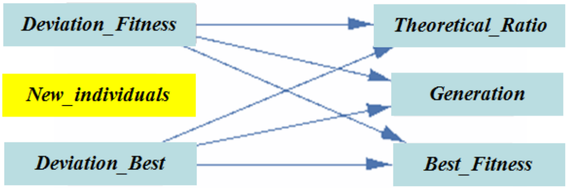


Fig. 8. Causal relations between algorithm behavior and final performance

Figure 9 includes two scatter plots that associate problem and algorithm behavior measures and the theoretical radius of the final solutions of the algorithm. Each plot shows the way a feature of problem (*t* is average weight of the items) affects the behavior (*Deviation_Best* and *Deviation_Fitness*) and the final performance of the algorithm (*Teoretical_Ratio*). From the examination of the plots it can be seen that instances with large items are the easiest. Also, in some cases, diversification of the population helps the algorithm to find better solutions. The plot a) shows a very important algorithmic behavior: for instances with small and medium weight the algorithm does not achieve good results when the best solutions are highly associated with each other, indicating that the algorithm is stuck at local optima. In plot b) we use different colors to highlight different sets of instances, it can be observed that within the same class of instances there are easy and difficult cases.

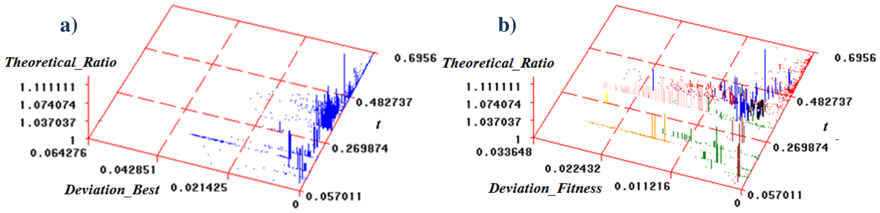


Fig. 9. Relations between problem, algorithm and final performance measures

Figure 10 shows scatter plots illustrating the behavior shown by the algorithm in two instances classes. The difference in the complexity inside every instance class appears to be caused by the shape of the weight distribution as well as the multiplicity and size of the problem. The instances for which the optimal was found were marked in blue and the more "difficult" were marked with black.

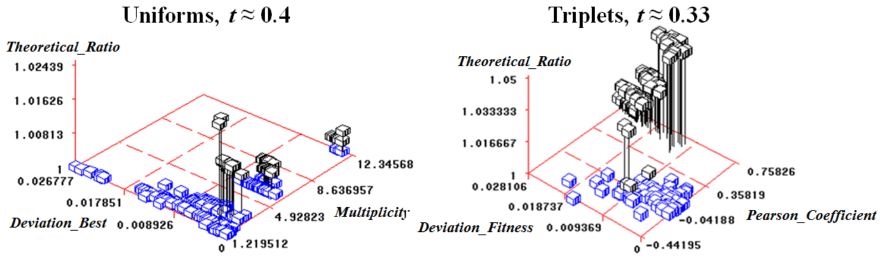


Fig. 10. Relations between problem, algorithm and performance measures

After analyzing the performance relations we can infer the following algorithm behavior explanations:

- The main features that reflect the degree of difficulty of an instance of BPP are the central tendency and variability of the weights of items.
- For instances that have similar values of central tendency and variability measures, the degree of difficulty of the instances is influenced by the form and multiplicity of the set of weights.
- The easier instances are those with larger items and greater variability in the weights of items.
- The bigger the problem, relative to the size of the bin (this implies that the items are large and/or there are many items), the instance is easier; this may possibly be because the solution space is smaller and flatter.
- The strategies included in the algorithm had good results for instances with large items but do not appear to be adequate to address instances with medium and small items.
- For "difficult" instances, strategies for generation of individuals do not allow to add diversity in the population and genetic operators do not lead the algorithm to new regions.

5.4 Redesign of the Algorithm and Performance Improvement

The algorithm behavior explanations gained from the experimental analysis of the optimization process of the HGGA-BP algorithm showed that the strategies used for the creation of solutions and the search space exploration were not helping the algorithm to obtain good solutions for instances with medium and small items, this knowledge was used to redesign the algorithm structure by modifying the procedures that had the greatest impact on the algorithm performance.

After a detailed analysis of the main strategies that define the structure and behavior of the algorithm, the procedures used for the creation and exploration of solutions were modified in order to include a greater diversification and exploitation of the search space. We incorporated efficient random heuristics to increase the solutions diversity and to prevent the premature convergence of the algorithm [13].

Diversification was added to the population by: a) including different packing strategies in the creation of individuals; b) increasing the size of the population; c) adding new individuals to replace individuals with repeated fitness.

Intensification was added to the search by: a) including a new strategy to improve the filling of the bins; b) including a new crossover operator for good solutions; c) including a new crossover operator for the improvement of solutions fitness.

The application of the knowledge obtained in the analysis of the characteristics of the optimization process of BPP using the algorithm HGGA -BP, yielded a significant improvement in the performance of the algorithm. The final results are shown in Table 2, which includes the results obtained by the old and the new version of the algorithm. For every class of instances, it first shows the number of test cases (Column inst.), followed by the results obtained by each version of HGGA-BP: the number of optimal solutions found (Column opt.), the average execution time measured in seconds (Column time (s)) and the average generation (Column gen.).

The experimental results make obvious the usefulness and applicability of the algorithm behavior explanations obtained by means of the experimental approach proposed in this work. The effectiveness of the heuristic algorithm HGGA-BP was improved from 84.8% to 95.4%. Also, it was possible to outperform results of the best state of the art algorithms. HI-BP algorithm [21] was outperformed in three instances of the Gau 1 set (TEST0058, TEST0082 and TEST0005). Perturbation-SAWMBS algorithm [22] was outperformed in three instances of Hard28 set (hBPP640, hBPP531 and hBPP814).

Table 2. Improved results for the HGGA-BP algorithm

Class	inst.	HGGA-BP Original			HGGA-BP Improved		
		opt.	time (s)	gen.	opt.	time(s)	gen.
Uniform	80	52	8.53	47	79	1.67	13
Triplets	80	0	1.90	69	80	5.14	53
Data Set 1	720	692	2.43	43	718	2.67	19
Data Set 2	480	450	0.57	10	480	0.90	6
Data Set 3	10	8	1.75	8	9	8.10	77
Was 1	100	99	0.05	6	100	0.04	3
Was 2	100	98	0.34	28	100	0.99	32
Gau 1	17	9	0.59	54	15	1.92	40
Hard28	28	5	1.53	90	8	6.75	87
NIRUP	53	3	1.09	81	3	4.45	89
Total	**1668**	**1416**	**1.88**	**44**	**1592**	**3.26**	**42**
Effectiveness		**0.848**			**0.954**		

6 Conclusions and Future Work

This work shows that the exploratory data analysis is useful in the study of NP-hard problems and heuristics algorithms, because it allows identifying clearly what are the characteristics of the problem that impact the final performance of the algorithms and to what extent they do.

We propose an experimental approach that combines exploratory data analysis techniques for the performance analysis of heuristic algorithms with the objective to explain the algorithmic optimization process. As a case study we perform a comprehensive study of the optimization process for the Bin Packing Problem (BPP) solved by a heuristic algorithm.

We studied and characterized the structure of bin packing instances and we proposed 5 indexes for the bin packing characterization. The optimization process of BPP was explained by relations, which allowed us to understand the behavior of a genetic algorithm in the solution of BPP instances with different structures. The case study confirmed the importance of applying exploratory data analysis techniques as a guide for understanding the performance of algorithms. The knowledge gained from models of explanation led to improve performance of the HGGA-BP algorithm, correcting the effectiveness of 84.89% to 95.44% for a set of 1668 instances, outperforming the effectiveness of the best state of the art algorithms in some instances.

As future work, in the case of the genetic algorithm, we are planning to make a deeper study within the features of the algorithm which seem to be the most promising to increase the final performance and design new models that includes the impact of the parameters that control the behavior of the algorithm in order to obtain more detailed explanations of the optimization process.

In general, it is expected that the work presented in this paper represents a guideline to study the performance of heuristic algorithms through the application of exploratory data analysis techniques in other algorithms and optimization problems. The experimental approach presented in this work allows obtaining a deep understanding about algorithmic behavior; this knowledge can be used to improve the performance.

References

[1] Garey, M.R., Jonson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company. A classic introduction to the field (1979)

[2] Basse, S.: Computer Algorithms, Introduction to Design and Analysis. Editorial Addison-Wesley Publishing Company (1998)

[3] Cruz Reyes, L., Nieto-Yáñez, D.M., Rangel-Valdez, N., Herrera Ortiz, J.A., González B, J., Castilla Valdez, G., Delgado-Orta, J.F.: DiPro: An Algorithm for the Packing in Product Transportation Problems with Multiple Loading and Routing Variants. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 1078–1088. Springer, Heidelberg (2007)

[4] McGeoch, C.C.: Experimental Analysis of Algorithms. In: Pardalos, P.M., Romeijn (eds.) Handbook of Global Optimization, vol. 2, pp. 489–513 (2002)

[5] Hoos, H.H., Stützle, T.: Empirical Analysis of Randomized Algorithms. In: Handbook of Approximation Algorithms and Metaheuristics. Chapman & Hall/CRC, Taylor & Francis Group (2007)

[6] Hooker, J.N.: Needed: An empirical science of algorithms. Operations Research 42(2), 201–212 (1994)

[7] Barr, S., Golden, L., Kelly, P., Resendez, M., Stewart, R.: Designing and Reporting on Computational Experiments with Heuristic Methods. Journal of Heuristics 1, 9–32 (1995)

[8] Cohen, P.R.: Empirical Methods for Artificial Intelligence. The MIT Press, Cambridge (1995)

[9] Lemeire, J.: Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs. PhD thesis. Vrije Universiteit Brussel (2007)

[10] Tukey, J.W.: Exploratory Data Analysis. Addison-Wesley (1977)

[11] Hartwig, F., Dearing, B.E.: Exploratory Data Analysis. Sage University Paper Series on Quantitative Research Methods, vol. 16. Sage, Newbury Park (1979)

[12] Liu, X.: Intelligent Data Analysis: Issues and Challenges. The Knowledge Engineering Review 11, 365–371 (1996)

[13] Quiroz, M.: Caracterización de Factores de Desempeño de Algoritmos de Solución de BPP. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México (2009)

[14] Beasley, J.E.: OR-library: Distributing test problems by electronic mail. Journal of the Operational Research Society 41(11), 1069–1072 (1990), http://people.brunel.ac.uk/~mastjjb/jeb/orlib/binpackinfo.html

[15] Klein, R., Scholl, A.: Bin Packing benchmark data sets, http://www.wiwi.uni-jena.de/Entscheidung/binpp/

[16] Euro Especial Interest Group on Cutting and Packing. One Dimensional Cutting and Packing Data Sets, http://paginas.fe.up.pt/~esicup/tiki-list_file_gallery.php?galleryId=1

[17] Cutting and Packing at Dresden University. Benchmark data sets, http://www.math.tu-dresden.de/~capad/cpd-ti.html#pmp

[18] Pérez, J., Pazos, R.R.A., Frausto, J., Rodríguez, G., Romero, D., Cruz, L.: A Statistical Approach for Algorithm Selection. In: Ribeiro, C.C., Martins, S.L. (eds.) WEA 2004. LNCS, vol. 3059, pp. 417–431. Springer, Heidelberg (2004)

[19] Álvarez, V.: Modelo para representar la Complejidad del problema y el desempeño de algoritmos. Tesis de maestría, Instituto Tecnológico de Cd. Madero, Tamaulipas, México (2006)

[20] The TETRAD Project: Causal Models and Statistical Data. TETRAD Homepage, http://www.phil.cmu.edu/projects/tetrad/

[21] Alvim, A.C.F., Ribeiro, C.C., Glover, F., Aloise, D.J.: A hybrid improvement heuristic for the onedimensional bin packing problem. Journal of Heuristics 10(2), 205–229 (2004)

[22] Fleszar, K., Charalambous, C.: Average-weight-controlled bin-oriented heuristics for the onedimensional bin-packing problem. European Journal of Operational Research 210(2), 176–184 (2011)

Local Survival Rule for Steer an Adaptive Ant-Colony Algorithm in Complex Systems

Claudia Gómez Santillán[1,2], Laura Cruz Reyes[1], Elisa Schaeffer[3],
Eustorgio Meza[2], and Gilberto Rivera Zarate[1]

[1] Instituto Tecnológico de Ciudad Madero (ITCM). 1ro. de Mayo y Sor Juana I. de la Cruz s/n CP. 89440, Tamaulipas, México. Phone: (52) 833 3574820 Ext. 3024
cggs71@hotmail.com, lcruzreyes@prodigy.net.mx,
riveragil@gmail.com

[2] Instituto Politécnico Nacional, Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada (IPN-CICATA). Carretera Tampico-Puerto Industrial Alt., Km.14.5. Altamira, Tamps., México. Phone: 018332600124
emezac@ipn.mx

[3] CIIDIT & FIME, Universidad Autónoma de Nuevo León (UANL), Av. Universidad s/n, Cd. Universitaria, CP.66450, San Nicolás de los Garza, N.L., México
elisa@yalma.fime.uanl.mx

Abstract. The most prevalent P2P application today is file sharing, both among scientific users and the general public. A fundamental process in file sharing systems is the search mechanism. The unstructured nature of real-world large-scale complex systems poses a challenge to the search methods, becasuse global routing and directory services are impractical to implement. In this paper, a new ant-colony algorithm, Adaptive Neighboring-Ant Search (AdaNAS), for the semantic query routing problem (SQRP) in a P2P network is presented. The proposed algorithm incorporates an adaptive control parameter tuning technique for runtime estimation of the time-to-live (TTL) of the ants. AdaNAS uses three strategies that take advantage of the local environment: learning, characterization, and exploration. Two classical learning rules are used to gain experience on past performance using three new learning functions based on the distance traveled and the resources found by the ants. The experimental results show that the AdaNAS algorithm outperforms the NAS algorithm where the TTL value is not tuned at runtime.

Keywords: parameter tuning, search algorithm, peer-to-peer, adaptive algorithm, local environment, ant-colony algorithms.

1 Introduction

Although popular for other uses, the World Wide Web is impractical for user-to-user file sharing as it requires centralized infrastructure such as an HTTP server.

P. Melin et al. (Eds.): Soft Comp. for Recogn. Based on Biometrics, SCI 312, pp. 245–265.
springerlink.com

In the past decade, a new class of networks called *peer-to-peer* (P2P) systems began to spread as a solution to the increasing demand of file sharing among Internet users. In P2P networks, the users interconnect to offer their files to one another [32]. The participants, called *peers*, may connect and disconnect freely, and do so constantly, which triggers frequent changes in the network structure [35].

One of the main advantages is that peers are equal in terms of functionality and tasks which are developed. This produces high fault tolerance and auto-organization: peers form unstructured networks with an acceptable connectivity and performance. The *Semantic Query Routing Problem* (SQRP) consists in deciding, based on a set of keywords, to which neighbor to forward the query to search files related with the keywords [25, 35].

The lack of global structure caused that flooding-based search mechanisms have been mainly employed. Flooding-based mechanisms are simple, but unfortunately generate vast amounts of traffic in the network and may produce congestion on Internet. Existing approaches for SQRP in P2P networks range from simple broadcasting techniques to sophisticated methods [27, 32, 33]. The latter includes proposals based on *ant-colony systems* [14] that are specifically suited for handling routing tables in telecommunications. There exist few algorithms used for SQRP, including SemAnt [25] and Neighboring-Ant Search (NAS) [11], the latter based on the former. In this work we propose an algorithm as an extension to NAS, called the Adaptive Neighboring-Ant Search (AdaNAS). AdaNAS is hybridized with three local strategies: learning, structural characterization and exploration. These strategies are aimed to produce a greater amount of results in a lesser amount of time. The time-to-live (TTL) parameter is tuned at runtime based on the information acquired by these three local strategies.

2 Background

In order to place the research in context, this section is divided in three parts. The first part models a P2P network with graph theory, and in we continue with structural characterization. The part describes the basic ant-colony algorithms for SQRP algorithms and the last part explains parameter tuning and adaptation.

2.1 Graph Theory

A P2P network is a distributed system that can be modeled mathematically as a *graph*, $G = (V,E)$, where V is a set of *nodes* and $E \subseteq V$ x V is a set of (symmetrical) *connections*. For more information on graph theory, we recommend the textbook by Diestel [13]. Each peer in the network is represented by a node (also called a *vertex*) of the graph. The direct communications among the peers are represented by the connections (also called the *edges*) of the graph. We denote by n the number of nodes in the system and identify the nodes by integers, $V = \{1, 2, 3, \ldots , n\}$. Two nodes that are connected are called *neighbors*; the set of all neighbors of a node i is denoted by $\Gamma(i)$. The number of neighbors of a node i is called *degree* and is denoted by k_i. Two nodes i and j are said to be connected if there exists at least one sequence of connections that begins at i, traverses from node to node through the connections of the graph, and ends at *j*. Such sequences of connections are called *routes* or *paths* and the number of connections traversed is the length of the route.

2.2 Structural Characterization

For the purpose of analyzing the structure and behavior of complex systems modeled as graphs, numerous characterization functions have been proposed [10]. There are two main types of these functions: those based on global information that require information on the entire graph simultaneously and those based on local information that only access the information of a certain node i and its neighborhood Γ(i) at a time. In this section we review the local structural characterization functions related with this work.

2.2.1 Degree Distribution

The degree k_i of a node *i* is a local measure of network. $P(k)$ denotes the number of nodes that have degree *k*, normalized by *n*. The measure of $P(k)$ can be interpreted as the probability that a randomly chosen node *i* has degree *k*. The values of $P(k)$ for $k \in [0, n\text{-}1]$ (supposing that there can only be at most one connection between each pair of distinct nodes) form the *degree distribution* of the graph. Whereas the degrees themselves are local properties, obtaining the degree distribution is a global computation.

The degree distribution is widely used to classify networks according to the *generation models* that produce such distributions. Among the first and most famous generation models are the *uniform random graphs* of Erdös and Rényi [15] and Gilbert [17] that yield a binomial distribution that at the limit, approaches the Poisson distribution and most of the nodes in the graph have similar degrees [9].

In the past decade, another type of generation models became popular as various studies revealed that the degree distribution of some important real-world networks (including the WWW, the Internet, biological and social systems) was not Poisson distribution at all, but rather a power-law distribution [1, 2, 16], $P(k) \sim k^{-\gamma}$ with values of γ typically ranging between two and three. The models that produce such distributions are called *scale-free* network models. The notable structural property in networks with powerlaw distribution is the presence of a small set of extremely well-connected nodes that are called *hubs*, whereas a great majority of the nodes has a very low degree [6, 28]. This property translates into high fault tolerance under random flaws, but high vulnerability under deliberate attack [2].

2.2.2 Parameter Tuning and Adaptation

Metaheuristics offer solutions that are often close to the optimum, but with a reasonable amount of resources used when compared to an exact algorithm. Unfortunately, the metaheuristics are usually rich in parameters. The choice of the values for the parameters is nontrivial and in many cases the parameters should vary during the runtime of the algorithm [8, 19].

The process of selecting the parameter values is known as *tuning*. The goal of offline tuning is to provide a static initial parameter configuration to be used throughout the execution of the algorithm, whereas online tuning, also known as *parameter control* or *adaptation*, is the process of adjusting the parameter values at runtime. We use a discrete model for adaptation based on the proposed by Holland in 1992 [22]. We assume that the system takes actions at discrete steps $t = 1, 2, 3, \ldots$, as this assumption applies to practically all computational System. The proposed model is described in section four.

3 SQRP Search Strategies

In this section we present the problem focused in this work. First, we describe the semantic query routing problem (SQRP) as a search process. Then, strategies for solve SQRP are shown including our proposed algorithm which uses an adaptive strategy for adjusting an important parameter for the search process: TTL.

3.1 SQRP Description

SQRP is the problem of locating information in a network based on a query formed by keywords. The goal in SQRP is to determine shorter routes from a node that issues a query to those nodes of the network that can appropriately answer the query by providing the requested information. Each query traverses the network, moving from the initiating node to a neighboring node and then to a neighbor of a neighbor and so forth, until it locates the requested resource or gives up in its absence. Due to the complexity of the problem [3, 24, 25, 33, 34, 35], solutions proposed to SQRP typically limit to special cases.

The general strategies of SQRP algorithms are the following. Each node maintains a local database of documents r_i called the *repository*. The search mechanism is based on nodes sending messages to the neighboring nodes to query the contents of their repositories. The *queries* q_i are messages that contain keywords that describe for possible matches. If this examination produces results to the query, the node responds by creating another message informing the node that launched the query of the resources available in the responding node. If there are no results or there are too few results, the node that received the query forwards it to one or more of its neighbors. This process is repeated until some predefined stopping criteria is reached. An important observation is that in a P2P network the connection pattern varies among the net (*heterogeneous topology*), moreover the connections may change in time, and this may alter the routes available for messages to take.

3.2 SQRP Algorithms

The most popular technique for searching in P2P systems is flooding, where each message is assigned a positive integer parameter known as the *time-to-live* (TTL) of the message. As the message propagates from one node to another, the value of TTL is decreased by one by each forwarding node. When TTL reaches zero, the message will be discarded and no longer propagated in the system. The main disadvantage of flooding is the rapid congestion of the communication channels [20]. Another widely used search strategy is the *random walk* [3]. A random walk in a graph is a route where the node following the initiating node is chosen uniformly at random among its neighbors.

3.2.1 AntSearch

Wu et al. [34] propose an algorithm called *AntSearch.* The main idea in the AntSearch algorithm is using pheromone values to identify the free-riders, prevent sending messages to those peers in order to reduce the amount of redundant messages the estimation of a proper TTL value for a query flooding is based on the popularity of the resources. Wu et al. use three metrics to measure the performance of the AntSearch. One is the *number of searched files* for a query with a required number of results, *R*: a good search algorithm should retrieve the number of results over but close to *R*. The second one is the *cost per result* that defines the total amount of query messages divided by the number of searched results; this metric measure how many average query messages are generated to gain a result. Finally, *search latency* is defined as the total time taken by the algorithm.

3.2.2 SemAnt

Algorithms that incorporate information on past search performance include the SemAnt algorithm [25, 26] that uses an ant-colony system to solve SQRP in a P2P environment. SemAnt seeks to optimize the response to a certain query according to the popularity of the keywords used in the query. The algorithm takes into account network parameters such as bandwidth and latency. In SemAnt, the queries are the ants that operate in parallel and place pheromone on successful search routes. This pheromone evaporates over time to gradually eliminate old or obsolete information. Also Michlmayr [25] considers parameter tuning for the SemAnt algorithm, including manual adjustment of the TTL parameter from a set of possible values {15, 20, 25, 30, 35} and concludes that 25 is the best value for the parameter. The adjustment of the TTL is made without simultaneous tuning of the other parameters.

3.2.3 Neighboring-Ant Search

NAS [11] is also an ant-colony system, but incorporates a local structural measure to guide the ants towards nodes that have better connectivity. The algorithm has three main phases: an evaluation phase that examines the local repository and incorporates the classical lookahead technique [27], a transition phase in which the query propagates in the network until its TTL is reached, and a retrieval phase in which the pheromone tables are updated.

Most relevant aspects of former works have been incorporated into the proposed NAS algorithm. The framework of AntNet algorithm is modified to correspond to the problem conditions: in AntNet the final addresses are known, while NAS algorithm does not has a priori knowledge of where the resources are located. On the other hand, differently to AntSearch, the SemAnt algorithm and NAS are focused on the same problem conditions, and both use algorithms based on AntNet algorithm. However, the difference between the SemAnt and NAS is that SemAnt only learns from past experience, whereas NAS takes advantage of the local environment. This means that the search in NAS takes place in terms of the classic local exploration method of Lookahead [27], the local structural metric DDC[29] its measures the differences between the degree of a node and the degree of its neighbors, and three local functions of the past algorithm performance.

3.2.4 Adaptative Neighboring-Ant Search

The proposed algorithm in this work, *Adaptive Neighboring Ant Search* (AdaNAS) is largely based on the NAS algorithm, but includes the adaptation of the TTL parameter at runtime, in addition to other changes. The mechanism that may extend the TTL for an ant is called the *survival rule*. It incorporates information on past queries relying on the learning strategies included in AdaNAS, basic characteristics of SQRP and a set of parameters that are adjusted according to the results found when using the *survival rule* itself. The rule evaluates the length of the shortest known route that begins with the connection (i, j) from the current node i to a node that contains good results for the query t. The form in which the algorithm operates is explained in detail later in Sections 4 and 5.

4 AdaNAS Model

In this section, we present a multi-agent model in order to describe the adaptive behavior of AdaNAS. We begin by formulating it as an adaptive system in the notation presented in Section 2.2.2.

4.1 The General Model

The environment is the P2P network, in which two stimuli are observed: the occurrences of the documents being searched (I_1) and the degree k_i of the node i (I_2). The environment has the following order to send stimuli: observing I_1 has a higher priority than observing I_2. AdaNAS is an ant-colony system, where each ant is modeled as an agent. AdaNAS has four agent types:

- The *query ant* is accountable for attending the users' queries and creating the *Forward Ant*; moreover it updates the pheromone table by means of evaporation. There is a *query ant* for each node in the net and it stays there while the algorithm is running.
- The *Forward Ant* uses the learning strategies for steering the query and when it finds resources creates the *backward ant*. It finishes the routing process when its TTL is zero or the amount of found resources is enough that is denoted by R then, it creates an *update ant*.
- The *backward ant* is responsible for informing to *query ant* the amount of resources in a node found by the *Forward Ant*. In addition, it updates the values of some learning structures that are the bases of the *survival rule* which will be explaining later (Section 4.2.3).
- The *update ant* drops pheromone on the nodes of the path generated by the *Forward Ant*. The amount of pheromone deposited depends on quantity of found resources (*hits*) and number of edges traveled (*hops*) by the *Forward Ant*.

The system is subdivided into four parts: the structures A to adapt to the environment (called *agents*), the adaptation plan P, the memory M, and the operators O. Typically, A has various alternative states A_1, A_2, A_3, . . . among which one is to be chosen for the system, according to the observations made on the environment. On the other hand, P is typically a set of rules, one or more which can be applied. These rules apply the operations in the set O. An operator is either a deterministic function, denoted as $(A_i, P_j) \rightarrow Ak$, or a stochastic function to a probability distribution over a set of states for selecting A_k. The memory M permits the system to collect information on the condition of the environment and the system itself, to use it as a base for the decision making. The observations of the environment are taken as stimuli that trigger the operators.

The routing process implemented in the *Forward Ant* is required to be adaptive, thus A is defined in function of this agent. The possible states for A are five:

A_1: No route has been assigned and the *Forward Ant* is at the initial node. The ant can be only activated when the *query ant* send it a query and can only receive once time each stimulus.

A_2: A route has been assigned and TTL has not reached zero.

A_3: TTL is zero.

A_4: *Forward Ant* used the *survival rule* to extend TTL.

A_5 = Terminal state is reached by the *Forward Ant.*
X:

The Figure 1 shows the AdaNAS adaptive model. According to the stimuli -the number of documents found (dotted line, I_1) and degree of the node (solid line, I_2)- an operator is selected. The line style for state transitions follows that of the stimuli: dotted line for transitions proceeding from I_1 and solid for I_2.

The memory M is basically composed of four structures that store information about previous queries. The first of these structures is the three dimensional pheromone table τ. The element $\tau_{i;j;t}$ is the preference for moving from node i to a neighboring node j when searching by a keyword t. In this work, we assume that each query contains one keyword and the total number of keywords (or *concepts*) known to the system is denoted by C.

The pheromone table $M_1 = \tau$ is split into n bi-dimensional tables, $\tau_{j;t}$, one for each node. These tables only contain the entries $\tau_{j;t}$ for a fixed node i and hence have at most dimensions C x $|\Gamma(i)|$. The other three structures are also three-dimensional tables $M_2 = D$, $M_3 = N$ and $M_4 = H$, each splits into n local bi-dimensional tables in the same manner. The information in these structures is of the following kind: currently being at node i and searching for t, there is a route of distance $D_{i;j;t}$ starting at the neighbor j that leads to a node identified in $N_{i;j;t}$ that contains $H_{i;j;t}$ hits or matching documents.

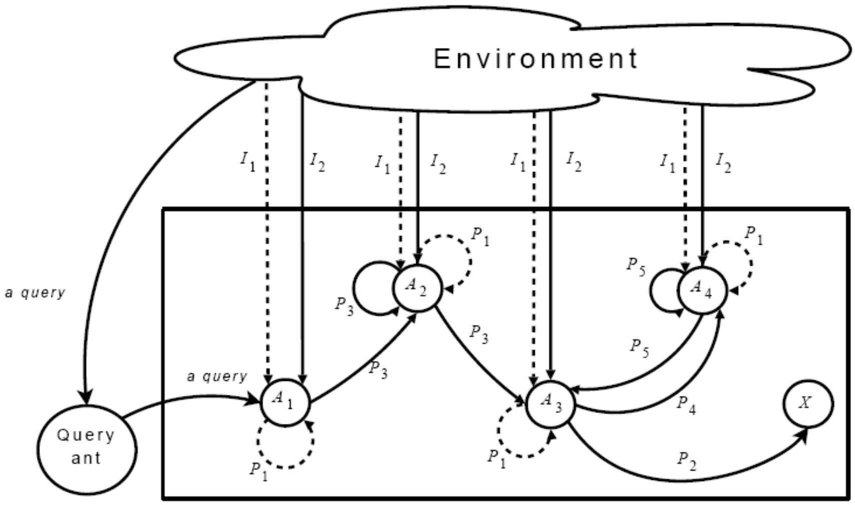


Fig. 1. AdaNAS Adaptive Model General

The adaptive plans P are the following:

P_1: **The *backward ant.*** Created when a resource is found; modifies M_2, M_3 and M_4.

P_2: **The *update ant.*** Modifies the pheromone table $M_1 = \tau$ when the *Forward Ant* reached 0 and *survival rule* can not to proceed.

P_3: **The *transition rule.*** Selects the next node applying the inherent learning stored in pheromone trails and in the memory structure $M_2 = D$.

P_4: **The *survival rule.*** Proceeds when the learning stored in M_2, M_3 and M_4 permits to extend TTL and determines how much TTL must be extended.

P_5: **The *modified transition rule.*** A variation of transition rule that eliminates the pheromone and degree effects.

The operators O of AdaNAS are the following:

O_1. $(A_1, I_1)\rightarrow A_1$: Documents are found and a *backward ant* (P_1) updates the memory -no change in the state of the system-.

O_2. $(A_1, I_2)\rightarrow A_2$: An ant selects the route according to the transition rule (P_3).

O_3. $(A_2, I_1)\rightarrow A_2$: Similar to O_1.

O_4.$(A_2,I_2)\rightarrow\{(p_{2;2;2},A_2),\ (p_{2;3;2},A_3)\ \}$: The transition rule ($P_3$) either keeps the ant in the same state with probability $p_{2;2;2}$ or moves to state A_3 with probability $p_{2;3;2}$.

O_5.$(A_3, I_1)\rightarrow A_3$ Similar to O_1.

O_6.$(A_3,I_2)\rightarrow\{(p_{3;4;2},A_4), (p_{3;X;2},X)\}$: The survival rule ($P_4$) and update ant ($P_2$) –

with probability $p_{3;4;2}$, the ant extends its TTL using P_4, or -with probability $p_{3;X;2}$, applying P_2 the ant reach its terminal state.

O_7. $(A_4, I_1) \rightarrow A_4$: Similar to O_1.

O_8.$(A_4,I_2) \rightarrow \{(p_{3;3;2},A_3), (p_{3;4;2},A_4)\}$: The modified transition rule (P_5) -the ant either stays in the same state or moves to state A_3-.

The general model is illustrated in Figure 1 where can be observed the transitions among states of the *Forward Ant*.

4.2 Behavior Rule

An ant-colony algorithm has rules that determine its behavior. These rules define why the ants construct and evaluate the solution and why the pheromone is updated and used. Although the pheromone is the main learning structure, AdaNAS has three more: *D*, *N* and *H*, for know the distances toward the nodes that contain in its repository matching documents. AdaNAS own several behavior rules: the *transition rule*, *the update rules*, the *survival rule* and the *modified transition rule*.

4.2.1 Transition Rule

The transition rule P_3 considers two structures to determine the next state: τ and *D*. The transition rule for an ant *x* that is searching by keyword *t* and is in the node *r* is the following, Equation 1:

$$\ell(x,r,t) = \begin{cases} \arg\max_{i \in (\Gamma(r)/\Lambda s)} \{\psi(r,i,t)\}, si\ \ p < q \\ \mathcal{L}(\chi, r, t) \qquad\qquad \textit{otherwise;} \end{cases} \qquad (1)$$

where *p* is a pseudo-random number, *q* is a algorithm parameter that defines the probability of using of the exploitation technique, $\Gamma(r)$ is the set of neighbors nodes of *r*, Λx is the set of nodes previously visited by *x*, and Equation 2, defined by:

$$\psi(r,i,t) = (w_d \cdot \kappa(r,i) + w_i \cdot (D_{r,i,t})^{-1})^{\beta_1} \cdot (\tau_{r,i,t})^{\beta_2}, \qquad (2)$$

where W_{deg} is the parameter that defines the degree importance, W_{dist} defines the distance importance toward the nearest node with matching documents($D_{r;i;t}$), β_1 intensifies the local metrics contribution (degree and distance), β_2 intensifies pheromone contribution ($\tau_{r;i;t}$), $\kappa(r, i)$ is a normalized degree measure expressed in Equation 3:

$$\kappa_{r,i} = \frac{k_i}{\max_{j \in \Gamma(r)} \{k_j\}}, \qquad (3)$$

and $\mathcal{L}$ is the exploration technique expressed, in Equation 4:

$$\mathcal{L}(x, r, t) = f(\{p_{x,r,i,t} \mid i \in \Gamma(r)\}), \tag{4}$$

where $f(\{p_{x;r;i;t} \mid i \in \Gamma(r)\})$ is a roulette-wheel random selection function that chooses a node i depending on its probability $p_{x;r;i;t}$ which indicates the probability of the ant x for moving from r to i searching by keyword t ant it is defined in Equation 5:

$$p_{x,r,i,t} = \frac{\psi(r,i,t)}{\sum_{j \in (\Gamma(r)/\Lambda x)} \psi(r,i,t)} \tag{5}$$

The parameters in these equations are: β_1**:** local measure intensification parameter, β_2**:** pheromone intensification parameter, W_{deg}**:** weight factor that defines the importance of the degree, W_{dist}**:** weight factor that defines the importance of the distance, q**:** the relative importance of exploitation versus exploration.

The tables D and τ were described in the previous section. The exploration strategy $\mathcal{L}$ is activated when $p \geq q$ and stimulates the ants to search for new paths. In case that $p<q$, the exploitation strategy is selected: it prefers nodes that provide a greater amount of pheromone and better connectivity with smaller numbers of hops toward a resource. As is shown in the transition rule, β_2 is the intensifier of the pheromone trail, and β_1 is the intensifier of the local metrics, this means that the algorithm will be only steered by the local metrics when $\beta_2 = 0$, or by the pheromone when $\beta_1 = 0$. In this work the initial values are $\beta_1 = 2$ and $\beta_2 = 1$.

4.2.2 Update Rules

There are two basic update rules in an ant colony algorithm: the evaporation and increment of pheromone. The evaporation method of AdaNAS is based on the technique used in SemAnt [25], while the increment strategy is based on the proposed in NAS [11]. Both update rules are described below.

Pheromone Evaporation Rule, the pheromone evaporation is a strategy whose finality is avoid that the edges can take very big values of pheromone trail causing a greedy behavior on the algorithm. Each unit time the query ant makes smaller the pheromone trail of the node where the query ant is, by multiplying the trail by the evaporation rate ρ, which is a number between zero and one. To avoid very low values in the pheromone the rule incorporates a second term consisting of the product $\rho\tau_0$, where τ_0 is the initial pheromone value. The Equation 6 expresses mathematically the evaporation pheromone rule.

$$\tau_{r,s,t} \leftarrow (1-\rho)\bullet\tau_{r,s,t} + \rho\bullet\tau_0 \tag{6}$$

Pheromone Increment Rule, when a *Forward Ant* finishes, it must express its performance in terms of pheromone by means of an *update ant* whose function is to increase the quantity of pheromone depending on amount of documents found and edges traversed by *Forward Ant*. This is done each time that an *update ant* passes on one node. The Equations 7 and 8 describe the *pheromone increment rule*.

$$\tau_{r,s,t} \leftarrow \tau_{r,s,t} + \Delta\tau_{r,s,t}(x) \qquad (7)$$

where $\tau_{r;s;t}$ is the preference of going to *s* when the *Forward Ant* is in *r* and is searching by keyword *t*, $\Delta\tau_{r;s;t}(x)$ is the amount of pheromone dropped on $\tau_{r;s;t}$ by a *backward ant* generated by the *Forward Ant x* and can be expressed like:

$$\Delta\tau_{r,s,t}(x) \leftarrow \left[w_h \frac{hits(x,s)}{\mathrm{R}} + (1-w_h)\frac{1}{hops(x,r)} \right] \qquad (8)$$

where *hits*(*x, s*) is the amount of documents found by the *Forward Ant x* from *s* to end of its path, and *hops*(*x, r*) is the length of the trajectory traversed by the *Forward Ant x* from *r* to the final node in its route passing by *s*.

4.2.3 Survival Rules

As can be seen in Figure 2, P_1 (the backward ant) updates the memory structures $M_2 = D$, $M_3 = N$, and $M_4 = H$. These structures are used in the survival rule (P_4) to increase time to live. This survival rule can be only applied when TTL is zero (see Figure 2(c)). The survival rule can be expressed mathematically in terms of the structures *H*, *D* and *N* as see in Equation 9:

$$\Delta TTL(x,i,t) = \begin{cases} D_{i,\omega(x,i,t),t};si\ \Omega(x,i,t) > Z_x \\ 0;en\ otro\ caso \end{cases} \qquad (9)$$

where ΔTTL(*x, i, t*) is the increment assigned to the TTL of ant *x* (that is, number of additional steps that the ant will be allowed to take) when searching for resources that match to *t*, currently being at node *i*. The number of additional steps $D_{i;\omega(x;i;t);t}$ for arriving in the node $\omega(x, i, t)$ is determined from the shortest paths generated by previous ants, and is taken when its associated efficiency $\Omega(x, i, t)$ is better than *Zx* which is a measure of current performance of the ant *x*. The auxiliary functions are shown in Equations10 and 11:

$$\Omega(x,i,t) = \max_{j\in(\Gamma(i)/\Lambda_x)} \left\{ \frac{H_{i,j,t}}{D_{i,j,t}} \middle| N_{i,j,t} \notin \Lambda_x \right., \qquad (10)$$

$$\omega(x,i,t) = \arg\Omega(x,i,t), \qquad (11)$$

where Γ(*i*) is the set of neighbors of node *i* and Λ*x* is the set of nodes previously visited by the ant *x*. The tables of hits *H*, of distances *D*, and of nodes *N* were explained in the previous section. The function $\omega(x, i, t)$ determines which node that is neighbor of the current node *i* and that has not yet been visited has previously produced the best efficiency in serving a query on *t*, where the efficiency is measured by $\Omega(x, i, t)$.

4.2.4 Modified Transition Rule

The *modified transition rule* is a special case of *transition rule* (see Equations 4 and 5) where $\beta_2 = 0$, $W_{\text{deg}} = 0$ and $q = 1$. This rule is greedy and provokes the replication of paths generated by previous ants. This rule takes place when TTL has been extended canceling the normal *transition rule*. Mathematically can be express in Equations 12 and 13, like:

$$\ell_m(x,r,t) = \left\{\arg\max_{i\in(\Gamma(r)/\Lambda s)} \{\psi(r,i,t)\}\right. \quad (12\ )$$

where ℓm is the *modified transition rule*, r is the current node in the path, t is the searched keyword, Λx is the set of nodes visited by the *Forward Ant x* and

$$\psi(r,i,t) = (w_i \cdot (D_{r,i,t})^{-1})^{\beta_1} \quad (13\ )$$

where W_{dist} is a parameter that defines the influence of $D_{r;i;t}$ that is the needed distance for arriving in the known nearest node with documents with keyword t, from r passing by i and β_1 is the distance intensifier.

5 AdaNAS Algorithm

AdaNAS is a metaheuristic algorithm, where a set of independent agents called ants cooperate indirectly and sporadically to achieve a common goal. The algorithm has two objectives: it seeks to maximize the number of resources found by the ants and to minimize the number of steps taken by the ants. AdaNAS guides the queries toward nodes that have better connectivity using the local structural metric degree (defined in Section 2.2), in addition, it uses the well known *lookahead* technique [26], which, by means of data structures, allows to know the repository of the neighboring nodes of a specific node.

The AdaNAS algorithm performs in parallel all the queries using query ants. Each node has only a query ant, which generates a *Forward Ant* for attending only one user query, assigning the searched keyword t to the *Forward Ant*. Moreover the *query ants* realize periodically the local pheromone evaporation of the node where it is. The process done by *query ant* is represented in Algorithm 1.

In the Algorithm 2 is shown the process realized by the *Forward Ant*. As can be observed all *Forward Ants* act in parallel. In an initial phase (lines 4- 8), the ant checks the local repository, and if it founds matching documents then creates a *backward ant*. Afterwards, it realizes the search process (lines 9-25) while it has live and has not found R documents.

The search process has three sections: Evaluation of results, evaluation and application of the extension of TTL and selection of next node (lines 24-28).

```
    Algorithm 1: Query ant algorithm
1   in parallel for each query ant w located in the node r
2   while the system is running do
3     if the user queries to find R documents with keyword t then
4       create Forward Ant x(r,t,R)
5       activate x
6     End
7     apply pheromone evaporation
8   End
9   end of in parallel
```

The first section, the evaluation of results (lines 10-15) implements the classical Lookahead technique. That is, the ant *x* located in a node *r*, checks the lookahead structure, that indicates how many matching documents are in each neighbor node of *r*. This function needs three parameters: the current node (*r*), the keyword (*t*) and the set of known nodes (*known*) by the ant. The set *known* indicates what nodes the lookahead function should ignore, because their matching documents have already taken into account. If some resource is found, the *Forward Ant* creates a *backward ant* and updates the quantity of found matching documents.

The second section (lines 16-23) is evaluation and application of the extension of TTL. In this section the ant verifies if TTL reaches zero, if it is true, the ant intends to extend its life, if it can do it, it changes the normal *transition rule* modifying some parameters (line 21) in order to create the *modified transition rule*.

The third section of the search process phase is the selection of the next node. Here, the *transition rule* (normal or modified) is applied for selecting the next node and some structures are updated. The final phase occurs when the search process finishes; then, the *Forward Ant* creates an *update ant* for doing the pheromone update.

The Algorithm 3 presents the parallel behavior for each *backward ant* which inversely traverses the path given by the *Forward Ant*. In each node that it visits, it tries to update the structures *D*, *H* and *N*, which will be used for future queries (lines 7-11). The update is realized if the new values point to a nearer node (line 7). After that, it informs to *ant query* of the initial node of the path how many documents the *Forward Ant* found and which path used (line 13).

The Algorithm 4 presents the concurrent behavior for each *update ant* which inversely traverses the path given by the *Forward Ant*. In each node that it visits, it updates the pheromone trail using the Equation 6. (line 5)

```
    Algorithm 2: Forward ant algorithm
1   in parallel for each Forward Ant x(r,t,R)
2   initialization: TTL = TTLmax, hops= 0
3   initialization: path=r, Λ=r, known=r
4   Results= get_ local_ documents(r)
5   if results > 0 then
6      create backward ant y(path, results, t)
7      activate y
8   End
```

```
while $TTL < 0$ and $results < R$ do
   La_results= look ahead(r,t,known)
   if $la\ results > 0$ then
      create backward ant y(path, la results, t)
      activate y
      results   results + la results
   End
   if $TTL > 0$ then
      $TTL = TTL - 1$
   Else
      if (results < R) and ( $\Delta TTL(x, results, hops) > 0$) then
         $TTL = TTL + \Delta TTL(x, results, hops)$
         change parameters: $q = 1$, $W$deg =0, $\beta 2 = 0$
      End
   End
Hops= hops + 1
Known= known $\cup$[ ( $r \cup \Gamma(r)$))
$\Lambda = \Lambda \cup r$
$r = \ell(x,r,t)$
add to path(r)
End
create update ant z(x, path, t)
activate z
kill x
end of in parallel
```

6 Experiments

In this section, we describe the experiments we carried during the comparisons of the AdaNAS and NAS algorithms.

6.1 Generation of the Test Data

A SQRP instance is formed by three separate files: topology, repositories, and queries. We generated the experimental instances following largely those of NAS reported by Cruz et al. [11] in order to achieve comparable results. The structure of the environment in which is carried out the process described is called *topology*, and refers to the pattern of connections that form the nodes on the network. The generation of the topology (*T*) was based on the method of Barabási et al. [7] to create a scale-free network. We created topologies with 1, 024 nodes; the number of nodes was selected based on recommendations in the literature [5, 25].

```
Algorithm 3: Backward ant algorithm
initialization: hops= 0
in parallel for each backward ant y(path, results, t)
for $i = |path| - 1$ to 1 do
   $r = path_{(i-1)}$
```

```
    $s = path_i$
    $hops = hops + 1$
    **if** $D_{r;s;t} > hops$ **then**
      $D_{r;s;t} = hops$
      $H_{r;s;t} = result$
      $N_{r;s;t} = path_h$
    **End**
**End**
**send** (*results, path*) to the query ant located in *path*1
**kill** y
**end of in parallel**
```

```
**Algorithm 4:** Update ant algorithm
**in parallel for each** *update ant* z(*path, t, x*)
**for** $i = |path| - 1$ *to* 1 **do**
    $r = path_{(i-1)}$
    $s = path_i$
    $\tau_{r;s;t} = \tau r;s;t + \Delta\tau_{r;s;t}(x)$
**End**
**kill** z
**end of in parallel**
```

The *local repository* (*R*) of each node was generated using "topics" obtained from ACM Computing Classification System taxonomy (ACMCCS). This database contains a total of 910 distinct topics. Also the content are scale-free: the nodes contain many documents in their repositories on the same topic (identified by keywords) and only few documents on other topics.

For the generation of the *queries* (*Q*), each node was assigned a list of possible topics to search. This list is limited by the total amount of topics of the ACMCCS. During each step of the experiment, each node has a probability of 0.1 to launch a query, selecting the topic uniformly at random within the list of possible topics of the node repository. The probability distribution of *Q* determines how often the query will be repeated in the network. When the distribution is uniform, each query is duplicated 100 times in average.

Table 1. Parameter configuration of the NAS algorithm.

PARAMETER	VALUE	DEFINITION
α	0.07	Global pheromone evaporation factor
ρ	0.07	Local pheromone evaporation factor
β	2	Intensifier of pheromone trail
τ_0	0.009	Pheromone table initialization
q_0	0.9	Relative importance between exploration and exploitation
R	10	Maximum number of results to retrieve
TTL_{max}	10	Initial TTL of the Forward Ants
W	0.5	Relative importance of the resources found and TTL

Table 2. Parameter configuration of the AdaNAS algorithm.

PARAMETER	VALUE	DEFINITION
ρ	0.07	Local pheromone evaporation factor
B_1	2	Intensification of local measurements (degree and distance) in transition rule.
B_2	1	Intensification of pheromone trail in the in the transition rule.
τ_0	0.009	Pheromone table initialization
q	0.9	Relative importance between exploration and Exploitation in the transition rule.
R	10	Maximum number of results to retrieve
TTL_{max}	10	Initial TTL of the Forward Ants
w_h	0.5	Relative importance of the hits and hops in the increment rule
w_{deg}	1	Degree's influence in the transition rule
w_{dist}	1	Distance's influence in the transition rule

The topology and the repositories were created static, whereas the queries were launched randomly during the simulation. Each simulation was run for 15,000 queries during 500 time units, each unit has 100ms. The average performance was studied by computing three performance measures of each 100 queries:

- **Average hops,** defined as the average amount of links traveled by a Forward Ant until its death, that is, reaching either the maximum amount of results required *R* or running out of TTL.
- **Average hits,** defined as the average number of resources found by each Forward Ant until its death.
- **Average efficiency,** defined as the average of resources found per traversed edge (hits/hops).

6.2 Parameters

The configuration of the algorithms used in the experimentation is shown in Tables 1 and 2. The first column is the parameter, the second column is the parameter value and the third column is a description of the parameter. These parameter values were based on recommendations of the literature [11, 14, 25, 30, 31].

6.3 Results

The goal of the experiments was to examine the effect of the strategies incorporated in the AdaNAS algorithm and determine whether there is a significant contribution to the average efficiency. The main objective of SQRP is to find a set of paths among the nodes launching the queries and the nodes containing the resources, such that the efficiency is greater, this is, the quantity of found resources is maximized and the quantity of steps given to find the resources is minimized.

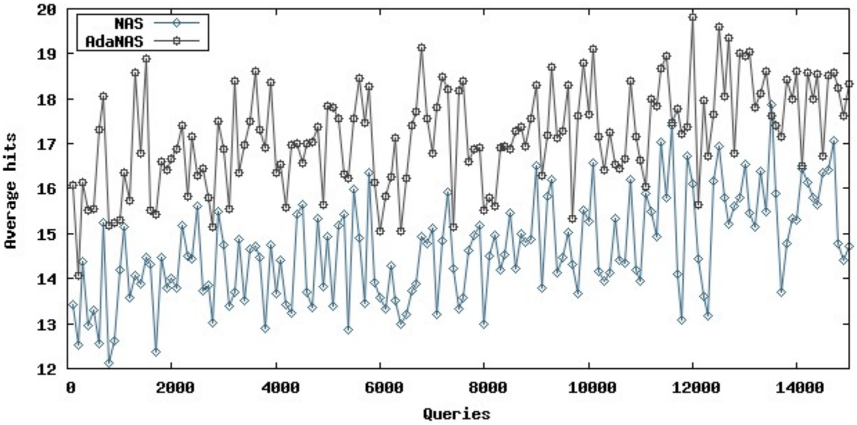


Fig. 2. Learning evolution in terms of the number of resources found for AdaNAS and NAS algorithms.

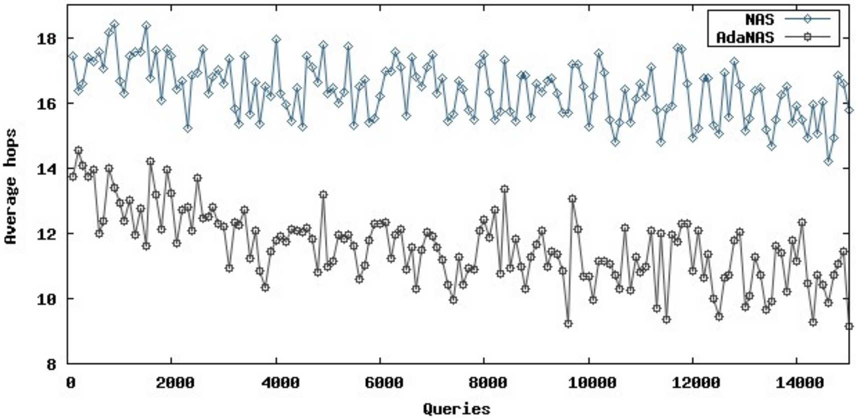


Fig. 3. Learning evolution in terms of the length of the route taken for AdaNAS and NAS algorithms.

Figure 2 shows the *average hits* performed during 15,000 queries with AdaNAS and NAS algorithms on an example instance. NAS starts off approximately at 13.4 hits per query; at the end, the average hit increases to 14.7 hits per query. For AdaNAS the average hit starts at 16 and after 15,000 queries the average hit ends at 18.3. On the other hand, Figure 3 shows the *average hops* performed during a set of queries with NAS and AdaNAS. NAS starts approximately at 17.4 hops per query; at the end, the average hops decrease to 15.7 hops per query. For AdaNAS the average hops starts at 13.7 and after 15, 000 queries the average hops ends at 9.1. Finally, Figure 4 shows the *average efficiency* performed during a set of queries. NAS starts approximately at 0.76 hits per hop; at the end, it increases to 0.93

hits per hop. For AdaNAS the average efficiency starts at 1.17 hits per hop and after 15, 000 queries the average efficiency ends at 2.

The adaptive strategies of AdaNAS show an increment of 24.5% of found documents, but the biggest contribution is a reduction of hops in 40%, giving efficiency approximately twice better on the final performance of NAS. This observation suggests that the use of degree instead of DDC was profitable. In addition, the incorporation of the survival rule permits to improve the efficiency, because it guides the Forward Ants to nodes that can satisfy the query. Moreover, in future works it will be important to study adaptive strategies for other parameters as well as the initial algorithm parameter configuration in search of further improvement in the efficiency.

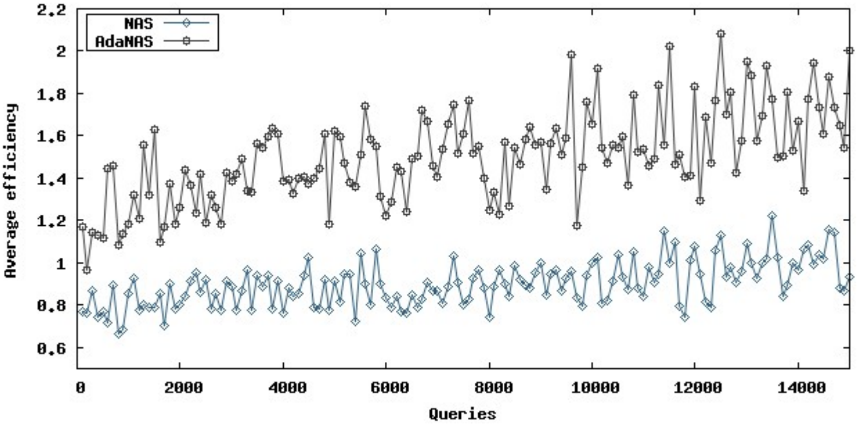


Fig. 4. Learning evolution in terms of the efficiency (hits/ hop) for AdaNAS and NAS algorithms.

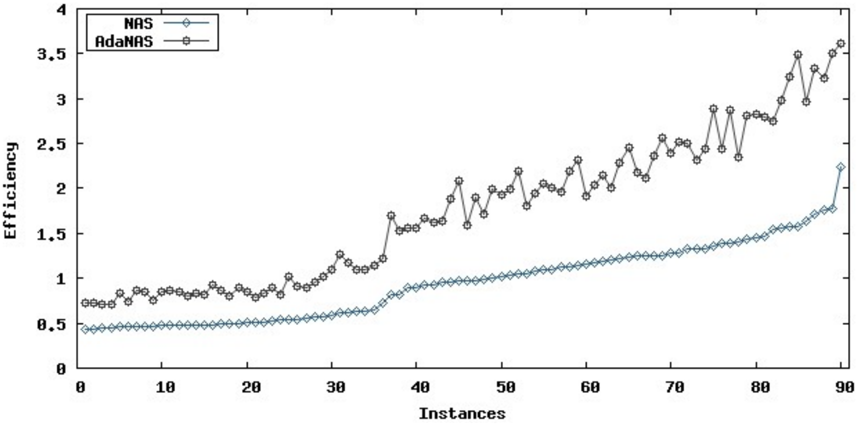


Fig. 5. Comparison between NAS and AdaNAS experimenting with 90 instances.

Figure 5 shows the results of the different experiments applied to NAS and AdaNAS on thirty runnings for each ninety different instances generated with the characteristics described in Section 6.1. It can been seen from it that on all the instances the AdaNAS algorithm outperforms NAS. On average, AdaNAS had an efficiency 81% better than NAS.

7 Conclusions

For the solution of SQRP, we proposed a novel algorithm called AdaNAS that is based on existing ant-colony algorithms. This algorithm incorporates parameters adaptive control techniques to estimate a proper TTL value for dynamic text query routing.

In addition, it incorporates local strategies that take advantage of the environment on local level, three functions were used to learn from past performance. This combination resulted in a lower hop count and an improved hit count, outperforming the NAS algorithm. Our experiments confirmed that the proposed techniques are more effective at improving search efficiency. Specifically the AdaNAS algorithm in the efficiency showed an improvement of the 81% in the performance efficiency over the NAS algorithm.

As future work, we plan to study more profoundly the relation among SQRP characteristics, the configuration of the algorithm and the local environment strategies employed in the learning curve of ant-colony algorithms, as well as their effect on the performance of hop and hit count measures.

References

[1] Adamic, L., Huberman, B.: Power-law distribution of the World Wide Web. Science 287(5461), 2115 (2000)

[2] Albert, R., Jeong, H., Barabási, A.: Error and attack tolerance of complex networks. Nature 506, 378–382 (2000)

[3] Amaral, L., Ottino, J.: Complex systems and networks: Challenges and opportunities for chemical and biological engineers. Chemical Engineering Scientist 59, 1653–1666 (2004)

[4] Androutsellis-Theotokis, S., Spinellis, D.: A survey of peer-to-peer content distribution technologies. ACM Computing Surveys 36(4), 335–371 (2004)

[5] Babaoglu, O., Meling, H., Montresor, A.: Anthill: An framework for the development of agent-based peer to peer systems. In: 22nd International Conference On Distributed Computing Systems. ACM, New York (2002)

[6] Barabási, A.: Emergence of scaling in complex networks, pp. 69–82. Wiley VHC, Chichester (2003)

[7] Barabási, A., Albert, R., Jeong, H.: Mean-field theory for scale-free random networks. Physical Review Letters 272, 173–189 (1999)

[8] Birattari, M.: The Problem of Tunning Mataheuristics as Seen From a Machine Learning Perspective. PhD thesis, Bruxelles University (2004)

[9] Bollobás, B.: Random Graphs, 2nd edn. Cambridge Studies in Advanced Mathematics, vol. 73. Cambridge University Press, Cambridge (2001)

[10] Costa, L., Rodríguez, F.A., Travieso, G., Villas, P.R.: Characterization of complex networks: A survey of measurements. Advances in Physics 56, 167–242 (2007)

[11] Cruz, L., Gómez, C., Aguirre, M., Schaeffer, S., Turrubiates, T., Ortega, R., Fraire, H.: NAS algorithm for semantic query routing systems in complex networks. In: DCAI. Advances in Soft Computing, vol. 50, pp. 284–292. Springer, Heidelberg (2008)

[12] DiCaro, G., Dorigo, M.: AntNet: Distributed stigmergy control for communications networks. Journal of Artificial Intelligence Research 9, 317–365 (1998)

[13] Diestel, R.: Graph Theory. Graduate Texts in Mathematics, vol. 173. Springer, New York (2000)

[14] Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)

[15] Erdos, P., Rényi, A.: On the evolution of random graphs, vol. 2, pp. 482–525. Akademiai Kiad´o, Budapest, Hungary, 1976. First publication in MTA Mat. Kut. Int. Kozl. (1960)

[16] Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationship on the internet topology. ACM SIGCOMM Computer Communication Review 29, 251–262 (1999)

[17] Gilbert, E.: Random graphs. Annals of Mathematical Statistics 30(4), 1141–1144 (1959)

[18] Glover, F., Kochenberger, G.: Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 57. Springer, Heidelberg (2003)

[19] Glover, F., Laguna, M.: Tabú Search. Kluwer Academic Publishers, Dordrecht (1986)

[20] Goldberg, P., Papadimitriou, C.: Reducibility among equilibrium problems. In: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, pp. 61–70. ACM, New York (2005)

[21] Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H., Zahorjan, J.: Measurement, modeling and analisys of a peer-tp-peer file-sharing workload. In: 19th ACM Symposium on Operating Systems Principles. ACM, New York (2003)

[22] Holland, J.H.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)

[23] Leibowitz, N., Ripeanu, M., Wierzbicki, A.: Deconstructing the kazaa network. In: 3rd IEEE Workshop on Internet Applications (2003)

[24] Liu, L., XiaoLong, J., Kwock, C.C.: Autonomy oriented computing — from problem solving to complex system modeling, pp. 27–54. Springer Science + Business Media Inc., Heidelberg (2005)

[25] Michlmayr, E.: Ant Algorithms for Self-Organization in Social Networks. PhD thesis, Vienna University of Technology (2007)

[26] Michlmayr, E., Pany, A., Kappel, G.: Using Taxonomies for Content-based Routing with Ants. In: Proceedings of the Workshop on Innovations in Web Infrastructure, 15th International World Wide Web Conference (WWW2006) (May 2006)

[27] Mihail, M., Saberi, A., Tetali, P.: Random walks with lookahead in power law random graphs. Internet Mathematics 3 (2004)

[28] Newman, M.E.J.: Power laws, pareto distributions and zipf's law. Contemporary Physics 46(5), 323–351 (2005)

[29] Ortega, R.: Estudio de las Propiedades Topológicas en Redes Complejas con Diferente Distribución del Grado y su Aplicación en la Búsqueda de Recursos Distribuidos. PhD thesis, Instituto Politécnico Nacional, México (2009)

[30] Ridge, E.: Design of Expirements for the Tuning of Optimization Algorithms. PhD thesis, University of York (2007)

[31] Ridge, E., Kudenko, D.: Tuning the Performance of the MMAS Heuristic in Engineering Stochastic Local Search Algorithms. In: Stützle, T., Birattari, M. (eds.) SLS 2007. LNCS, vol. 4638, pp. 46–60. Springer, Heidelberg (2007)

[32] Sakarayan, G.: A Content-Oriented Approach to Topology Evolution and Search in Peer-to-Peer Systems. PhD thesis, University of Rostock (2004)

[33] Tempich, C., Staab, S., Wranik, A.: REMINDIN: Semantic Query Routing in Peer-to-Peer Networks based on Social Metaphers. In: 13th World Wide Web Conference, WWW (2004)

[34] Wu, C.-J., Yang, K.-H., Ho: AntSearch: An ant search algorithm in unstructured peer-to-peer networks. In: ISCC, pp. 429–434 (2006)

[35] Wu, L.-S., Akavipat, R., Menczer, F.: Adaptive query routing in peer Web search. In: Proc. 14th International World Wide Web Conference, pp. 1074–1075 (2005)

Memetic Algorithm for Solving the Problem of Social Portfolio Using Outranking Model

Claudia G. Gómez S., Eduardo R. Fernández Gonzalez Laura Cruz Reyes,
S. Samantha Bastiani M, Gilberto Rivera Z., and Victoria Ruíz M.

Instituto Tecnológico de Ciudad Madero, 1ro. de Mayo y Sor Juana I. de la Cruz s/n CP.
89440, Tamaulipas, México
{cggs71,b_shulamith}@hotmail.com,
{riveragil,victoria.rzrz}@gmail.com,
lcruzreyes@prodigy.net.mx
Universidad Autónoma de Sinaloa (UAS), Justicia Social SN, Ciudad Universitaria,
81229 Los Mochis, Sinaloa. (52) 668 812 33 01.
eddyf@uas.uasnet.mx

Abstract. The government institutions at all levels, foundations with private funds or private companies that support social projects receiving public funds or budget to develop its own social projects often have to select the projects to support and allocate budget to each project. The choice is difficult when the available budget is insufficient to fund all projects or proposals whose budget requests have been received, together with the above it is expected that approved projects have a significant social impact. This problem is known as the portfolio selection problem of social projects. An important factor involved in the decision to make the best portfolio, is that the objectives set out projects that are generally intangible, such as the social, scientific and human resources training. Taking into account the above factors in this paper examines the use of multi objective methods leading to a ranking of quality of all selected projects and allocates resources according to priority ranking projects until the budget is exhausted. To verify the feasibility of ranking method for the solution of problem social portfolio constructed a population memetic evolutionary algorithm, which uses local search strategies and cross adapted to the characteristic of the problem. The experimental results show that the proposed algorithm has a competitive performance compared to similar algorithms reported in the literature and on the outranking model is a feasible option to recommend a portfolio optimum, when little information and the number of projects is between 20 and 70.

1 Introduction

Organizations are constantly making decisions about how to invest and managing their resources to satisfy social needs, as: money, time, equipment and people, among others. Usually resources are lower than demand, resulting in not being

O. Castillo et al. (Eds.): Recent Advances on Hybrid Intelligent Systems, SCI 451, pp. 335–348.
springerlink.com

able to provide benefits to all competing projects because of these organizations are forced to select the best portfolio that will consist of a subset of projects that maximizes the benefits social. One of the main tasks of the manager is to select projects that best meet the objectives of the company [12]. Incorrect decisions regarding the selection of projects have two main consequences: i) resources generally limited, are wasted on projects that, although they may be good, not the most appropriate for the company and ii) the organization loses the benefits that could have gotten if it had invested in more suitable projects.

The selection of projects for a portfolio of social projects needs special treatment for the following reasons [7]:

1. The quality of projects is generally described by multiple criteria that are often in conflict.
2. Typically, requirements are not accurately known. Many concepts have no mathematical support for having entirely subjective nature.
3. Heterogeneity, or differences between the objectives of the projects, makes it difficult to compare.

These features of the *Social Portfolio Problem* (*SPP*) represent a challenge for multi-objective optimization algorithms [8]. Moreover, although optimal solutions are found, the problem has not been completely resolved. Even the decision maker has the task of implementing just one of the alternatives presented. The decision maker will evaluate the alternatives according to her/his criteria and preferences.

In this paper we are interested in solving the Social Portfolio Problem using a set of ranked projects as input data instance and thereafter shall be adopted in preference relations using methods of categorization and outranking model. All these of these features will be included in a memetic algorithm that aims to generate the best recommendation for the decision maker.

2 Background

In order to place the research in context, this section is divided in six parts. The first part define portfolio problem and the second part explain social project. The third part defines multi-objetive optimization and the next part describes the problem formulation, finally the last part explains multi-objective evolutionary algorithm and memetic algorithm.

2.1 Portfolio Problem

A *project* is a temporary, unique and unrepeatable process which pursues a specific set of objectives [2]. In this work, it is not considered that the projects can be broken down into smaller units such as tasks or activities. In other words, a project cannot be divided to run only a part, however, different versions of the same project can be proposed, each version may vary in amount of activity, time required and requested resources.

A *portfolio* consists of a set of projects that can be performed in the same period of time [2]. For this reason, projects in the same portfolio share available resources in the organization. Thus, it is not sufficient to compare the projects individually, but must compare groups of projects to identify what portfolio makes the greatest contribution to the organization objectives.

The proper selection of projects to integrate the portfolio, which will receive the organization's resources, is one of the most important decision problems for both public and private institutions [3, 11]. The main economic and mathematical models to the portfolio problem assume that there is a defined set of *n* projects, each project well characterized with costs and revenues, of which the distribution over time is known. The *Decision Maker* (DM) is responsible for selecting the portfolio that the company will implement [7].

2.2 Social Projects

Social projects are characterized by objectives whose fulfillment benefits society. These objectives are generally intangible, such as social and scientific impact, as well as human resource training, among others, without regard to potential economic benefit as the main element of measure. In addition, the amount of desired objectives in these projects can be of several tens, depending on the level of detail and the conditions under which it is restricted.

It is also important to note that such projects are usually assigned to one area and region. The project area is mainly the social sector, e.g. education, public health, safety, scientific development, among others. The region is primarily concerned with the physical area that will benefit, for example by state, county, district, or similar. Thus, to form social portfolios should be considered [11]:

1. No area/region monopolizes most of the budget, leaving remaining areas/regions with poor resources.
2. All areas/regions receive at least a minimal budget, ensuring its permanence and growth.

2.3 Multi-objective Optimization

From the several emergent research areas in which EAs have become increasingly popular, multi-objective optimization has had one of the fastest growing in recent years. A multi-objective optimization problem (MOP) differs from a single-objective optimization problem because it contains several objectives that require optimization [4].

Real-world optimization problems are extremely complex with many attributes to evaluate and multiple objectives to optimize [4, 15]. The attributes correspond to quantitative values that describe the problem and are expressed in terms of decision variables. The objectives are the directions for improvement of the attributes and can be to maximize or minimize.

In many cases, due to the conflicting nature of attributes is not possible to obtain a single solution and therefore the ideal solution for a MOP cannot be achieved because there is no one solution the problem. Typically, solving a MOP has a set of solutions that reached an aspiration level expected by the DM [4].

Therefore they are solutions that, although different, their performance is mathematically equivalent and cannot be overcome on both objectives simultaneously without leaving the feasible solution space [9]. This set of solutions is called the Pareto front, and find it is one of the main purposes of solving a MOP [6].

But finding the Pareto front does not completely solve a MOP. Now the DM should choose a solution from the front, according to his/her own criteria. This is not a difficult task if you are managing two or three objectives. However, when the number of objectives increases, three major difficulties arise:

1. The capacity of algorithms for finding the Pareto front is rapidly degraded [23]. It becomes extremely difficult for the DM, and even impossible, to establish valid criteria for comparing solutions when there are conflicting objectives [9].
2. The size of the Pareto front can grow exponentially with respect to the number of objectives. This complicates the task of the DM to choose a solution [9].

A *compromise solution* is understood as a Pareto solution in which the objectives achieved acceptable values for the DM, and therefore could be selected. The *best compromise* is the compromise solution that meets best the preferences of the DM. Thus, the solution to a MOP is not only finding the Pareto front, but also to identify the best compromise.

Identify the Pareto front (or at least an approximation) has been commonly the task of multi-objective algorithms, leaving the identification of the best compromise to the user. However, a typical DM is capable of processing only at most five to ten pieces of information at once [14], thus being unable to identify the best compromise when he/she needs to compare sets of solutions of a MOP over five or nine objectives. To address this problem requires the creation of algorithms for MOP that show a set of solutions as small as possible, but without discarding those that the DM could choose as a final solution.

Since all Pareto solutions are mathematically equivalent, the DM should provide information about his/her preferences to MOP algorithms. Such information can be provided before or after to generate the Pareto solutions or the process can be interactive, progressively consulting DM preferences [8].

2.4 Problem Formulation

The ***portfolio selection problem*** is defined by Fernández [8, 10, 21] based on the following premises:

- Consider a set A composed by N_A competing and non-interacting projects.
- Each project is described by a set of attributes Q that specify their quality as public-policy, and each project entails a somewhat imprecise request for funds. N_A is considered to be a large number, and Q contains both tangible and intangible attributes.

- A *DM*, representing the higher-level preferences in the organization, is assumed.
- It is assumed that some multicriteria-analysis technique was previously applied to a set of competing projects. Thus, unacceptable projects have been eliminated in advance and the remaining projects are ranked in a quality-descendant ordering.
- This ranking is the input information for the problem. Let *A′* denote the set of acceptable projects and *N* = *card*(*A′*).
- The projects may be classified (e.g., regarding activity, function or geographical impact) in accordance with some demands from the *DM*.
- There is a fixed budget that intends to be fairly distributed among the projects.

The solution concept consists in establishing a subset *C* of *A′* whose components will be financed according to specific assignations. In what follows, *C* will denote the project portfolio.

In first place the *DM* should weigh the number of supported projects against quality. As stated, the quality of the projects is comprised in the ranking. The conventional way for allocating resources according to ranks arises from the necessity of respecting the information on the projects' quality. However, a rank contains imprecise information. A ranking is basically qualitative and depends on the multi-criteria evaluation method that is applied for its construction.

2.5 Multi-Objective Evolutionary Algorithms

MOEA have become a popular technique for solving multi-objective problems [4]. Thus, using MOEA, the DM does not need to do a series of optimizations for each objective, as is usually done in the methods of operations research [22, 18].

The objective of a *Multi-Objective Evolutionary Algorithms* (MOEA) is to converge to the true Pareto front of a problem which normally consists of a diverse set of points. MOPs can present an uncountable set of solutions, which when evaluated produce vectors whose components represent trade-offs in decision space [4].

However, a limitation on the MOEA is the fact that only involves the process of finding a solution set without considering the most important aspect, the decision process. Most current approaches to MOEA are focused on finding an approximation to the optimal set of Pareto front, however, identify the best compromise has usually been omitted.

2.6 Memetic Algorithm (MAs)

The method is based on a population of agents and proved to be of practical success in a variety of problem domains and in particular for the approximate solution of NP-hard optimization problems [17].

An important characteristic is the use of a meme, suggests that in cultural evolution processes, information is not simply transmitted unaltered between individuals. This enhancement is accomplished in MAs by incorporating heuristics, approximation algorithms, local search techniques, specialized recombination operators, truncated exact methods, etc. In essence, most MAs can be interpreted as a search strategy in which a population of optimizing agents cooperate and compete. The success of MAs can probably be explained as being a direct consequence of the *synergy* of the different search approaches they incorporate [17]. The Figure 1 shows the basic template of MAs.

```
0 Memetic Algorithm ()
1 Population←RandomGeneratepopulation( );
2 ImprovePopulation←Local Search (Population);
3 while (not stop condition)
4       if stagnation then
5 Population←Select best individual (Population);
6 Population←Random Generate Population( );//diversification
7 ImprovePopulation←(Population);
8       end_if
9       for i←1..#crossoversdo
10        select $\pi_a$, $\pi_b$ from Population( );
11        offsprings←crossover( );
12      end_for
13      Population←SortPopulation(Population);
14      Best_individual←Select_best(SortedPopulation)
15 end_while;
16 end_Memetic Algorithm
```

Fig. 1. Template basic of the MAs

3 Proposed Algorithm

This section describes in detail the *Memetic Algorithm* (*MAs*) used for SPP and the method to evaluate the solutions in according to the model proposed. The *MAs* are evolutionary algorithms that are intimately coupled with local search algorithms, resulting in a population-based algorithm at effectively searches in the space of local optimal.

The Figure 2 shows the proposed *MAs* that we used in our implementation. In the **line 1**, a portfolio initial is obtained, the first individual is generated by following the order provided by the ranking; it is assumed a distribution of the budget among the projects. Then, the rest of population is randomly built; we experimented with a population of *200* individuals.

```
Memetic Algorithm ( )
1   Genes←Generate_ population ( );
2   do
3       EvaluatePopulation (Genes);
4       //Selection of parents per tournament
5           for i=0 to i<population
6                   Candidate1←randomPosition(Genes)
7                   Candidate2←randomPosition(Genes)
8                   while (candidato1 != candidato2);
9                           if (netFlow_calculation(Candidate1)
                            <netFlow_calculation(Candidate2))
10                              parents[i]←Candidate1;
11                          else
12                              parents[i+1]←Candidate2;
13                          end_if
14                  end_while
15          end_for
16      //Crossover
17          for i=0 to i<population
18              Cross_point←randomPosition(n_projects);
19              for j=0 to i<Cross_point
20                  temp[i]←parents[i];
21              end_for
22              for j=Cross_point to i<n_projects
23                  temp[i]← parents[i+1];
24              end_for
25              for int j=0 to i<n_projects
26                  offspring[i]←temp[i];
27              end_for
28          end_for
29      LocalSearch( );
30      EvaluatePopulation( );
31  end_while // until reach the set number of generations
```

Fig. 2. Memetic algorithm propose

From **line 2 to 31**,in each iteration (generation) the memetic procedure is shown*,* this algorithm is composed by a set of operators, evaluating population, parent's selection, crossover and local search, these operators are described below. In the **line 3** the population is evaluated, this means that the current values maximums local and global of the solutions are calculated. Then, from **line 4** ***to 15***, the parent's selection is performed per tournament, each individual from the population are evaluate to determine the best candidate in according with the *net flow calculation*. Soon, from the **line 16 to 28** the crossover operator is performed by a random point and exchanging the two individuals selected before, so are generated new individuals. In the **line 29**, the local search is applied to a solution generated by the genetic operators; once again an evaluation of population is performed.

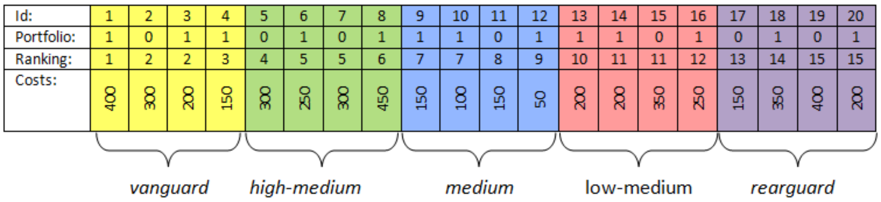
To determine the *calculation of net flow* [11], consider a comparison between solutions composing a set *E*, let *x, y* be members of E. By analogy with outranking methods [17], then may be defined the affirmation "x outranks y" as: $\sigma(x,y) = \Sigma w_i$, where the sum is carried out upon the indices of the criteria where "x is at least as preferred as y" holds, then an outranking net flow measure is associated to every $a \in E$, as follows: $fn\ (a)\ \Sigma\ \sigma\ (a,x)\ -\ \Sigma\ \sigma(x,a)$, in which the sum involves every $x \in E$, $x \neq a$. The *outranking net flow* is considered as a quality criterion for every solution.

The Figure 3 describes the procedure to compare the solutions in the model proposed [21]; this validation is given by a net flow calculation and each solution is characterized by three attributes (strong disagreements, weak disagreements and cardinality of the portfolio).

```
Net Flow Calculation ( )
1       Categorization of projects ( );
2       Calculating preference relations ( );
3       Calculation superiority ( );
4       Calculation weak disagreements ( );
5       Calculation strong disagreements ( );
6     return net flow
```

Fig. 3. Outranking model procedures

To illustrate the procedure of calculations shown in the Figure 4, first suppose a normalized weight *w* is assigned by the DM to each attribute, acting as a measure of its importance. In step one performs the categorization of projects, from a given instance of ranked projects; this categorization is given by 5 divisions, which are vanguard, high-medium, medium, low-medium and rearguard as shows the example in the Figure 4.



Id:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Portfolio:	1	0	1	1	0	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1
Ranking:	1	2	2	3	4	5	5	6	7	7	8	9	10	11	11	12	13	14	15	15
Costs:	400	300	200	150	300	250	300	450	150	100	150	50	200	200	350	250	150	350	400	200

vanguard
high-medium
medium
low-medium
rearguard

Fig. 4. Example of the categorization of the initial portfolio

Then a calculation of preference relation is executed, and the possible values that can make this calculation are: absolute, strict, weak and indifferent. These are shown in the Table 1, which are calculated to institute a relation of superiority between projects, considering the veto and concordance conditions as shown in the Table 2 the thresholds *V1, V2* and *U1* are attributes which express cost.

The relation of superiority is given formally considering (a,b) projects such that $b \in$ Portfolio and $a \notin$ Portfolio, then a relation of superiority indicates that aS'b if and only if $(a >> b) \vee (a > b) \vee (a > \sim b)$ and there are no veto conditions. Finally to generate the net flow calculation is necessary to calculate the weak and strong disagreements in addition with the cardinality of portfolio.

The disagreement is calculated by the next rules: Let *C* a set of projects which integrate the portfolio, formally a disagreement is calculated by: $D = \{ (a,b) \in A \; x \; A \mid aS'b, \; b \in C \; y \; a \notin C \}$; To calculate weak disagreements must be considered the relation of weak preference as follows: $D_d = \{(a,b) \in AxA \mid a > \sim b, \; costo(b) + V1 > costo(a) > costo(b) + U1, b \in C \; y \; a \notin C\}$ The strong disagreements are composed by the elements which are not consider in weak disagreements, that is considering the relations absolute, strict and indifferent.

Table 1. Preference relations

Notation	Preference
a>>b	*Absolute*
a>b	*Strict*
a>~b	Weak
a~b	Indifference

Table 2. Concordance and veto conditions

Concordance Conditions	Veto Conditions
a>>b	No veto
a>b	Cost(*a*) >> Cost(*b*) + *V1*
a>~b	Cost(*a*) > Cost(*b*) + *V2*
A~b	Cost(*a*) > Cost(*b*) + *U1*

4 Experimental Result

In this section, we describe experiments carried out on the memetic algorithm. The objective of the first experiments is study the performance of memetic algorithm in comparison with exact algorithm, and the second experiment memetic algorithm was compared against a genetic algorithm state of the art proposed in the [10].

4.1 Experimental Environment

The following configuration corresponds to the experimental conditions that are common to the tests described in this work:

1. **Software.** Operating system, Windows 7; programming language Java; compiler,
2. **Hardware.** Computer equipment dual-processor Xeon (TM) CPU 3.06 GHz in parallel and 4 GB RAM.

3. **Instances.** The 5 instances used for this study are randomly generated with twenty projects each; in addition we use other instance reported in [10].
4. **Performance measurement.** Performance is measured through the cardinality of the portfolio, which indicates the maximum number of projects included and the cost of the best portfolio found.

4.2 Algorithm Performance

The purpose of this section is to verify the quality of the solutions obtained by memetic Algorithm, which was implemented to solve the SPP. In our implementation, an SPP instance is formed by three attributes: Id, the budget of projects, and the ranking of the projects.

We generated the experimental instances as follows: we used 5 instances of size 20 that were generated randomly, finding optimal solutions for each of the instances and two instances of size 25 and 40 taken from work [10]. The experimentation with MAs was carried out in a total of 20 times and the optimal portfolio is selected as the best solution found.

For each instance were executed 100 generations as a limit. Table 3 shows the results obtained with the instances Opt_o9p200 until Opt_o9p205, which are the random instances. These results show that the algorithm has good performance since it achieves the number of optimal project portfolio that is 11, which was calculated with an exact algorithm, another important factor is the amount of money required to be applied in the optimal portfolio, observing that the difference of selected projects amounts vary with a small difference.

The table 4 shows the results of the instances Opt_o9p200 until Opt_o9p205 resolved with the exact algorithm, finding the optimal number of projects that can be financed with $ 80,000.00 is 11, show that both algorithms report the same number of projects that satisfy the optimal portfolio.

Continuing the analysis of results is observed that the cost of the best portfolios found is nearly equal. Taking into account the observations mentioned above it is concluded that the MAs that includes: the model of outranking and ranked instances, it is a good option to be applied on the recommendation of optimal portfolios.

Table 3. Experimental Result of the MAs (random instance)

Inst.	Cost Best Portfolio	MEMETIC RESULT Generations = 100; Population=200; Projects= 25; Amount = 1200; V1=30; V2=20; U1=10 Projects																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
20.0	79860	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1	1	0	1	1	1
20.1	79945	0	0	1	0	1	0	1	0	1	0	1	1	0	1	1	0	0	1	1	1
20.2	77920	1	0	1	1	0	1	0	0	0	1	1	1	1	1	0	1	0	1	0	0
20.3	78175	0	1	1	0	0	0	1	1	1	0	1	1	1	0	1	0	1	0	1	0
20.4	79975	1	1	0	1	0	1	0	0	1	1	1	0	1	0	0	1	0	1	0	1

Table 4. Experimental Result of the Exact Algorithm (random instance)

Inst.	Cost Best Portfo-lio	EXACT ALGORITHM RESULT Projects																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
20.0	79985	0	1	0	1	1	0	1	0	1	0	0	1	1	0	1	0	0	1	1	1
20.1	76795	1	0	1	0	1	0	0	1	1	1	0	1	0	1	0	0	0	1	1	0
20.2	78255	0	0	1	1	0	1	0	1	1	0	0	0	1	1	1	1	0	1	0	1
20.3	79115	0	1	1	1	0	0	1	1	1	0	1	0	1	0	1	1	0	0	1	1
20.4	79520	1	1	1	1	0	0	1	1	0	0	1	1	1	0	1	0	0	0	0	1

Table 5. Comparison of the memetic algorithm, against genetic algorithm to form the optimal portfolio with 25 projects

Memetic Algorithm (a)Vs Genetic Algorithm Result (b)																											
(a) Generations = 100; Population=200; Projects= 25; Amount = 1200; V1=30; V2=20; U1=10 (b) Generations = 200; Population=200; Projects= 25; Amount = 1200; V1=30; V2=20; U1=10																											
ALGORITHM	No. Project Best Port folio	Cost Best Port folio	**Projects**																								
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
MAs	17	1180	1	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0
GA	18	1190	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0

In the Table 5 show the results of the instances that were used in the work of [10]. Observing the comparison of best results for the two algorithms with the instance of 25 projects. Both algorithms start with a portfolio that includes only 11 projects, and spends a budget of $ 1180. After the execution, the genetic algorithm finds 18 projects in its portfolio optimum, unlike the memetic algorithm that finds 17 projects in its portfolio optimal. In analyzing the two optimal portfolios shows that the genetic algorithm eliminates the two most expensive projects, unlike the memetic algorithm that eliminates five projects not so expensive and optimal portfolio include eleven projects.

In the Table 6 and 7 show the results of the instances that were used in the work of [13]. Observing the comparison of best results for the two algorithms with the instance of 40 projects. Both algorithms start with a portfolio that includes only 24 projects, and spends a budget of $ 4982. After the execution, the genetic algorithm finds 27 projects in its portfolio optimum, unlike the memetic algorithm that finds 26 projects in its portfolio optimal.

Table 6. Comparison of the memetic algorithm, against genetic algorithm to form the optimal portfolio with 40 projects

Memetic Algorithm (a)Vs Genetic Algorithm Result (b)																						
(a) Generations=100; Population=200; Projects=40; Amount=5000; V1=30; V2=20; U1=10 (b) Generations=200; Population=200; Projects=40; Amount=5000;V1=30; V2=20; U1=10																						
ALGORITHM	No. Project Best Port folio	Cost Best Port folio	Projects																			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Mas	26	4883	0	1	1	0	0	0	1	0	1	0	1	1	1	1	1	1	1	0	1	1
GA	27	4965	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1

Table 7. Comparison of the memetic algorithm, against genetic algorithm to form the optimal portfolio with 40 projects

Memetic Algorithm (a)Vs Genetic Algorithm Result (b)																						
(a) Generations=100; Population=200; Projects=40; Amount=5000; V1=30; V2=20; U1=10 (b) Generations=200; Population=200; Projects=40; Amount=5000;V1=30; V2=20; U1=10																						
ALGORITHM	No. Project Best Portfolio	Cost Best Portfolio	Projects																			
			21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Mas	26	4883	1	1	0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	1	0
GA	27	4965	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0

5 Conclusions

This article provides a solution to the Social Portfolio Problem by Evolutionary Algorithms, creating a Memetic algorithm that includes the model of outranking and ranked instance through which DM´s preferences are modeled during the search process.

The results presented show that Memetic algorithm has a competitive performance. The solution that follows the ranking information is generally dominated by other solutions which increase the number of projects in the portfolio. The quality of solutions indicates that the algorithm reaches the zone where the best portfolios are located it helps the *DM* to explore the solutions at hand, analyze his preferences and to clarify his decision policies.

In the first experiment the memetic algorithm reaches the optimal number of projects that can be supported. In the second experiment the two algorithms evaluated 25 projects, the memetic algorithm achieving an average error of 6.66% by comparing the amount of optimal portfolio projects found by the genetic algorithm and the second experiment evaluated 40 project, achieving an average error of 4.7% by comparing the amount of optimal portfolio projects found by the genetic algorithm.

References

[1] Punkka, A., Salo, A.: Rank-Based Sensitivity Analysis of Multiattribute Value Models. Abstract to INFORMS Annual Meeting 2008, Washington, DC, USA (2008)

[2] Carazo, A.F., Gómez, T., Molina, J., Hernández-Díaz, A.G., Guerreo, F.M., Caballero, R.: Solving a comprehensive model for multiobjective project portfolio selection. Computers & Operations Research 37(4), 630–639 (2010)

[3] Castro, M.: Development and implementation of a framework for I&D in public organizations. Master´s thesis. Universidad Autónoma de Nuevo León (2007)

[4] Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Genetic and Evolutionary Computation. Springer (2007)

[5] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester-New York-Weinheim-Brisbane-Singapore-Toronto (2001)

[6] Durillo, J.J., Nebro, A.J., Coello Coello, C.A., García-Nieto, J., Luna, F., Alba, E.: A study of multi-objective metaheuristics when solving parameter scalable problems. IEEE Transactions on Evolutionary Computation 14(4), 618–635 (2010)

[7] Fernández, E., Navarro, J.: A genetic search for exploiting a fuzzy preference model of portfolio problems with public projects. Annals OR 117, 191–213 (2002)

[8] Fernández, E., López, E., Bernal, S., Coello Coello, C.A., Navarro, J.: Evolutionary multiobjective optimization using an outranking-based dominance generalization. Computers & Operations Research 37(2), 390–395 (2010a)

[9] Fernández, E., López, E., López, F., Coello Coello, C.A.: Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. Information Sciences 181(1), 44–56 (2010b)

[10] Fernández, E., Félix, L.F., Mazcorro, G.: Multi-objective optimization of an outranking model for public resources allocation on competing projects. Int. J. Operational Research 5(2), 190–210 (2009)

[11] García, R.: Hyper-Heuristic for solving social portfolio problem. Master´s Thesis, Instituto Tecnológico de Cd. Madero (2010)

[12] Ghasemzadeh, F., Archer, N., Iyogun, P.: A zero-one model for project portfolio selection and scheduling. Journal of the Operational Research Society 50(7), 745–755 (1999)

[13] Greco, S., Mousseau, V., Słowinski, R.: Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. European Journal of Operational Research (2007) doi:10.1016/ j.ejor

[14] Marakas, G.M.: Decision Support Systems in the 21th Century. Prentice Hall, New Jersey (1999)

[15] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springuer Series Artificial Intelligence, p. 390. Springer, New York (1996)

[16] Olmedo Pérez, R.A.: Avances en la modelación basada en Relaciones borrosas de Sobre-clasificación para sistemas de Apoyo a la decisión en grupo. Tesis para obtener el grado de Doctor en Ciencias de la Computación (2009)

[17] Osyczka, A.: Multicriteria optimization for engineering design. En Design Optimization, pp. 155–183. Academic Press (1985)

[18] Peñuela, C., Granada, M.: Optimización multiobjetivo usando un algoritmo genético y un operador elitista basado en un ordenamiento no dominado (NSGA-II). Scientia Et Technica 8(35), 175–180 (2007)

[19] Reiter, P.: Metaheuristic Algorithms for Solving Multi-objective/Stochastic Scheduling and Routing Problems. Tesis Doctoral, University of Wien (2010)

[20] Rivera Zárate, G.: Tesis de Doctorado "Solución a Gran Escala del Problema de Cartera de Proyectos Sociales". Instituto Tecnológico de Ciudad Madero (2011)

[21] Roy, B.: The Outranking Approach and the Foundations of ELECTRE methods. In: Reading in Multiple Criteria Decision Aid, pp. 155–183. Spinger (1990)

[22] Roy, B., Slowinski, R.: Handling effects of reinforced preference and counter-veto in credibility of outranking. European Journal of Operational Research 188(1), 185–190 (2008)

[23] Wang, Y., Yang, Y.: Particle swarm optimization with preference order ranking for multi-objective optimization. Information Sciences 179(12), 1944–1959 (2009)

Multicriteria optimization of interdependent project portfolios with ‘a priori’ incorporation of decision maker preferences

Laura Cruz[1], Eduardo R. Fernandez[2], Claudia G. Gomez[1] and Gilberto Rivera[1]

[1]Madero Institute of Technology, Cd. Madero, Mexico
[2]Autonomous University of Sinaloa, Culiacán, Mexico
lauracruzreyes@itcm.edu.mx, eddyf@uas.edu.mx, cggs71@hotmail.com, riveragil@gmail.com

Abstract

One of the most important management issues lies in determining the best portfolio of a given set of investment proposals. This decision involves the pursuit of multiple criteria, and has been commonly addressed by implementing a two-phase procedure whose first step identifies the efficient solution space. In this paper we introduce our algorithm called Non-Outranked Ant Colony Optimization (NO-ACO) that optimizes portfolios with inter-projects interactions whilst takes into account the DM’s preferences by incorporating *a priori* preferences articulation. Experimental tests show the advantages of our proposal over the two-phase approach. Also, NO-ACO performed particularly well for problems with high dimensionality.

Keywords: portfolio selection, interdependent projects, multicriteria optimization, preferences incorporation.

1. Introduction

Portfolio problems are ubiquitous in business and government organizations. Usually, there are more good ideas for projects or programmes than resources (funds, capacity, time, etc.) to support them ([1]). Manufacturing enterprises recognize that success depends on the selection of research and development (R&D) project portfolios, expecting that these projects permit them to develop new products that generate growing benefits. Local governments allocate public funds to projects and programmes that improve social and educational service. Environmental regulations and alternative policy measures attempt to mitigate harmful consequences of human activity ([2]). To fight poverty, governments in underdeveloped countries fund many helpful social programmes.

Portfolio consequences are usually described by multiple attributes related to the organizational strategy. A vector $z(x)$=<$z_1(x)$, $z_2(x)$, …, $z_p(x)$> is associated to the consequences of a portfolio x considering p criteria. This is a vector representation of the portfolio’s impact. In the simplest case, $z(x)$ is obtained from a cumulative sum of benefits of the selected projects, but under interacting project conditions, it is necessary to consider the contribution of interdependent project groups. Without loss of generality, we can assume that higher criterion values are preferred to lower values. The best portfolio is obtained by solving:

$$\max_{x \in R_F} \{< z_1(x), z_2(x), \ldots, z_p(x) >\} \qquad (1)$$

where R_F is the feasible portfolios space, usually determined by the available budget, and by constraints for the kinds of projects, social roles and geographic zones. Problem 1 is badly defined mathematically, yet people must solve it. To solve Problem 1 means to find the best compromise solution according to the system of preferences and values of the decision maker (DM).

In the scientific literature, the problem expressed by (1) has received great interest in R&D management by manufacturing and industrial enterprises (e.g. [3, 4, 5, 6, 7, 8]). Most of these approaches can also be applied in public sector. Perhaps, what best characterizes the portfolio problems in non-profit organizations are the emphasis on intangible criteria and, likely, a higher number of project proposals and objectives to optimize. For example, in socially responsible organizations, the number of criteria used for capital investment may be about a dozen ([9]). Even more objective functions should be considered in basic research project management (cf. [10]). A high number of project proposals can apply for public support in a simple call for projects. For instance, in 2012 the U. S. state of Georgia had a list of over 1600 applicant projects only at the State Department of Transportation ([11, 12, 13, 14]). There should be a large set of Pareto-efficient solutions to Problem 1. However, the decision maker has to select only one portfolio according to her/his preferences on the portfolio’s consequences expressed by $z(x)$.

2. An outline of the state of the art

Only non-dominated solutions to (1) can fulfill the necessary conditions for being considered the best portfolio. So, most solution methods seek to generate the Pareto frontier, and later, by some interactive method, multicriteria procedure or heuristic, try to identify the best compromise. These approaches assume that the DM has the capacity to make valid judgments about the set of efficient points until reaching the best compromise. This way to identify the best solution is commonly referred as the *a posteriori* preferences modeling (cf. [15]).

Ghasemzadeh et al. ([16]) model preferences using a weighted-sum function. They approximate the Pareto frontier by changing the weights and solving the resultant model by 0-1 programming. Stummer and Heidenberger in [5] include synergy and redundancy in selecting R&D projects; their procedure consists of three phases: 1) filtering the proposals and retaining those most promissory projects in order to reduce the set of projects to a "manageable" size, 2) generating the efficient frontier of portfolios for the reduced set by an integer linear programming method, and 3) supporting the decision making process, helping the DM to identify the best compromise by an interactive process.

However, most recent works show the advantages of multiobjective metaheuristics methods to approximate the Pareto set (e.g. [8, 17, 18, 19, 20, 21, 22, 23]). In [24] Doerner et al. combine Ant Colony Optimization (ACO) with 0-1 dynamic mathematical programming to initialize the algorithm with enhanced solutions. One of the most complete proposals was suggested by Carazo et al. in [18, 25], which model interactions among projects (such as Stummer and Heidemberger in [5]) and temporal dependencies, enabling the allocation of resources not used in previous periods. By means of a Scatter Search, Carazo et al. ([18]) outperform SPEA2 [26] in the range of 25-60 projects considering up to six objective functions.

Compared to multi-objective optimization methods based on mathematical programming, metaheuristic approaches exhibit relevant advantages: 1) they have the ability to deal with a set of solutions (called population) at the same time, allowing to approximate the efficient frontier in a single algorithm run, and 2) they are less sensitive to the mathematical properties of objective functions and problem constraints ([27]).

Despite their advantages, most metaheuristic algorithms are degraded when trying to solve problems with more than a small number of objectives ([28, 29]). Also, when they try to approximate the efficient frontier, generate a very large amount of solutions. This exceeds the cognitive abilities of an average DM to identify satisfactorily the best compromise. Even if we could apply the multicriteria decision analysis methods, this process can turn too hard, because these methods do not perform well on decision problems with so many alternatives.

In order to address these drawbacks, in [30] Fernandez et al. proposed a method of preference incorporation in multiobjective evolutionary optimization, which was after extended in [10] to project portfolio optimization. They use a fuzzy outranking preference system to identify a small privileged subset of Pareto-efficient solutions. The model is independent of the number of criteria considered by the DM, and achieves to solve instances in a range of 100-500 projects and 9-16 objectives. Another advantage is its high tolerance to imprecise objective values, and its capacity of handling ordinal and qualitative criteria. However, the model of Fernandez et al. ([10, 30]) does not consider interactions among projects, what is an important concern in most practical applications.

In light of this feedback, we propose a portfolio optimization metaheuristic approach based on the preferential model of Fernandez et al. ([10]). So, our metaheuristic inherits all advantages of their model, but we have incorporated the capacity to solve portfolios with interdependent projects. Several papers in the literature consider the synergy as an inherent characteristic of the portfolio problem (e.g. [5, 18, 24, 31]). Our solution approach, called *Non-Outranked Ant Colony Optimization* (NO-ACO) shows promising results compared to other related algorithms. Experimental results provide evidence that is very capable to get close to the Pareto frontier when is looking for the best compromise.

3. Preference incorporation in multicriteria optimization metaheuristic approaches

Because it would be difficult to determine the Pareto frontier in real applications, most search algorithms are limited to a predetermined number of efficient solutions. With the intention of finding a representative sample of the Pareto frontier, some algorithms include distance measures that favor the spread among solutions (e.g. [32, 33]). However, this do not ensure that the best compromise can be found, and if even so, the solution set exceeds the capacity of an average DM to make the decision process successfully.

In order to make easier the decision making phase, the DM would agree with incorporating his/her multicriteria preferences into the search process. This preference information is used to guide the search towards the *Region Of Interest* (ROI) ([34]), the privileged zone of the Pareto frontier that best matches the DM's preferences.

DM preference information can be expressed in different ways. According to Bechikh ([35]), the most commonly used ways are the following:

- those in which importance factors (weights) are assigned by the DM to each objective function (e.g. [36, 37]),
- those in which the DM makes pair-wise comparisons on a subset of the current population in order to rank the sample's solutions (e.g. [38, 39]),
- when pair-wise comparisons between pairs of objective functions are performed in order to rank the set of objective functions (e.g. [40, 41]),
- those based on goals or aspiration levels to be achieved by each objective (reference point) (e.g. [42, 43]),
- when the DM identifies acceptable trade-offs between objective functions (e.g. [44]);
- when the DM supplies the model's parameters to build a fuzzy outranking relation (e.g. [10, 30]);

- construction of a desirability function which is based on the assignment of some desirability thresholds (e.g. [45]).

In the field of portfolio optimization, the model of Fernandez et al. ([10]) has shown substantial benefits for tackling these problems. This model is briefly explained below.

3.1. The best portfolio in the sense of Fernandez et al. ([10])

The proposal by Fernandez et al. ([10, 30]) is based on the relational system of preferences described by Roy in [46]. A crucial model is the degree of credibility of the statement "*x* is at least as good as *y*"; this is represented as $\sigma(x,y)$ and could be calculated using proven methods of literature, such as ELECTRE ([47]) and PROMETHEE ([48]). The proposal by Fernandez et al. ([10]) identifies one of the following relations for each pair of portfolios (x,y) controlled by the parameters λ, β, and ε ($0 \le \varepsilon \le \beta \le \lambda$ and $\lambda \ge 0.5$):

1) *Indifference*: From the DM perspective, both alternatives have a high degree of equivalence; therefore he/she cannot state that one is preferred over other. This relationship is denoted as $x\mathrm{I}y$. In terms of $\sigma(x,y)$ is defined as the conjunction of:

a. $\sigma(x,y) \ge \lambda \wedge \sigma(y,x) \ge \lambda$.

b. $|\sigma(x,y) - \sigma(y,x)| \le \varepsilon$.

2) *Strict preference*: Denoted as $x\mathrm{P}y$, represents the situation when the DM significantly prefers *x*. It is defined as a disjunction of the conditions:

a. *x* dominates *y*.

b. $\sigma(x,y) \ge \lambda \wedge \sigma(y,x) < 0.5$.

c. $\sigma(x,y) \ge \lambda \wedge (0.5 \le \sigma(y,x) \le \lambda) \wedge (\sigma(x,y) - \sigma(y,x)) \ge \beta$.

3) *Weak preference*: Represented as $x\mathrm{Q}y$, models a state of doubt between $x\mathrm{P}y$ and $x\mathrm{I}y$. It can be defined as the conjunction of:

a. $\sigma(x,y) \ge \lambda \wedge \sigma(x,y) \ge \sigma(y,x)$.

b. $\neg x\mathrm{P}y \wedge \neg x\mathrm{I}y$.

4) *Incomparability*: From the point of view of DM, there is a high heterogeneity between the alternatives, so he/she cannot set a preference relation between them. It is denoted as $x\mathrm{R}y$, and is expressed in terms of $\sigma(x,y)$ as $x\mathrm{R}y \Rightarrow \sigma(x,y) < 0.5 \wedge \sigma(y,x) < 0.5$.

5) *k-Preference*: Represents a doubt between $x\mathrm{P}y$ and $x\mathrm{R}y$, and is denoted as $x\mathrm{K}y$. $(x,y) \in \mathrm{K}$ if the following three conditions are true:

a. $0.5 \le \sigma(x,y) < \lambda$.

b. $\sigma(y,x) < 0.5$.

c. $\sigma(x,y)$-$\sigma(y,x) > \beta/2$

Indifference corresponds to the existence of clear and positive reasons that justify equivalence between the two options. Besides, incomparability represents situations where the DM cannot, or does not want to, express a preference. Strict preference is associated with conditions in which the DM has clear and well-defined reasons justifying the choice of an alternative over the other. However, due to the DM usually has a non-ideal behavior, there exist the weak preference and the *k*-preference. These relations can be considered as "weakened" ways of the strict preference.

The model parameters need to be adjusted according to the specific characteristics of the problem and the DM. This can be done by an interaction between the DM and a decision analyst, utilizing, if necessary, indirect elicitation methods to support this task ([49, 50, 51]).

From a set of feasible portfolios *O*, the preferential system defines the following sets:

1) $S(O,x) = \{y \in O \mid y\mathrm{P}x\}$, composed of the solutions that strictly outrank *x*.

2) $NS(O) = \{x \in O \mid S(O,x) = \varnothing\}$, is known as *non-strictly-outranked frontier*.

3) $W(O,x) = \{y \in NS(O) \mid y\mathrm{Q}x \vee y\mathrm{K}x\}$, composed of the non-strictly-outranked solutions that weakly outrank *x*.

4) $NW(O) = \{x \in NS(O) \mid W(O,x) = \varnothing\}$, is known as *non-weakly-outranked frontier*.

Obviously, solutions that presumed to be the best compromise among a set *O* of actions must belong to $NS(O)$. However, there may be more than one solution with such feature, so more information is needed to describe the DM's preferences and enhance the optimization process. A solution belonging to $NW(O)$ has a greater potential to be the best compromise that those that do not have this condition.

Besides the weak outranking, the net flow score is another measure used by Fernandez et al. ([10, 30]) to identify the DM's preferences in the non-strictly-outranked frontier. It can be defined as:

$$F_n(x) = \sum_{y \in NS(O)\setminus\{x\}} [\sigma(x,y) - \sigma(y,x)] \qquad (2)$$

Since $F_n(x) > F_n(y)$ indicates a preference for *x* over *y*, Fernandez et al. ([10]) define:

1) $F(O,x) = \{y \in NS(O) \mid F_n(y) > F_n(x)\}$, as the set of non-strictly-outranked solutions that surpass in net flow to *x*.

2) $NF(O) = \{x \in NS(O) \mid F(O,x) = \varnothing\}$, is known as *net flow non-outranked frontier*.

Then, the model proposed in [10] suggests finding the best compromise in *O* by solving Problem 3:

$$\min_{x \in O}\{|S(O,x)|, |W(O,x)|, |F(O,x)|\} \qquad (3)$$

with pre-emptive priority favoring $|S(O,x)|$. Fernandez et al. proved in [10] that the best portfolio compatible with the fuzzy outranking relation σ should be a (0,0,0) solution to Problem 3 with $O=R_F$.

4. Our proposal

Fernandez et al. ([10]) solve Problem 3 by using an evolutionary algorithm inspired by NSGA2. This performs well when interdependent projects are not considered. However, if project interaction is addressed, the crossover operation could remove convenient synergetic projects from the portfolio. Therefore, we prefer to use a building-oriented metaheuristic approach.

Our algorithm, NO-ACO (*Non-Outranked Ant Colony Optimization*), is based on the optimization idea proposed in [52] by Dorigo and Gambardella which has been adapted more than once to find a set of Pareto solutions (e.g. [31, 53, 54]). Unlike other multiobjective ant-based optimization methods, NO-ACO incorporates the preference model from [10, 30]. The algorithm performs the optimization process through a set of agents called ants. Each ant in the colony builds a portfolio by selecting a project at a time. The way how to choose each project is called *selection rule*. When all ants have finished constructing their portfolios, these are evaluated and each ant drops pheromone according to this assessment. Pheromone is a learning kind that allows next generation of ants to acquire knowledge of the structure of the best solutions. To prevent premature convergence, the colony includes a strategic oblivion mechanism, known as evaporation, which reduces the pheromone trail every specified period of time.

In order to improve the intensification, NO-ACO includes a variable neighborhood search for the best solutions. This local search runs once per iteration.

This intensifier scheme is complemented by a diversifier mechanism, in which portfolios that have remained non-strictly-outranked for more than γ generations are removed from the solution set. This allows relaxing the selective pressure. This behavior is desirable whether the algorithm has only found out local optima.

The optimization process ends when reaching a predetermined termination criterion, such as a maximum number of iterations, or subsequent recurrence of the best solution. The following sections describe in further detail the elements of NO-ACO algorithm.

4.1. Pheromone representation

Pheromone is usually represented by the Greek letter τ and is modeled in NO-ACO as a two dimensional array of size $N \times N$, where N is the total number of applicant project proposals. The pheromone between two projects i and j is represented as $\tau_{i,j}$, and indicates how good is that both projects receive financial support. Pheromone values are in range (0,1], initializing at the upper limit to prevent premature convergence. The pheromone matrix acts as a reinforcement learning structure reflecting the knowledge gained by ants that formed high-quality portfolios. Pheromone transmits it to ants of the next generation for building better solutions.

4.2. Selection rule

Each ant builds its portfolio by selecting one by one the projects, taking into account two factors:

1) *Local knowledge*: This considers the benefits provided by the project to the portfolio and how much resource it consumes. Local knowledge for a project i is denoted as η_i and is calculated by the expression:

$$\eta_i = \frac{\frac{1}{c_i}\sum_{j=0}^{p} f_j(i)}{\max_{k \in X}\left\{\frac{1}{c_k}\sum_{j=0}^{p} f_j(k)\right\}}$$

(4)

where c_i is the cost of project i, p is the number of objectives, X is the applicant project list, and $f_j(i)$ the benefits of the project i to the jth objective. Formula 4 promotes the inclusion of projects that have a good balance between intended objectives and requested budget.

2) *Global knowledge*: This takes into account the experience of previous generations ants, expressed in the pheromone matrix. The global knowledge for the project i to be included in a portfolio x is denoted by $\overline{\tau(x,i)}$ and is defined by the expression:

$$\overline{\tau(x,i)} = \frac{\sum_{j=0}^{N}(x_j)\tau_{i,j}}{\sum_{j=0}^{N} x_j} \qquad (5)$$

where N is the total number of applicant projects, x_j is the binary value indicating whether the jth project is included in the portfolio x, and $\tau_{i,j}$ is the pheromone for projects i and j. The numerator in (5) is the total sum of pheromone between i and each project in the portfolio x; and the denominator is the cardinality of x. The global knowledge favors the selection of projects that were part of the best portfolios in previous generations. At the first iteration this knowledge has no effect on portfolio formation process.

Both knowledge factors are linearly combined into a single evaluation function:

$$\Omega(x,i) = w \cdot \eta_i + (1-w) \cdot \overline{\tau(x,i)} \qquad (6)$$

where w is a weight parameter between global and local knowledge, and should receive a value between zero and one. Each ant in the colony has a different value for w generated at random. Function Ω forms the basis of the selection rule.

If x is a partially-constructed portfolio, one or more projects may be included to x. From among all project proposals, only those ones that are not part of x and whose inclusion favors the fulfillment of budgetary constraints should be considered. This set is known as *candidate project list* and is denoted by X^{Θ}. Note that $X^{\square\Theta}$ is a subset of X. The choice of what $j \in X^{\Theta}$ will be added is made by using the selection rule:

$$j = \begin{cases} \arg\max_{i \in X^{\theta}} \{\Omega(x,i)\} & \text{if } \wp \le \alpha_1, \\ \mathcal{L}_{i \in X^{\theta}} \{\Omega(x,i)\} & \text{if } \alpha_1 < \wp \le \alpha_2, \\ \ell_{i \in X^{\theta}} & \text{otherwise.} \end{cases} \tag{7}$$

where j is the next project to be included, $\wp$ is a pseudorandom number between zero and one; α_1 is a parameter that sets the intensification probability in the algorithm (choosing the project with the greatest value of Ω); whilst $\alpha_2 - \alpha_1$ is the probability to trigger a middle state between intensification and diversification (selecting randomly a project i with probability proportional to its assessment Ω), this selection scheme is represented by $\mathcal{L}$; in the event that $\wp > \alpha_2$, diversification is promoted by means of the function ℓ (taking a project uniformly at random).

4.3. Pheromone laying and evaporation

At the beginning of the first iteration, the pheromone matrix is initialized to $\tau_{i,j} = 1$ for all $(i, j) \in N \times N$. After that, each ant constructs a feasible portfolio. In a colony with n ants, n new solutions are generated at the end of each iteration, and also there is a set of size m with the best portfolios found out in previous iterations. If all alternatives are integrated into a set O whose cardinality is $n+m$, we can identify the non-strictly-outranked front $NS(O)$.

In addition, $NS(O)$ is subdivided into domination fronts similarly to NSGA-II ([32]). The fronts are obtained considering two objectives to minimize: $W(O,x)$ and $F(O,x)$, according to the best-compromise definition given in (3). The set composed by these fronts is denoted by $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{k+1}, \ldots\}$, where $\mathcal{F}_1$ contains the non-dominated solutions, $\mathcal{F}_2$ contains the dominated by only one solution, $\mathcal{F}_3$ the dominated by two solutions, and so forth. In general, the solutions dominated by k solutions are in $\mathcal{F}_{k+1}$. The set $\mathcal{F}$ will be used in the pheromone intensification in order to increase the selective pressure towards the best compromise.

Each pair of projects (i,j) for each solution $c \in O$ intensifies the pheromone trail according to the expression:

$$\tau_{i,j} = \begin{cases} \tau_{i,j} + \Delta\tau_{i,j} & \text{if } c \in NS(O), \\ \tau_{i,j} & \text{otherwise.} \end{cases} \tag{8}$$

If c is a non-strictly-outranked solution, then there is a k such that $c \in \mathcal{F}_k$. The pheromone increase depends on k, and is defined as:

$$\Delta\tau_{i,j} = \left(\frac{|\mathcal{F}| - k + 1}{|\mathcal{F}|} \right) (1 - \tau_{i,j}) \quad \text{if } c \in \mathcal{F}_k. \tag{9}$$

If there are cycles in the strict preference relation, no solution can be identified into $NS(O)$. This may result from a wrong settlement of model parameters; in this case, a closer interaction with the DM will be required for reaching a consistent preference representation. Another reason may be a high heterogeneity in preferences when the DM is a conflicting group.

At the end of each iteration, the entire pheromone matrix is evaporated through a multiplication by a constant factor between zero and one, denoted as ρ.

4.4. Local search

The algorithm intensification is promoted by a greedy variable-neighborhood local search that is only carried out on non-strictly-outranked solutions. This search explores regions near to the best known solutions by a simple scheme consisting of selecting randomly v projects, and generating all possible combinations of them for each solution in the non-strictly-outranked frontier. Small values for v provoke a too greedy behavior, whereas large values produce intolerable computation times. In our experiments we obtained a good balance between both by using $v = \lceil \ln N \rceil$.

5. Case study: Optimization of social assistance portfolios

Consider a DM facing a portfolio problem, with 100 project proposals that attempt to benefit the most precarious social classes. The project quality is measured as the number of beneficiaries for each of nine criteria established previously. Each objective is associated to one of three classes (extreme poverty, lower class and lower-middle class) and one of three levels of impact (low, medium and high).

Table 1: Effect of preferences incorporation on the Pareto Ant Colony Optimization algorithm

Instance	Algorithm	Time (seconds)	Size of the solution set	Non-dominated solutions in $O_1 \cup O_2$	Solutions belonging to $NS(O_1 \cup O_2)$	Obtains the best compromise in $O_1 \cup O_2$
1	P-ACO	3448.07	2006	928	10	
	P-ACO-P	536.66	15	15	10	✓
2	P-ACO	3470.29	2514	1295	7	
	P-ACO-P	775.94	19	19	13	✓
3	P-ACO	3485.16	2456	280	13	
	P-ACO-P	1112.49	34	34	17	✓
4	P-ACO	3591.27	2587	1392	10	✓
	P-ACO-P	734.58	38	37	19	✓
5	P-ACO	3525.85	2245	1165	10	
	P-ACO-P	1035.85	21	21	15	✓

Note: O_1 and O_2 are the solution sets generated by P-ACO and P-ACO-P respectively
The best compromise is a (0,0,0) solution to Problem 3

The total budget to distribute is 250 million dollars. The proposals can be grouped into three types according to their nature, and into two geographic regions according to the impact location. Furthermore, desiring to provide equitable conditions, the DM imposes the following restrictions: 1) the budget allocated to support each project type should vary between 20% and 60% of the total budget, and 2) the financial support allocated to each region must be at least 30% of the total, and no more than 70%.

Also, the DM has identified 20 relevant interactions among projects: four of them are cannibalization phenomena, six correspond to situations of mutually excluding projects, and ten are synergism interactions. There are up to five projects per interaction. Into our algorithm, these relations are modeled as in [5].

Below, we present a range of experiments to verify the validity and advantages of our approach to solve this case study. They give evidence of the benefits of incorporating DM preferences during the optimization process, and thus, they also prove that our approach has good potential in solving real resource-allocation problems.

5.1. Effect of the DM´s preference incorporation

In order to appraise which is the effect on a multi-objective optimization algorithm by incorporating DM preferences, we implemented the P-ACO algorithm proposed by Doerner et al. in [31].

To the best of our knowledge, P-ACO is the most prominent ant colony algorithm applied to solve project portfolio selection.

We also developed a version of P-ACO including the preferential model described in Section 3. This adaptation was called P-ACO with preferences (P-ACO-P). The latter, instead of approximating the Pareto frontier defined by the nine maximizing objectives of the problem, searches the best compromise expressed by (3). It is easy to prove that the set of solutions pursued by P-ACO-P is a subset of P-ACO's.

In order to reflect a credible decision situation, we assign the values suggested by Fernandez et al. in [30] to the preferential model parameters.

Both algorithms were programmed in Java language, using the JDK 1.6 compiler, and NetBeans 6.9.1 as integrated development environment. The experiments were run on a Mac Pro with processor Intel Quad-Core 2.8 GHz and 3 GB of RAM. The P-ACO parameter setting was the suggested by Doerner et al. ([31]). The version that incorporates preferences has the same setting values.

Table 1 shows the experimental results on five artificial instances following the case-study features.

As is observed in Table 1, incorporating preferences provides a closer approximation to a privileged region of the Pareto frontier. The version considering preferences provides solutions that dominated the 57%, on average, of solutions from the algorithm original version. There is also a significant run-time reduction (in the test cases, it was 76% on average). Also, if the model of preferences matches with the DM´s preferences, the real best compromise among the set of all portfolios generated is always identified by P-ACO-P. Furthermore, when the DM has to choose one alternative as the final decision, the thousands of portfolios from P-ACO shall make difficult to make a decision. By incorporating preferences, this drawback is very strongly reduced.

5.2. Analysis of the algorithm performance

This section is presented with the intention to provide evidence of the performance of our algorithm NO-ACO. The main differences from the P-ACO-P (with preferences) are presented in Table 2. In order to verify whether the NO-ACO strategies have been properly instantiated, in this section we compare the performance of NO-ACO with P-ACO incorporating preferences.

We have deactivating the local search of our algorithm, with the intention of achieving comparison conditions as balanced as possible.

The experimental results are shown in Table 3, where can be observed that, although P-ACO-P finds larger solution sets, most of these are suboptimal solutions with respect to NO-ACO's. So the non-strictly-outranked frontier is better approximated by NO-ACO.

Table 2: Main differences between P-ACO (with preferences) and NO-ACO algorithms

Algorithm element	The way like P-ACO-P carries out it	The way like NO-ACO carries out it
Pheromone representation	A two-dimensional matrix with size *N* ´ *p* . Where *N* is the number of applicant projects and *p* is the total of criteria.	A two-dimensional matrix with size *N* ´ *N* .
Pheromone laying	The best and the second best solutions for each objective intensify the pheromone.	The solutions intensify the pheromone according to dominance fronts.
Pheromone evaporation	The ants do it during the solution building.	The entire pheromone matrix is evaporated once per iteration.
Lifespan for the ants	It is randomly generated. Every time an ant adds a project, the lifespan is decreased by one.	It is equal to budget. Every time an ant adds a project, the project cost is deducted from the lifespan.
Local knowledge	It promotes the forming of feasible portfolios.	It promotes the inclusion of projects with higher ratio between benefits and cost.
Ignoring old solutions of the search process.	It is not considered.	Non-strictly-outranked solutions with more than λ iterations are taken out from the search.

In addition, for all test instances, our proposal is able to identify the best compromise from both solution sets. Concerning run times, there are no significant differences according to a Student's *t* test for paired samples, using a confidence level of 90%.

The NO-ACO parameter setting used to obtain the results in Table 3 is: $\alpha_1 = 0.65$, $\alpha_2 = 0.75$, $\rho = 0.10$, $\gamma = 5$, $rep_{\max} = 21$, and $iter_{\max} = 1000$. Moreover, the colony has one hundred ants. This setting was obtained from exploring parameters values with the objective of achieving a good algorithmic performance.

5.3. Solving problems with high dimensionality

The tests shown in this section are limited to one hundred projects and nine objectives. These dimensions exceed those addressed by most studies in the scientific literature (e.g. [24, 25, 31, 61]). These dimensions are appropriate for most portfolio problems in the business sector; however, in public organizations, the problem size may be larger. In order to explore the capacity of our algorithm to solve instances with a large size, we generated a set of instances with 500 projects and 16 criteria to optimize.

The interpretation is similar to that described at the beginning of this section: there is a budget to distribute to 250 million dollars, also the DM want to keep balancing conditions and has grouped the projects into two areas and in three regions and imposed budgetary constraints for each one (30-70% for each area and 20-60% for each region). In addition, the DM has identified 100 relevant interactions between projects: 20 are cannibalization phenomena, 30 correspond to redundancy among projects and 50 are synergies that generate added value.

Unlike the 100-projects case, in these instances it is not possible to generate an acceptable approximation of the Pareto frontier that can be used as reference for comparison purposes. Even the best multiobjective algorithms are degraded attempting to generate it. This combined with computation times that would be intolerable or an abrupt interruption of the algorithms if they fail to converge towards the frontier.

Among several heuristics frequently used, we chose one based on assigning budgetary resources according to project-ranking information. Here, a project ranking is built by using a cost-benefit ratio; the benefit is modeled by a weighted sum, whose weights are adjusted to reflect the DM's preferences. The project ranking is built following the order given by the cost-benefit ratio.

Once the set of projects has been ranked, the resources may be allocated by following the priorities implicit in the rank order until no resources are left. This ensures, at least, the inclusion of projects that provide more benefit per dollar.

Synergism can be tackled if the inter-projects interactions are modeled as dummy projects that can be ranked. Table 4 concentrates only five of 164 solutions found out by NO-ACO as an approximation to non-strictly-outranked frontier. Our algorithm converges after 21,625 seconds. The best compromise found (Solution 1) outperforms the ranking-based portfolio, even in Pareto sense.

Table 3: Comparative analysis of the NO-ACO performance

Instance	Algorithm	Time (seconds)	Size of the solution set	Non-dominated solutions in $O_1 \cup O_2$	Solutions belonging to $NS(O_1 \cup O_2)$	Obtains the best compromise in $O_1 \cup O_2$
1	P-ACO-P	536.66	15	0	0	
	NO-ACO	248.68	10	10	10	✓
2	P-ACO-P	775.94	19	0	0	
	NO-ACO	891.76	6	6	6	✓
3	P-ACO-P	1112.49	34	0	0	
	NO-ACO	789.09	5	5	5	✓
4	P-ACO-P	734.58	38	0	0	
	NO-ACO	763.98	7	7	7	✓
5	P-ACO-P	1035.85	21	16	9	
	NO-ACO	456.43	10	10	9	✓

Note: O_1 and O_2 are the solution sets generated by P-ACO and P-ACO-P respectively
The best compromise is a (0,0,0) solution to Problem 3

Table 4: A sample of the non-strictly-outranked frontier generated by NO-ACO compared to the ranking-based solution.

	Portfolio	Values of objective functions																Number of solutions that outranks it		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	strictly	weakly	in net flow score
by NO-ACO	1	106	806	504	612	107	811	502	605	983	871	473	610	108	847	499	597	0	0	0
	2	96	766	467	556	98	786	459	562	988	772	457	565	98	756	454	545	0	0	1
	3	98	730	461	562	99	740	475	564	988	796	464	563	95	767	453	541	0	1	2
	4	100	742	479	545	94	744	459	565	992	785	451	547	96	745	447	535	0	2	1
	5	96	742	462	553	95	751	456	562	999	809	454	562	94	776	452	546	0	2	1
Ranking-based		96	736	471	558	95	762	453	561	944	768	469	565	97	756	436	540	5	0	5

Another ten instances were generated following the same features. When they were solved by NO-ACO, we observe the same behavior: the ranking-based portfolio was dominated by the best compromise by NO-ACO. This test gives some evidence of the applicability of our approach to solve large-scale real instances.

6. Conclusions and future work

We have presented an original proposal to optimize interdependent projects portfolios. This proposal is an adaptation of the well-known Ant Colony Optimization metaheuristic, but incorporating preferences based on the outranking model by Fernandez et al. ([10]).

Our algorithm (NO-ACO) searches for optimal portfolios in synergetic conditions and can handle interactions impacting both objectives and costs. Redundancy is also considered during portfolio formation.

By incorporating preferences, the selective pressure toward a privileged zone of the Pareto frontier is increased. Thus, a zone that matches better the DM's preferences can be identified.

In comparison with other metaheuristic approaches that do not incorporate preferences, NO-ACO achieves a better closeness to the true Pareto front with less computational effort.

Being enriched by preferences, our proposal acquires the ability to solve efficiently portfolio problems with higher dimensions than those reported in scientific literature.

Compared to the popular ranking-based method, NO-ACO finds out solutions that outperform to the ranking-based portfolio, both in Pareto dominance and strict outranking.

As future work we are going to add the alternative of partial project support. It will also be important to explore the limits of this approach, by finding the top size within instances can be solved with acceptable performance.

7. References

[1] Kleinmuntz, D.N. (2011). “Improved methods for resource allocation”, in Salo, A., Keisler, J. and Morton, A. (eds.) *Portfolio Decision Analysis, Improved methods for resource allocation* , Springer, New York-Dordrecht-Heidelberg-London, v-vii.

[2] Salo, A., Keisler, J, Morton, A. (2011). “An invitation to Portfolio Decision Analysis”, in Salo, A., Keisler, J., Morton, A. (eds.), *Portfolio Decision Analysis, Improved methods for resource allocation*, Springer, New York-Dordrecht-Heidelberg-London, 3-27.

[3] Coffin, M.A, Taylor, B.W. (1996). “Multiple criteria R&D project selection and scheduling using fuzzy sets”, *Computers & Operations Research*, 23(3): 207-220.

[4] Klapka, J., Pinos, P., and Sevcik, V. (2013). “Multicriterial Projects Selection”, *Handbook of Optimization, Intelligent Systems* Reference Library, vol. 38, Springer, 245-261.

[5] Stummer, C. and Heidenberger, K. (2003). “Interactive R&D Portfolio Analysis with Project Interdependencies and time Profiles of Multiple Objectives”. *IEEE Transactions on Engineering Management*, 50:175–183.

[6] Ringuest, J.L., Graves, S.B., and Case, R.H. (2004). “Mean-Gini analysis in R&D portfolio selection”, *European Journal of Operational Research*, 154(1): 157-169.

[7] Carlsson, Ch., Fuller, R., Heikkila, M., Majlender, P. (2007). “A fuzzy approach to R&D portfolio selection”, *International Journal of Approximate Reasoning*, 44 (2): 93-105.

[8] Zhao, X., Yang, Y., Wu, G., Yang, J., and Xue, X. (2012). “A dynamic and fuzzy modeling approach for multi-objective R&D project portfolio selection”, *Journal of Convergence Information Technology*, 7(1): 36-44.

[9] Hallerbach, W., Ning, H., Soppe, A., and Spronk, J., “A framework for managing a portfolio of socially responsible investments”, *European Journal of Operational Research*, 153(2):517-529

[10] Fernandez, E., Lopez, E., Mazcorro, G., Olmedo, R., and Coello, C. (2013). “Application of the Non-Outranked Sorting Genetic Algorithm to public project portfolio selection”, *Information Sciences*, 228: 131-149.

[11] Georgia Department of Transportation (2010). Project list and final investment report. Available in http:// www.dot.ga.gov/ localgovernment/ FundingPrograms/ transreferendum/ Pages/ProjectList.aspx (October 4th, 2012).

[12] Georgia Department of Transportation (2012a). Central Savannah River Area, unconstrained project list by county. Available in http:// www.it3.ga.gov/ Doc-

uments/ UnconstrainedList/ CentralSavannah-Unconstrainedlist.pdf (October 4th, 2012).

[13] Georgia Department of Transportation (2012b). Heart of Georgia, Altamaha unconstrained project list by county. Available in http:// www.it3.ga.gov/ Documents/ UnconstrainedList/ HeartofGeorgia-UnconstrainedList-FullSet.pdf (October 4th, 2012).

[14] Georgia Department of Transportation (2012c). River Valley Area, unconstrained project list by county. Available in http:// www.it3.ga.gov/ Documents/ UnconstrainedList/ RiverValley-UnconstrainedList.pdf (October 4th, 2012).

[15] Hwang, C. L., and Masud, A.S. (1979). “Multiple Objective Decision Making. Methods and Applications”, *Lecture Notes in Economic and Mathematical Systems*, vol. 164, Springer Verlag, Berlin.

[16] Ghasemzadeh, F., Archer, N., Iyogun, P., (1999), “A zero-one model for project portfolio selection and scheduling*”, Journal of the Operational Research Society*, 50(7): 745-755.

[17] Amiri, B. (2012). “A multi-objective hybrid optimization algorithm for project selection problem”, *Journal of Basic and Applied Scientific Research*, 2(7): 6995-7002.

[18] Carazo, A. F., Contreras, I., Gómez, T., Pérez, F. (2012). “A project portfolio selection problem in a group decision-making context”. *Journal of Industrial and Management Optimization*; 8 (1); 243-261.

[19] Kremmel, T., Kubalik, J., and Biffl, S. (2011). “Software project portfolio optimization with advanced multi-objective evolutionary algorithm”, *Applied Soft Computing*, 11(1): 1416-1426.

[20] Chen, A. and Chyu, Ch. (2010). “Applying memetic algorithm in multi-objective resource allocation among competing projects”, *Journal of Software*, 5(8): 802-809.

[21] Gaytán, J., and García, J. (2009). “Multicriteria decision on interdependent infrastructure transportation projects using an evolutionary-based framework”, *Applied Soft Computing* 9(2): 512-526.

[22] Ghorbani, S., and Rabbani, M. (2009). “A new multi-objective algorithm for a project selection problem”, *Advances in Engineering Software*, 40(1): 9-14.

[23] Lin, Ch. M., and Gen, M. (2008). “Multicriteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm”, *Expert Systems with Applications*, 34(4): 2480-2490.

[24] Doerner, K., Gutjahr, W., Hartl, R., Strauss, C., and Stummer, C. (2006). “Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection”. *European Journal of Operational Research*, 171(3):830 – 841.

[25] Carazo, A. F., Gómez, T., Molina, J., Hernández-Díaz, A. G., Guerrero, F. M., and Caballero, R. (2010). “Solving a comprehensive model for multiobjective project portfolio selection”. *Computers & Operations Research*, 37(4):630–639.

[26] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical report No. 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

[27] Coello, C. A. C. (1999). “An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends”. In *Proceedings of the Congress on Evolutionary Computation*, pp. 3–13. IEEE Press.

[28] Wang, Y., and Yang, Y. (2009). “Particle swarm optimization with preference order ranking for multi-objective optimization”, *Information Sciences*, 179(12): 1944-1959.

[29] Coello, C., Van Veldhuizen, D., Lamont, G., (2007). “Evolutionary Algorithms for Solving Multi-Objective Problems”, Springer, New York.

[30] Fernandez, E. R., Lopez, E., Lopez, F., and Coello, C. A. C. (2011). “Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method”. *Information Sciences*, 181(1):44– 56.

[31] Doerner, K. F., Gutjahr, W. J., Hartl, R. F., Strauss, C., and Stummer,C. (2004). “Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection”. *Annals OR*, 131(1-4):79–99.

[32] Deb, K. (2001). “Multi-Objective Optimization Using Evolutionary Algorithms”. Wiley, first edition.

[33] “Knowles, J. and Corne, D. (2000). M-PAES: A memetic algorithm for multiobjective optimization”. In *Evolutionary Computation*, 2000, pp. 325–332. IEEE Press.

[34] Deb, K., Sundar, J., Bhaskara, U., and Chaudhuri, S. (2006). “Reference point based multiobjective optimization using evolutionary algorithms”. *International Journal of Computational Intelligence Research*, 2(3): 273–286.

[35] S. Bechikh, (2013), “Incorporating Decision Maker’s Preference Information in Evolutionary Multi-objective Optimization”, *Diss. PhD thesis, High Institute of Management of Tunis, University of Tunis, Tunisia*.

[36] Deb, K. (1999). “Multi-objective evolutionary algorithms: Introducing bias among Pareto optimal solutions”. KanGAL Report 99002, Indian Institute of Technology, Kanpur, India.

[37] Branke, J., and Deb, K. (2004). “Integrating user preferences into evolutionary multi-objective optimization”. In Y. Jin (ed.), *Knowledge Incorporation in Evolutionary Computation,* Springer, Berlin, Heidelberg, 461–478.

[38] Köksalan, M., and Karahan, I. (2010). “An interactive territory defining evolutionary algorithm: iTDEA”. *IEEE Transactions on Evolutionary Computation*, 14(5): 702–722.

[39] Battiti, R., and Passerini, A. (2010). “Brain–computer evolutionary multiobjective optimization: A genetic algorithm adapting to the decision maker”. *IEEE*

Transactions on Evolutionary Computation, 14(5): 671–687.

[40] Jin, Y. C., and Sendhoff, B. (2002). "Incorporation of fuzzy preferences into evolutionary multiobjective optimization". In *Proceedings of the 4th Asia-Pacific conference on Simulated Evolution and Learning*, Nanyang, Singapore, 26–30.

[41] Cvetkovic, D., and Parmee, I. C. (2002). "Preferences and their application in evolutionary multiobjective optimization". *IEEE Transactions on Evolutionary Computation*, 6(1): 42–57.

[42] Allmendinger, R., Li, X., and Branke, J. (2008). "Reference point-based particle swarm optimization using a steady-state approach". In *Proceedings of the 7th international conference on Simulated Evolution and Learning* (SEAL'08), Springer, Melbourne, Australia, 200–209.

[43] Molina, J., Santana-Quintero, L. V., Hernández-Díaz, A.G., Coello Coello, C. A., and Caballero, R. 2009. "g-dominance: Reference point based dominance for multiobjective metaheuristics". *European Journal of operational Research*, 197(2): 685–692.

[44] Branke, J., Kaussler, T., and Schmeck, H. (2001). "Guidance in evolutionary multi-objective optimization". *Advances in Engineering Software*, 32 (6): 499–507.

[45] Wagner, T., and Trautmann, H. (2010). "Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions". *IEEE Transactions on Evolutionary Computation*, 14(5): 688–701.

[46] Roy, B. (1996). "Multicriteria Methodology for Decision Aiding. Nonconvex Optimization and its Applications". Springer.

[47] Roy, B. (1990). "The Outranking Approach and the Foundations of ELECTRE methods", in Bana e Costa, C.A. (ed.), *Reading in multiple criteria decision aid*, Springer-Verlag, Berlin , 155-183.

[48] Brans, J. and Mareschal, B. (2005). "Promethee methods". In Greco, S. (ed.), *Multiple Criteria Decision Analysis: State of the Art Surveys*, International Series en Operations Research & Management Science, pp. 163–190. Springer-Verlag.

[49] Jacquet-Lagreze, E. and Siskos, Y. (2001). "Preference disaggregation: 20 years of MCDA experience".

[50] Doumpos, M., Marinakis, Y., Marinaki, M., and Zopounidis, C. (2009). "An evolutionary approach to construction of outranking models for multicriteria classification: The case of the electre tri method". *European Journal of Operational Research*, 199(2):496–505.

[51] Fernandez, E., Navarro, J., Mazcorro, G. (2012). "Evolutionary multi-objective optimization for inferring outranking model's parameters under scarce reference information and effects of reinforced preference", *Foundations of Computing and Decision Sciences*, 37(3): 163-197.

[52] Dorigo, M. and Gambardella, L. M. (1997). "Ant colony system: A cooperative learning approach to the traveling salesman problem". *IEEE Transactions on Evolutionary Computation*.

[53] Alaya, I., Solnon, C., y Ghedira, K. (2007). "Ant Colony Optimizationfor Multi-objective Optimization Problems". In *19th IEEE International Conference on Tools with Artificial Intelligence* (ICTAI), pp. 450–457.IEEE Computer Society.

[54] Chaharsooghi, S. and Kermani, A. H. M. (2008). "An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP)". *Applied Mathematics and Computation*, 200(1):167 – 177.

[55] Liesio, J., Mild, P., Salo, A. (2008). "Robust portfolio modeling with incomplete cost information and project interdependency", *European Journal of Operational Research*, 190(3): 679-695.

2

New Implementations of Data Mining in a Plethora of Human Activities

Alberto Ochoa[1,2], Julio Ponce[3,4], Francisco Ornelas[4],
Rubén Jaramillo[7], Ramón Zataraín[5], María Barrón[5],
Claudia Gómez[6], José Martínez[6] and Arturo Elias[3]
[1]*Juarez City University*
[2]*UNICAMP Instituto de Computacão*
[3]*Aguascalientes University*
[4]*Cuauhtémoc University*
[5]*ITC*
[6]*ITCM*
[7]*CIMAT*
[1,3,4,5,6,7]*México*
[2]*Brazil*

1. Introduction

The fast growth of the societies along with the development and use of the technology, due to this at the moment have much information which can be analyzed in the search of relevant informationto make predictions or decision making. Knowledge Discovery and Data Mining are powerful data analysis tools. The term Data mining is used to describe the non-trivial extraction of implicit, Data Mining is a discovery process in large and complex data set, refers to extracting knowledge from data bases. Data mining is a multidisciplinary field with many techniques. Whit this techniques you can create a mining model that describe the data that you will use (Ponce et al., 2009a).

Typical Data Mining techniques include clustering, association rule mining, classification, and regression.

We show an overview of some algorithms that used the data mining to solve problems that arisen from the human activities like: Electrical Power Design, Trash Collectors Routes, Frauds in Saving Houses, Vehicle Routing Problem.

One of the reasons why the Data Mining techniques are widely used is that there is a need to transform a large amount of data on information and knowledge useful.

Having a large amount of data and not have tools that can process a phenomenon has been described as rich in data but poverty in information (Han & Kamber, 2006). This steady growth of data, which is stored in large databases, has exceeded the ability of human beings to understand. Moreover, various problems they might present a constant stream of data, which may be more difficult to analyze the power of information.

1.1 Tree decisions to improve electrical power design

A decision tree (DT) is a directed acyclic graph, consisting of a node called root, which has no input arcs, and a set of nodes that have an entrance arch. Those nodes with output arcs are called internal nodes or nodes of evidence and those with no output arcs are known as leaf nodes or terminal nodes of decision (Rokach & Maimon, 2005).

The main objectives pursued by creating a DT (Safavian & Landgrebe, 1991) are:

- Correctly classify the largest number of objects in the training set (TS).
- Generalize, during construction of the tree, the TS to ensure that new objects are classified with the highest percentage of correct answers possible.
- If the dataset is dynamic, the structure of DT should be upgraded easily.

An algorithm for decision tree generation consists of two stages: the first is the induction stage of the tree and the second stage of classification. In the first stage is constructed decision tree from training set, commonly each internal node of the tree is composed of an attribute of the portion of the test and training set present in the node is divided according to the values that can take that attribute. The construction of the tree starts generating its root node, choosing a test attribute and partitioning the training set into two or more subsets, for each partition generates a new node and so on. When nodes are more objects of a class generate an internal node, when it contains objects of a class, they form a sheet which is assigned the class label. In the second stage of the algorithm, each new object is classified by the tree constructed, the tree is traversed from the root to a leaf node, from which membership is determined to some kind of object. The way forward in the tree is determined by decisions made at each internal node, according to attribute this to the test.

Pattern Recognition one of the most studied problems is the supervised classification, where it is known that a universe of objects is grouped into a given number of classes which have of each, a sample of known objects belong to it and the problem is given a new order to establish their relationships with each of those classes (Ruiz et al., 1999).

Supervised classification algorithms are designed to determine the membership of an object (described by a set of attributes) to one or more classes, based on the information contained in a previously classified set of objects (training set - TS).

Among the algorithms used for solving supervised classification are decision trees. A decision tree is a structure that consists of nodes (internal and leaves) and arches. Its internal nodes are characterized by one or more attributes of these nodes test and emerge one or more arcs. These arcs have an associated attribute value test and these values determine which path to follow in the path of the tree.

Leaf nodes contain information that determines the object belongs to a class. The main characteristics of a decision tree are: simple construction, no need to predetermine parameters for their construction, can treat multi-class problems the same way he works with two-class problems, ability to be represented by a set of rules and the easy interpretation of its structure.

1.1.1 Classifications of decision trees

There are various classifications of decision trees, for example according to the number of test attributes in their internal nodes there are two types of trees:

- Single-valued: only contain a test attribute on each node. Examples of these algorithms include ID3 (Mitchell, 1997), C4.5 (Quinlan, 1993), CART (Breiman et al., 1984), FACT (Vanichsetakul & Loh, 1988), QUEST (Shis & Loh, 1988), Model Trees (Shou et al., 2005), CTC (Perez et al., 2007), ID5R (Utgoff, 1989), ITI (Utgoff et al., 1997), UFFT (Gama & Medes, 2005), StreamTree (Jin & Agrawal, 2003), FDT (Janikowo, 1998), G-DT (Pedrycz, 2005) and Spider (Wang, et al., 2007).

- Multivalued: they have to a subset of attributes in each of its nodes. For example, PT2 (Utgoff & Brodley,1990), LMDT (Utgoff & Brodley,1995), GALE (Llora & Wilson, 2004) and C-DT.

According to the type of decision made by the tree, there are two types of trees:

- Fuzzy: give a degree of membership of each class of the data set, for example, C-DT, FDT, G-DT and Spider.
- Drives: assign the object belongs to only one class, so the object is or does not belong to a class, are examples of such algorithms: ID3, C4.5, CART, FACT, QUEST, Model Trees, CTC, LMDT, GALE, ID5R, ITI, UFFT, and StreamTree PT2.

The algorithms for generation of decision trees can be classified according to their ability to process dynamic data sets, i.e. sets in which lets you add new objects.

According to this there are two types of algorithms for generation of decision trees:

- Incremental: can handle dynamic data sets which are getting a partial solution as they are looking at the objects. Examples of such algorithms are: ID5R, ITI, UFFT, and StreamTree PT2.
- No Incremental: can only work on static data sets as needed for the solution to the dataset in its entirety. Examples include: ID3, C4.5, CART, FACT, QUEST, Model Trees, CTC, FDT, G-DT, Spider LMDT, GALE and C-DT.

1.1.2 Decision tree application

To diagnose the electric power apparatus, the decision tree method can be a highly recommended classification tool because it provides the if-then-rule in visible, and thus we may have a possibility to connect the physical phenomena to the observed signals. The most important point in constructing the diagnosing system is to make clear the relations between the faults and the corresponding signals. Such a database system can be built up in the laboratory using a model electric power apparatus, and we have made it. The next important thing is the feature extraction (Llora & Wilson, 2004).

2. Trash collectors routes organized by profiles

Waste. It is something that we produce as part of everyday living, but we do not normally think too much about our waste. Actually many cities generates a waste stream of great complexity, toxicity, and volume (see fig. 1). It includes municipal solid waste, industrial solid waste, hazardous waste, and other specialty wastes, such as medical, nuclear, mining, agricultural waste, construction and demolition (C&D) waste, household waste, etc. (OECD, 2008).

In the management of solid waste have the problem relates to the household waste is the individual decision-making over waste generation and disposal. When the people decide how much to consume and what to consume, they do not take into account how much waste they produce.

Therefore garbage collection is a very complex (even though in most cases do not perceive it) as not only identify routes used by vehicles for this purpose (which by itself is highly complex, to be taken into consideration many factors including the capability of vehicles, the amount of waste that can each container, the type of waste, which is held in each container, the distance between containers, street address, etc.), but to determine what the best way to make such collection (Marquez, 2009).

Currently a major concern in the world is the way which must be stored, recycled or destroy the waste that we produce (as they have done studies that indicate that the daily waste production per person is about an extra kilogram to the produced in the manufacture of the products we use daily) which starts with the garbage collection process.

There are many algorithms and techniques being used to improve the collection process, creating different routes on the basis of the different profiles from those who generate the garbage and of the type of waste, some of these algorithms and techniques are: Ant Colony Algorithms, Hybrid Genetic Algorithms, Data Mining, among others.

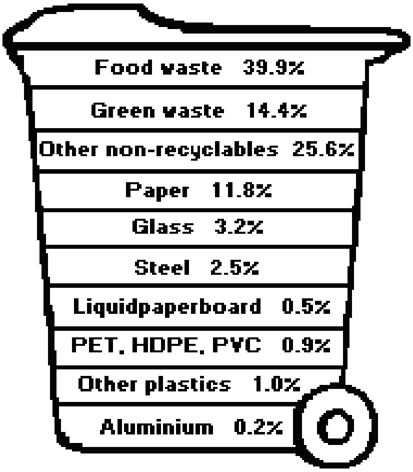


Fig. 1. Example of composition by weight of household garbage

3. Fraud analysis in saving houses

Fraud is an illegal activity, which has many variants and is almost as old as mankind. Fraud tries to take advantage in some way, usually economic, by the fraudster with respect to the shame. Specifically in the case of plastic card fraud there are several variants (Sánchez et al., 2009). The total cost of plastic card fraud is bigger respect to other forms of payment. The first line of defence against fraud is based on preventive measures such as the Chip and PIN cards. Next step is formed by methods employed to identify potential fraud trying to minimize potential losses. These methods are called fraud detection systems (FDS), and a variety of ways are used to detect the most behavior potential fraudulent.

3.1 Techniques for detection of frauds

There are two major frameworks to detect fraud through statistical methods. If fraud is conducted in a known way, the pattern recognition techniques are typically used, especially supervised classification schemes (Whitrow et al., 2009). On the other hand if the way in which fraud is not know, for example, when there are new fraudulent behaviors, outlier analysis methods are recommended (Kou et al., 2004). Previous research has established that the use of outlier analysis is one of the best techniques for the detection of fraud in general. Some studies show simple techniques for anomaly detection analysis to discover plastic card fraud. (Juszczak et al., 2008). However, to establish patterns to identify anomalies, these patterns are learned by the fraudsters and then they change the way to make de fraud. Other problem with this approach is not always abnormal behaviors are fraudulent, so a successful system must locate the true positive events, that is, transactions that are detected as fraud, but they really are fraud and not only appear to be fraudulent. Time is a factor against it, because to reduce losses, fraud detection should be done as quickly as possible. In practical applications it is possible to use supervised and unsupervised methods together.

3.1.1 Clustering

The clustering is primarily a technique of unsupervised approach, even if the semi-supervised clustering has also been studied frequently (Basu et al., 2004). Although often clustering and anomaly detection appear to be fundamentally different from one another, have developed many techniques to detect anomalies based on clustering, which can be grouped into three categories which depend on three different assumptions regarding (Chandola et al., 2009):

a. Normal data instances belong to a pooled data set, while the anomalies do not belong to any group clustered.
b. Normal instances of data are close to the cluster centroids, while anomalies are further away from these centroids.
c. The normal data belongs to large, dense clusters, whereas the anomalies belong to small and sparse clusters.

Each of the above assumptions has their own forms of detect outliers which have advantages and disadvantages between them.

3.1.2 Hybrid systems

However, as in many aspects of artificial intelligence, the hybridization is a very current trend to detect abnormalities. The reason is because many developed algorithms do not follow entirely the concepts of a simple classical metaheuristic (Lozano et al., 2010), to solve this problem is looking for the best from a combination of metaheuristics (and any other kind of optimization methods) that perform together to complement each other and produce a profitable synergy, to which is called hybridization (Raidl, 2006).

Some possible reasons for the hybridization are (Grosan et al., 2007):

1. Improve the performance of evolutionary algorithms.
2. Improve the quality of solutions obtained by evolutionary algorithms.
3. Incorporate evolutionary algorithms as part of a larger system.

In this way, Evolutionary Algorithms (EAs) have been the most frequently technique of hybridization used for clustering. However previous research in this respect has been limited to the single objective case: criteria based on cluster compactness have been the objectives most commonly employed, as the measures provide smooth incremental guidance in all parts of search space.

Since many years ago there has been a growing interest in developing and applying of EAs in multi-objective optimization (Deb, 2001).

The recent studies on evolutionary algorithms have shown that the population-based algorithms are potential candidate to solve multi-objective optimization problems and can be efficiently used to eliminate most of the difficulties of classical single objective methods such as the sensitivity to the shape of the Pareto-optimal front and the necessity of multiple runs to find multiple Pareto-optimal solutions.

In general, the goal of a multi-objective optimization algorithm is not only to guide the search towards the Pareto-optimal front but also to maintain population diversity in the set of the Pareto optimal solutions. In this way the following three main goals need to be achieved:

- Maximize the number of elements of the Pareto optimal set found.
- Minimize the distance of the Pareto front produced by the algorithm with respect to the true (global) Pareto front (assuming we know its location).
- Maximize the spreads of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible (Dehuri et al., 2009).

So it looks like a good proposal to develop a FDS with a foundation of multi-objective clustering, which places the problem of detecting fraud in an appropriate context to reality. In the same way, the system is strengthened through hybridization using PSO for the creation of clusters, and then finds the anomalies using the clustering outlier concept.

The FDS is running on the plastic card issuing institution. When a transaction arrived is sent to the FDS to be verified, the FDS receives the card details and purchase value to verify if the transaction is genuine, by calculating the anomalies, based on the expenditure profile of each cardholder, purchasing and billing locations, time of purchase, etc. When FDS confirms that the transaction is malicious, it activates an alarm and the financial institution decline the transaction. The cardholder concerned is contacted and alerted about the possibility that your card is at risk.

To find information dynamically observation for individual transactions of the cardholder, stored transactions are subject to a clustering algorithm. In general, transactions are stored in a database of the financial institution, which contain too many attributes. Although there are several factors to consider, many proposals working only with the transaction amount, with the idea of reducing the dimensionality of the problem. However, to improve the accuracy of the system is recommended to use other factors such as location and time of the transaction. So, if the purchase amount exceeds a certain value, the time between the uses of the card is low or the locations where different transactions are distant are facts to consider activating the alarm. Therefore, the alarm must be activated with a high level of accuracy.

Overall accuracy is simply the percentage of correct predictions of a classifier on a test set of "ground truth". TP means the rate of predicting "true positives" (the ratio of correctly predicted frauds over all of the true frauds), FP means the rate of predicting "false positives" (the ratio of incorrectly predicted frauds over those test examples that were not frauds, otherwise known as the "false alarm rate") (Stolfo et al., 1997).

Other two types of rates are considered for the results delivered by FDS, FN means the rate of predicting "false negatives" (the ratio of no predicted frauds over all the true frauds) and TN means the rate of predicting "true negatives" (the ratio of normal transactions detected). Table I shows the classification rate of results obtained by the FDS after analyzing a transaction.

Once clusters are established, new transaction is entered and evaluated in the FDS, to see if it belongs to a cluster set or is outside of it, seeing the transaction as an anomaly and becoming a candidate to be fraudulent. All this required the calculation of anomalies through the clustering of transaction information through a multi-objective Pareto front with the support of Particle Swarm Optimization (PSO).

Outcome	Classification
Miss	False Negative (FN)
False Alarm	False Positive (FP)
Hit	True Positive (TP)
Normal	True Negative (TN)

Table 1. Classification rate of results.

The accuracy of the FDS is represented as the fraction of total transactions (both genuine and fraudulent) that are detected as correct, which can be expressed as follows (Stolfo et al., 2000). The equation 1 shows the way to computing the precision.

$$Precision = \frac{\#\ of\ TN + \#\ of\ TP}{Total\ of\ carry\ out\ transaction} \tag{1}$$

Fig. 2 shows the idea of the full flow of the process proposed for the FDS. As shown in the figure, the FDS is divided into two parts, one that involves the creation of clusters and the second in the detection of anomalies.

Transactions outside of clusters are candidates to be considered fraudulent, however as mentioned above the accuracy of the system is a factor to be considered, which is expected to maximize in order to increase the functionality of the FDS.

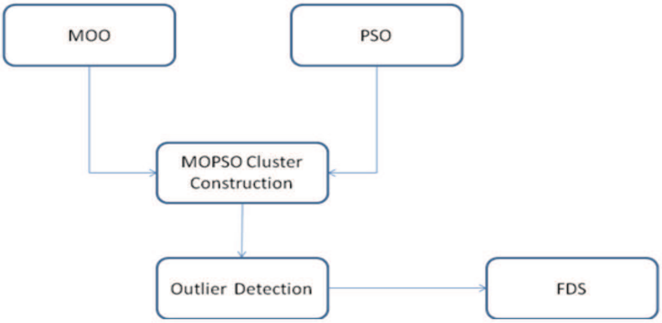


Fig. 2. Research model

4. Data mining in vehicle routing problem

With the rapid development of the World-Wide Web (WWW), the increased popularity and ease of use of its tools, the World-Wide Web is becoming the most important media for collecting, sharing and distributing information. Progress in digital data acquisition and storage technology has resulted in the growth of huge distributed databases. Due that interest has grown in the possibility of tapping these data, of extracting from them information that might be of value to the owner of the database.

The discipline concerned with this task has become known as data mining, is the analysis of observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. The relationships and summaries derived through a data mining exercise are often referred to as models or patterns. Examples include linear equations, rules, clusters, graphs, tree structures and recurrent patterns in time series.

These patterns provide knowledge on the application domain that is represented by the document collection. Such a pattern can also be seen as a query or implying a query that, when addressed to the collection, retrieves a set of documents. Thus the data mining tools also identify interesting queries which can be used to browse the collection. The system searches for interesting concept sets and relations between concept sets, using explicit bias for capturing interestingness. A set of concepts (terms, phrases or keywords) directly corresponds to a query that can be placed to the document collection for retrieving those documents that contain all the concepts of the set.

In this work, a new ant-colony algorithm, Adaptive Neighboring-Ant Search (AdaNAS), for the semantic query routing problem (SQRP) in a P2P network is presented. The proposed algorithm incorporates an adaptive control parameter tuning technique for runtime estimation of the time-to-live (TTL) of the ants. AdaNAS uses three strategies that take advantage of the local environment: learning, characterization, and exploration. Two classical learning rules are used to gain experience on past performance using three new learning functions based on the distance travelled and the resources found by the ants. These strategies are aimed to produce a greater amount of results in a lesser amount of time. The time-to-live (TTL) parameter is tuned at runtime, though a deterministic rule based on the information acquired by these three local strategies.

4.1 Semantic Query Routing Problem (SQRP)

SQRP is the problem of locating information in a network based on a query formed by keywords. The goal in SQRP is to determine shorter routes from a node that issues a query to those nodes of the network that can appropriately answer the query by providing the requested information. Each query traverses the network, moving from the initiating node to a neighboring node and then to a neighbor of a neighbor and so forth, until it locates the requested resource or gives up in its absence. Due to the complexity of the problem (Amaral, 2004) (Lui et al., 2005) (Tempich et al., 2004), (Wu et al., 2006) solutions proposed to SQRP typically limit to special cases.

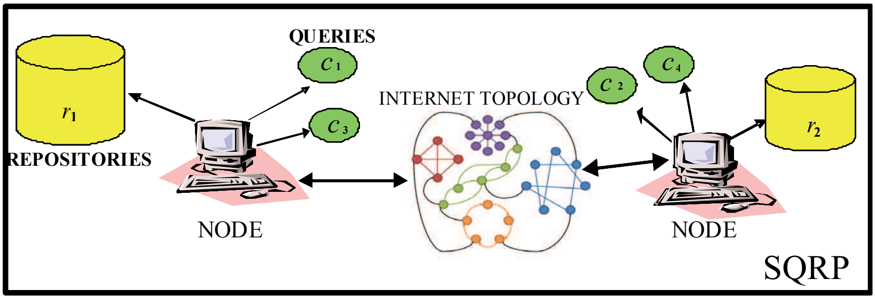


Fig. 3. SQRP Componets

The general strategies of SQRP algorithms are the following. Each node maintains a local database of documents r_i called the repository. The search mechanism is based on nodes sending messages to the neighboring nodes to query the contents of their repositories. The queries q_i are messages that contain keywords that describe for possible matches. If this examination produces results to the query, the node responds by creating another message informing the node that launched the query of the resources available in the responding node. If there are no results or there are too few results, the node that received the query forwards it to one or more of its neighbors. This process is repeated until some predefined stopping criteria is reached. An important observation is that in a P2P network the connection pattern varies among the net (heterogeneous topology), moreover the connections may change in time, and this may alter the routes available for messages to take. As showed in the Figure 1 each node has associated a database of documents ri (repository). Those are available to all nodes connected in the network. A node seeks information at the repository sending messages to its nodes neighbors.

4.2 Neighboring-Ant Search (NAS)

NAS (Cruz et al., 2008) is also an ant-colony system, but incorporates a local structural measure to guide the ants towards nodes that have better connectivity. The algorithm has three main phases: an evaluation phase that examines the local repository and incorporates the classical lookahead technique (Mihail etal., 2004), a transition phase in which the query propagates in the network until its TTL is reached, and a retrieval phase in which the pheromone tables are updated.

Most relevant aspects of former works have been incorporated into the proposed NAS algorithm. The framework of AntNet algorithm is modified to correspond to the problem conditions: in AntNet the final addresses are known, while NAS algorithm does not has a priori knowledge of where the resources are located. On the other hand, differently to AntSearch, the SemAnt algorithm and NAS are focused on the same problem conditions, and both use algorithms based on AntNet algorithm. However, the difference between the SemAnt and NAS is that SemAnt only learns from past experience, whereas NAS takes advantage of the local environment. This means that the search in NAS takes place in terms of the classic local exploration method of Lookahead (Mihail et al., 2004), the local structural metric DDC (Ortega, 2009) its measures the differences between the degree of a node and the degree of its neighbors, and three local functions of the past algorithm performance.

4.3 Adaptative Neighboring-Ant Search (AdaNAS)

AdaNAS is a metaheuristic algorithm, where a set of independent agents called ants cooperate indirectly and sporadically to achieve a common goal.

The algorithm has two objectives: it seeks to maximize the number of resources found by the ants and to minimize the number of steps taken by the ants. AdaNAS guides the queries toward nodes that have better connectivity using the local structural metric degree; in addition, it uses the well known lookahead technique, which, by means of data structures, allows to know the repository of the neighboring nodes of a specific node.

The algorithm performs in parallel all the queries using query ants. Each node has only a query ant, which generates a Forward Ant for attending only one user query, assigning the searched keyword t to the Forward Ant. Moreover the query ants realize periodically the local pheromone evaporation of the node where it is. In the Algorithm is shown the process realized by the Forward Ant. As can be observed all Forward Ants act in parallel. In an initial phase (lines 4-8), the ant checks the local repository, and if it founds matching documents then creates a backward ant. Afterwards, it realizes the search process (lines 9-25) while it has live and has not found R documents. The search process has three sections: Evaluation of results, evaluation and application of the extension of TTL and selection of next node (lines 24-28).

The first section, the evaluation of results (lines 10-15) implements the classical Lookahead technique. That is, the ant x located in a node r, checks the lookahead structure, that indicates how many matching documents are in each neighbor node of r. This function needs three parameters: the current node (r), the keyword (t) and the set of known nodes (known) by the ant. The set known indicates what nodes the lookahead function should ignore, because their matching documents have already taken into account. If some resource is found, the Forward Ant creates a backward ant and updates the quantity of found matching documents.

Algorithm: Forward ant algorithm

```
1   in parallel for each Forward Ant x(r,t,R)
2   initialization: TTL = TTLmax, hops= 0
3   initialization: path=r, $\Lambda$=r, known=r
4   Results= get_ local_ documents(r)
5   if results > 0 then
6      create backward ant y(path, results, t)
7      activate y
8   End
9   while TTL < 0 and results < R do
10     La_ results= look ahead(r,t,known)
11     if la results > 0 then
12        create backward ant y(path, la results, t)
13        activate y
14        results   results + la results
15     End
16     if TTL > 0 then
17        TTL=  TTL – 1
18     Else
19        if (results < R) and ( $\Delta TTL(x, results, hops) > 0$) then
20           TTL=  TTL + $\Delta TTL(x, results, hops)$
21          change parameters: $q= 1, Wdeg =0, \beta2=0$
22        End
23      End
24  Hops=  hops + 1
25  Known=  known$\cup$[ ( r $\cup$ $\Gamma(r)$)
25  $\Lambda = \Lambda \cup r$
27  r =  $\ell(x,r,t)$
28  add to path(r)
29  End
30  create update ant z(x, path, t)
31  activate z
32  kill x
33  end of in parallel
```

Fig. 4. AdaNAS algorithm

The second section (lines 16-23) is evaluation and application of the extension of TTL. In this section the ant verifies if TTL reaches zero, if it is true, the ant intends to extend its life, if it can do it, it changes the normal transition rule modifying some parameters (line 21) in order to create the modified transition rule. The third section of the search process phase is the selection of the next node. Here, the transition rule (normal or modified) is applied for selecting the next node and some structures are updated. The final phase occurs when the search process finishes; then, the Forward Ant creates an update ant for doing the pheromone update.

Figure 5 shows the results of the different experiments applied to NAS and AdaNAS on thirty runnings for each ninety different instances generated with the characteristics showed in (Cruz et al., 2004). It can been seen from it that on all the instances the AdaNAS algorithm outperforms NAS. On average, AdaNAS had an efficiency 81% better than NAS. The topology and the repositories were created static, whereas the queries were launched randomly during the simulation. Each simulation was run for 15,000 queries during 500 time units, each unit has 100ms. The average performance was studied by computing three performance measures of each 100 queries. Average efficiency, defined as the average of resources found per traversed edge (hits/hops).

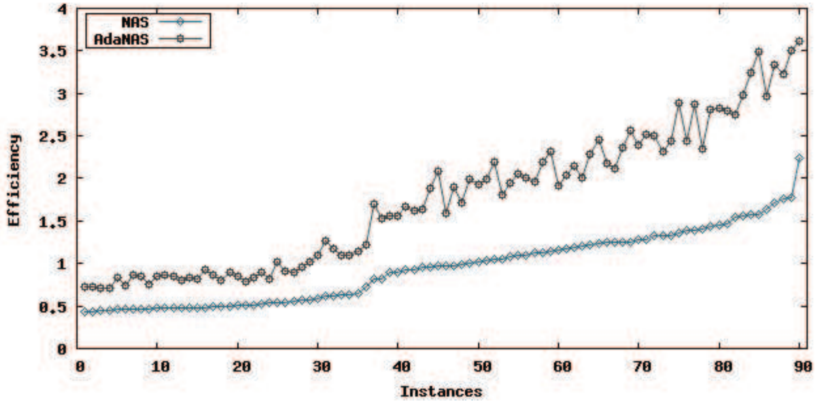


Fig. 5. Comparison between NAS and AdaNAS experimenting with 90 instances.

5. Text mining in the media

Today it is common to use computational tools to retrieve information, in fact it is an everyday and in many cases necessary. Information retrieval is performed on structured or unstructured data, IR systems commonly have recovered information from unstructured text (text without markup) while the database systems has been created to query relational data (sets of records that have values for predefined) , the principal differences between are in terms of retrieval model, data structures and query language. (Christopher et al., 2009).

According to the literature reviewed, nowadays do not exist techniques for Natural Language Processing to achieve 100% accurate results, either with the statistical approach, or the linguistic approach, in such a situation some researchers have blended both techniques (Chaudhuri et al. , 2006) (Gonzalez et al., 2007) (Vallez & Pedraza, 2007). For example, in (Sayyadian, 2004) they propose several methods to exploit structured information in databases and present a query expansion mechanism based on information extraction from structured data. The experimental results obtained show that using more structured information to expand the textual queries to improve performance in the recovery of entities in texts.

It is common that the amount of data with which one interacts is considerably larger and cannot be worked and in some cases it would be very difficult to work with these manually, in addition, these digital resources increase rapidly every day, reason by which the World Wide Web has become so popular, and is notorious as well as increased information systems. Because of this, it is very important to retrieve information efficiently (Hristidis & Papakonstantinou, 2002).

The search motor of Google, is the clearest example of how a computational tool can facilitate a user the information retrieval, unfortunately does not allow elaborate searches successfully, since it is designed mainly to operate with key words on documentary data bases; email servers are other type of tools very useful and popular.

Due to the diversity of existing digital media (heterogeneous data) has been investigated in diverse areas, as much in information retrieval as in natural language processing, whose final objective is to facilitate access to information and improve performance . In (Vallez & Pedraza, 2007) classified research areas as follows:

- The information extraction is the removal of a text or a set of texts entities, events and relationships between existing elements.
- The generation of summaries must like objective condense the most relevant information of a text. The techniques used vary according to compression rate, the purpose of summary, the genre of the text, the language (or languages) of the source texts, among other factors.
- The quest for answers can give a concrete answer to the question raised by the user, is important that the information needs to be well defined: dates, places, people, etc.
- The multilingual information retrieval consists of the possibility of recovering information although the question and/or the documents are in different languages, situation that reigns at the moment in the Web.

Automatic classification techniques Search text automatically assign a set of documents to predefined classification categories, mainly by using statistical techniques, processing and parameterization.

IR systems not only seek to identify only one object in a collection, but several items that can answer the query that satisfy user requirements, objects are usually text documents, but may be of multimedia content such as image, video or audio. For recovery to be efficient, the data are transformed into adequate representation, in addition, to answer satisfactorily the demands made by the user, the system can use various techniques and models, for example, the statistical processing that represents the classical model the information retrieval systems. In (Noy, 2006) use data mining to test their analytical approach, whereas in (Oren, 2002) use the genetic programming paradigm with satisfactory results.

In (Iskandar, 2007) “The retrieval strategy has been evaluated using Wikipedia, a social media collection that is an online encyclopedia. Social media describes the online tools and platforms that people use to share opinions, insights, experiences, and perspectives with each other. Social media can take many different forms, including text, images, audio, and video. Popular social mediums include blogs, message boards, podcasts, wikis, and blogs”, see Figure 6.

5.1 Experiments

We simulated by means of the developed tool -WREID- the expectations of successfully in a circuit of Wrestling and interests of obtain popularity based on their performance associated with specific features. One of most interesting characteristics observed in the experimental

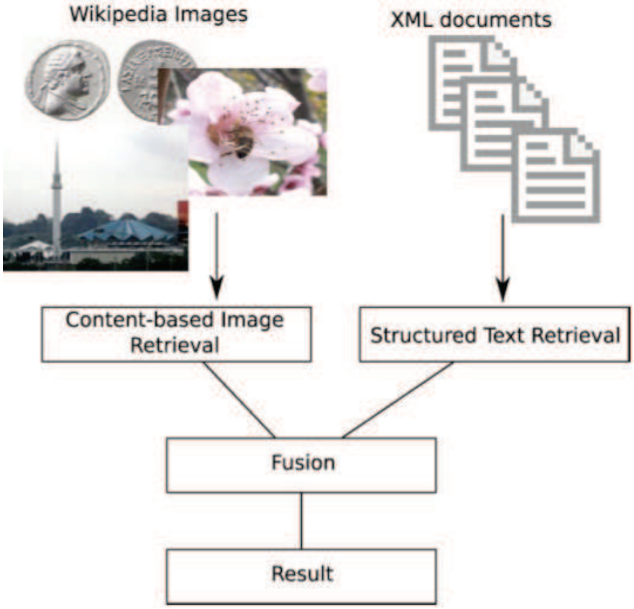


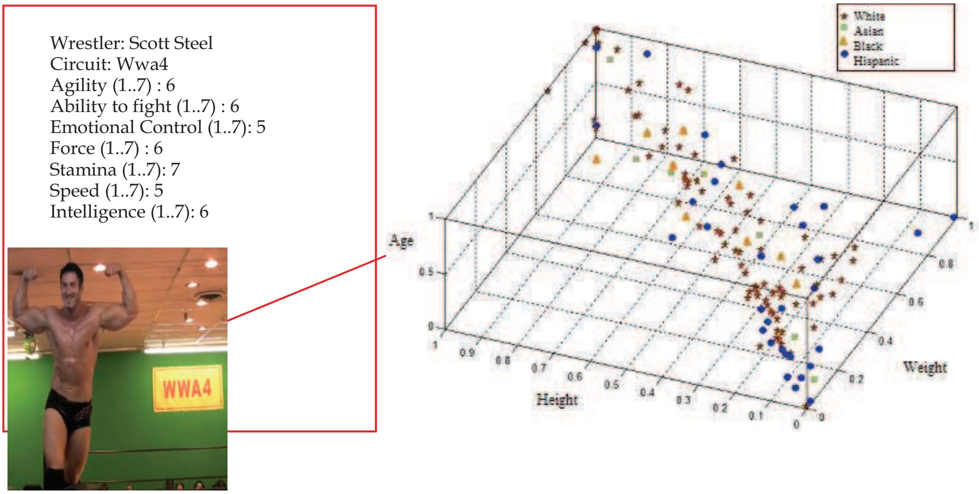
Fig. 6. Social Media Retrieval using image features and structured text

analysis were the diversity of cultural patterns established by each society because the selection of different attributes in a potential best wrestler: Agility, ability to fight, Emotional Control, Force, Stamina, Speed, Intelligence. The structured scenes associated the agents cannot be reproduced in general, so that the time and space belong to a given moment in them. They represent a unique form, needs and innovator of adaptive behavior which solves a followed computational problem of a complex change of relations. Using Social Data Mining implementing with agents was possible simulate the behavior of many followers in the selection of a best wrestler and determinate whom people support this professional career. With respect at Node attributes, we summarize the measures required to describe individual nodes of a graph. They allow identifying elements by their topological properties. The degree -or connectivity- (k_i) of a node v_i is defined as the number of edges of this node. From the adjacency matrix, we easily obtain the degree of a given node as:

$$k_i = \sum\nolimits_{j=1}^{N} a_{ij} \tag{2}$$

See examples of *k* values in figure 7. For directed graphs, we distinguish between incoming and outgoing links. Thus, we specify the degree of a node in its *indegree, ini k* , and *outdegree,* k_i^{out} . The *clustering coefficient* C_i is a local measure quantifying the likelihood that neighboring nodes of *vi* are connected with each other. It is calculated by dividing the number of neighbors of *vi* that are actually connected among them, *n*, with all possible combinations excluding autoloops, i.e., *ki(ki-1)*. Formally, we have:

$$C_i = \frac{2n}{k_i(k_i - 1)} \tag{3}$$



Wrestler: Scott Steel
Circuit: Wwa4
Agility (1..7) : 6
Ability to fight (1..7) : 6
Emotional Control (1..7): 5
Force (1..7) : 6
Stamina (1..7): 7
Speed (1..7): 5
Intelligence (1..7): 6

Fig. 7. Individual features of an element and classification of wrestling performance to a sample of 127 Wrestlers.

We first observe that Professional Wrestler Idol (support in features related with age, height and weight are considered) always plays a very significant role, which should of course not be surprising. Hidden patterns observed in the agents are related with size of circuit, match records and cultural distances (ethnicity), and the expectative of selection of a good wrestler whit specific attributes. The nodes with more value in their degree are considered more popular and obtain the best contracts. To get some insight, we run 100 regressions on 100 random samples of half the number of observations, and count the number of times each parameter affect the graph built. A Wrestler with the features similar to Scott Steel was selected as the most popular by the majority of societies because the attributes offered by it are adequate for others. In Figure 7 is shown the results of a sample of American Wrestlers.

6. Data mining with Ant Colony and Genetic Algorithm

6.1 Artificial Ant Colony

This section describes the principles of any Ant System (AS), a meta-heuristic algorithm based in the form in how the natural ants find food sources. The description starts with the ant metaphor, which is a model of this behavior. Then, it follows a discussion of how AS has evolved, and show as the ant algorithms can be applied to the Data Mining process. The Ant System was inspired by collective behavior of certain real ants (forager ants). While they are traveling in search of food, they deposit a chemical substance called pheromone on the traversed path. The communication through the pheromone is an effective way of coordinating the activities of these insects. For this reason, pheromone rapidly influences the behavior of each ant: they will choose the paths where is the biggest pheromone concentration. The behavior of real ants to search food is modeled as a probabilistic process. When there are paths without any amount of pheromone, the ants explores the neighboring area in a totally random way. In presence of an amount of pheromone, the ants follow a path with a probability based in the pheromone concentration. The ants deposit additional pheromone concentrations during his travels. Since the pheromone evaporates, the pheromone concentration in non-used paths tends to disappear slowly. The Ant System (AS) or Ant Colony Optimization (ACO) was introduced by Marco Dorigo (Dorigo, 1991). The Ant System is inspired in the natural optimization process of real ants to create paths. This type of algorithms can be applied to the solution of many combinatorial optimization problems. The artificial ants, repeat the search process to find solutions. Each ant builds a possible solution to the optimization problem. The ants share information through the pheromone, which is a common memory (global information) that can be accessed by all. The Ant System is a multi-agent system, where the ant-agents have simple behavior but the interactions between they have like result a complex behavior of the whole ant colony. They need the collaboration of whole colony to get the final objective. The AS was originally proposed to solve the Traveling Salesman Problem (TSP), and the Quadratic Assignment Problem (QAP). Now exist a lot of applications like scheduling, machine learning, data mining, and others. There are several variants of AS designed to solve specific problems or to extend the characteristics of the basic algorithm (Ochoa et al., 2010). Some of the most important variants of AS in order of appearance are. Ant Colony Optimization (ACO) was introduced initially by Dorigo (Dorigo, 1991), the Ant-Q algorithm designed by Gambardela and Dorigo (Gambardela, 1995), Max-Min Ant System algorithm (MMAS) was developed by Stützle and Hoose (Stützle, 1996), other variant of AS, named ASrank, was developed by Bullnheimer, Hartl and Strauss (Bullnheimer et al., 1997).

Actually exist some AS to solve task of Data Mining, like classified and clustering, some of this algorithms are: ANT-LGP, ANT-BASED Clustering, AntClass, Ant-Miner, others.

The maximum clique problem is a problem classified within the NP-Hard problems; this problem has real applications eg: Codes Theory, Errors Diagnosis, Computer Vision, Clustering Analysis, Information Retrieval, Learning Automatic, Data Mining, among others. Therefore it is important to use new heuristic and/or meta-heuristics techniques to try to solve this problem (Ponce et al., 2009b). The general Ant Colony Algorithm for the maximum clique problem proposed by Fenet and Solnon (Fenet and Solnon, 2003). The proposed algorithm is based on the Ant Algorithm created to solve the clique maximum; the construction process is showed in figure 8.

```
To initialize the pheromone signs
To place Ants Randomly
Repeat
For $k$ en 1..nb Ants do:
Build the clique (Solution) $C_k$
Update the pheromone signs $\{C_1, \dots, C_{nbAnts}\}$
If is the first iteration to keep in lists all the solutions without repeating no one
Else only are added to the list the solutions that not exist in the list
Until Reaching the Number of Cycles or Finding the optimum solution
```

Fig. 8. Pseudo code of Ant Clustering Algorithm.

Construction of cliques: An initial vertex is selected randomly to put an ant, and iteratively it chooses vertices to add to clique of a set of candidates (all the vertices that are connected with all vertices of the partial clique), to see figure 9.

```
Choose the first vertex randomly $v_f \in V$
$C \leftarrow \{v_f\}$
```

```
Candidates $\leftarrow \{v_i / (v_f , v_i) \in$ E$\}$
While Candidate $\neq 0$ do
    Choose a vertex $v_i \in$ Candidates with a probability p($v_i$), see Ec. (2)
    $C \leftarrow C \cup \{v_i\}$
    Candidates $\leftarrow$ Candidates $\cap \{v_j / (v_i , v_j) \in E\}$
End While
Return $C$
```

Fig. 9. Construction of Clique.

This Ant Colony Algorithm can be using to realize data clustering by the natural form that have a clique.

6.2 Genetic Algorithm with migration operator

Genetic Algorithms are algorithms that group techniques or methods based on natural evolution and genetics, taking as basis the "Theory of Evolution of Species" proposed by Charles Darwin and the discoveries made by Gregor Mendel in the field of genetics. (Holland, 1975) (Goldberg, 1989).

As in nature, the AG's evolving populations of individuals (possible solutions) usually of better quality solutions through operators for evaluation, selection, crossover and mutation. These have proved to be a good tool for solving optimization problems. Unfortunately one of its major limitations is that due to the loss of genetic diversity due to inbreeding between individuals within populations is that they tend to converge to local optima. For this reason we have proposed hybrid genetic algorithms somehow preventing the loss of diversity and achieve more efficient and fast tools.

Of these proposals are currently working largely with AG's side, where it seeks to improve the diversity of populations and their performance, this dividing both the computational load of each of the operators on different nodes for an intensification of themselves or by dividing the initial population in subpopulations that evolve individually until certain criteria laid down in that share some of the best individuals (Whitley et al., 1998) (Lu and Areibi, 2004) (Tzung-Pei et al., 2007).

Also have the AG's with immigration adapters that have a major population and a population parallel evolve independently and each number of generations are immigrants the best individuals of the population parallel to the main population (as shown in Figure 10), allowing the introduction of new genetic material in the major population allowing a greater diversity (Ornelas et al., 2009).

To evolve independently and through the parallel population has no influence from the main population evolves in a totally different which results in a process called speciation that is that genetic material that evolved independently in different conditions generates new species with very different characteristics that depend largely on the adaptive process.

The AG's with adaptive migration have been used to solve optimal route generation, water distribution networks and wastewater, design postcards, in data mining processes, among others.

These algorithms are currently used in data mining to make the process of cauterization and classification of information, and thanks to the way they work can process large volumes of information without extensive searches, which is of great importance because by the volume of information that is currently in the databases is impossible to use this type of research.

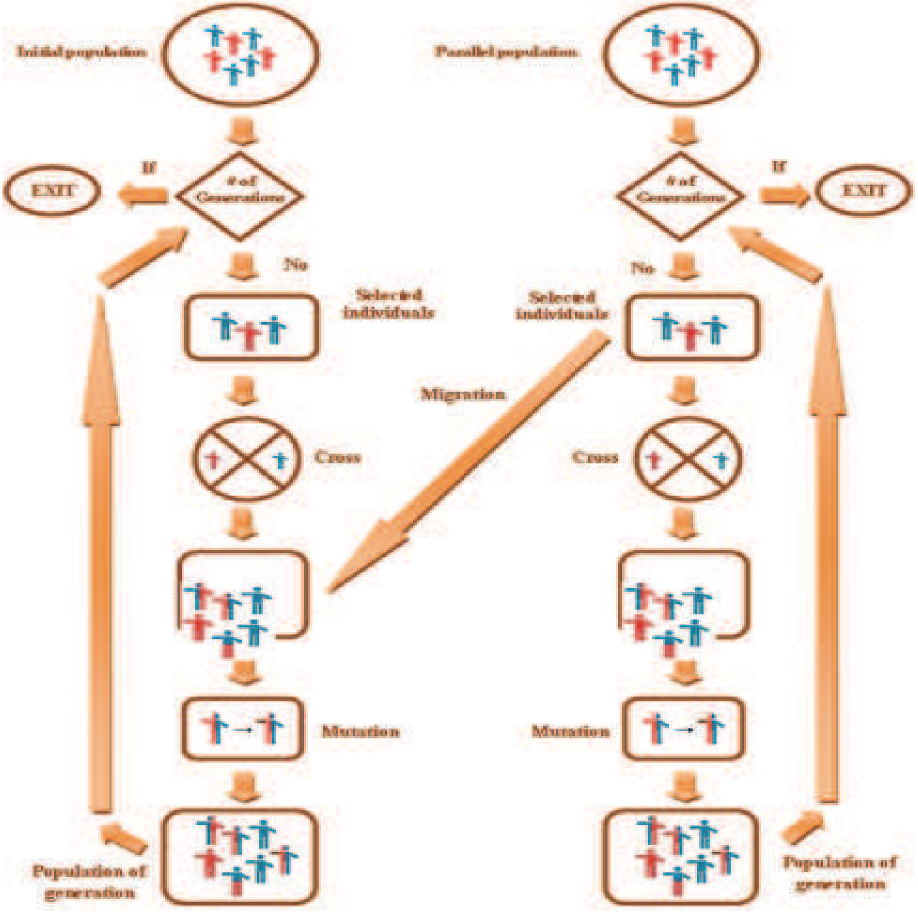


Fig. 10. Diagram of the AG's model with adaptive migration.

7. Intelligent Tutor Systems

Intelligent Tutoring Systems (ITS) are those computer systems that provide students with direct customization instructions or feedback without human intervention. ITSs were conceived around 1970, but not popularized until the 90's. They have four modules: the Interface Module, the Expert Module, the Student Module, and the Tutor Module. The Interface Module controls the communication between the student and the Intelligent Tutor System; the Expert Module contains a domain model that describes the knowledge or behavior that represents a high expert in the domain; the Student Module describes the student knowledge, behavior, etc.; and the Tutor Module is responsible for simulating the task of a teacher.

In this section, we present EDUCA, a Web 2.0 software tool to allow a community of authors and learners to create, share, and view learning materials and web resources for authoring Intelligent Tutoring Systems which combine collaborative, mobile and e-learning methods.

EDUCA applies different artificial intelligence techniques like a neural network and a genetic algorithm for selecting the best learning style or a recommendation-web mining system for adding and searching new learning resources.
Figure 11 illustrates the overall architecture of EDUCA. As we can observe, there are two authors: the main tutor (a teacher or instructor) and the community of learners. The student or learner is an important author of the course and participate actively adding learning resources to the courses. The learner has a user profile with information like the GPA, the particular learning style, or the recommended resources to the course. When the authors add learning material, they first create four different instances corresponding to four different learning styles according to Felder-Silverman Learner Style Model (Felder and Silverman, 1988). When a mobile course is exported to a mobile device, a XML interpreter is added to the course. A SCORM file for the course can also be exported. Once a course is created, a Course Publication Module saves it into a Course Repository. Whenever a learner accesses a course, a recommender system implemented in EDUCA presents links or Web sites with learning material related to the current topic. Such material is stored in a resource repository of EDUCA, which was searched previously by using Web mining techniques implemented also in EDUCA.

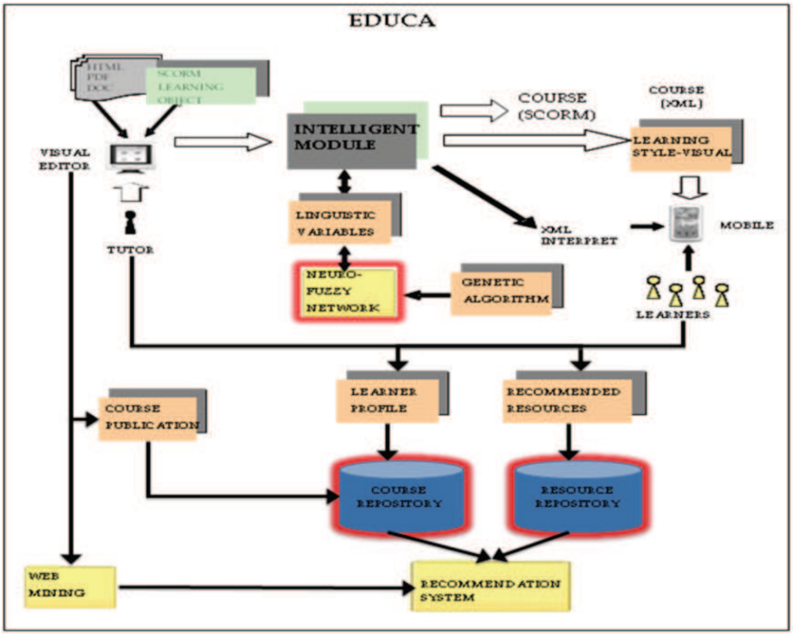


Fig. 11. EDUCA Architecture

We implemented a fuzzy-neural network using the fuzzy input values previously defined. The output of the network is the learning style for each student using a course. We also implemented a genetic algorithm (Bucket Sort) for the optimization of the weights used in the network. The network was trained for 800 generations using a population of 150 chromosomes. In order to train the network, we created three set of courses for high school students. Each course was presented in four different teaching styles according to the

Felder-Silverman model. When a mobile course is exported to a mobile device, a XML interpreter is added to the course. A SCORM file for the course can also be exported. Once a course is created, a Course Publication Module saves it into a Course Repository. Whenever a learner accesses a course, a recommender system implemented in EDUCA presents links or Web sites with learning material related to the current topic. Such material is stored in a resource repository of EDUCA, which was searched previously by using Web mining techniques implemented also in EDUCA.

We tested the tool with 15 professors/teachers and their respective students of different teaching levels. They developed different kinds of courses like a GNU/Linux course, a Basic Math Operation course, and learning material for preparation to the Mexico's Admission-Test for College EXANI-II. The students participated by reading, evaluating and adding material (Web resources) to the courses. Next, we present an example of how an author creates/updates learning material for a Basic Math course (figure 12). We first create the structure of the course (left-top). Then, we add learning material for each learning style (right-top and left-bottom). In this stage, we also assign fuzzy set values to each linguistic variable, and use recommended and actual resources for inclusion in the course. Last, we export and display the course (right-bottom).

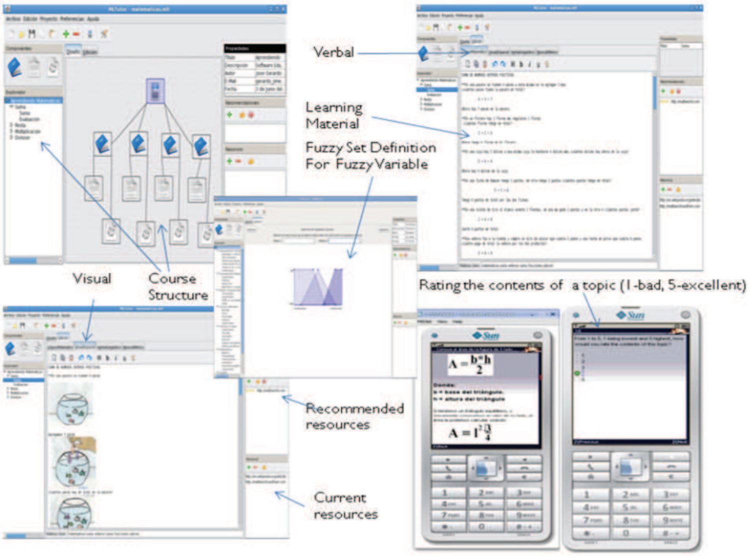

Verbal
Learning Material
Fuzzy Set Definition For Fuzzy Variable
Visual
Course Structure
Rating the contents of a topic (1-bad, 5-excellent)
Recommended resources
Current resources

Fig. 12. Authoring Learning Material

8. Conclusion and the future research

Nowadays exist a lot of applications in real life problems, where is possible used data mining to analyse data base to obtain important information in different areas, in this chapter was present some algorithms and applications that us data mining such as like Electrical Power Design, Trash Collectors Routes, Fraud Analysis, Vehicle Routing Problem, Text Mining in the Media, Intelligent Tutor Systems, Ant Colony Optimization, Genetic Algorithms, Particle Swarm Optimization and Web Mining Techniques.

As shown there are multiple areas in which data mining can be used to retrieve information that is not easy to detect with the naked eye using different tools and algorithms.

We describe how decision trees work where structures are used if-then and allow the creation of recommender systems to facilitate decision making, such as diagnostic system for identifying electrical signals the device occurrence, related to physical phenomena and provide a quick and better solution to the problem presented.

For the problem of garbage collection to do a catheterization to determine how best to plan it based on the type of waste, areas collection, type and number of vehicles used for this purpose, among others, using algorithms such as Ant Colony Optimization, Genetic Algorithms and Particle Swarm Optimization.

Once clustered can use these same tools to generate optimal routes that shorten the distance travelled, fuel consumption, deterioration of vehicles, among others.

The methodologies for the detection of fraud have their own strengths and weaknesses characteristics. The overall strength of FDS using anomaly detection is the adaptability to new patterns fraudsters, in the particular case of this study is strengthened with the application of hybridization clustering processes giving a greater dynamism to the system and making it look like a promising component within the fraud detection systems with potential advantages in regard to: upgrade and management of the heterogeneity of customers and their transactions, achieving a better accuracy in the results, and greater dynamism in the system.

Additionally, the multi-objective approach place it in a better position compared to other systems, due to the characteristics of fraud detection problem where there are several factors to consider for best results.

For the solution of SQRP, we proposed a novel algorithm called AdaNAS that is based on existing ant-colony algorithms. This algorithm incorporates parameters adaptive control techniques to estimate a proper TTL value for dynamic text query routing. In addition, it incorporates local ruler that take advantage of the environment on local level, three functions were used to learn from past performance. This combination resulted in a lower hop count and an improved hit count, outperforming the NAS algorithm. Our experiments confirmed that the proposed techniques are more effective at improving search efficiency. Specifically the AdaNAS algorithm in the efficiency showed an improvement of the 81% in the performance efficiency over the NAS algorithm.

Using Social Data Mining in Media Richness we improve the understanding of change for the best paradigm substantially, because we classify the communities of agents appropriately based on their related attributes approach, this allows determine a “American Wrestler Idol” which exists with base on the determination of acceptance function by part of the remaining communities to demonstrate best performance. Each year 7000 new wrestlers arrive to different American Wrestling Circuits. Social Data Mining offers a powerful alternative for optimization problems, for that reason it provides a comprehensible panoramic of the cultural phenomenon (Ochoa et al., 2006). This technique lead us about the possibility of the experimental knowledge generation, created by the community of agents for a given application domain. How much the degree of this knowledge is cognitive for the community of agents is a topic for future work. The answer can be similar to the involved in the hard work of communication between two different societies and their respective perspectives. A new Artificial Intelligence that can be in charge of these systems, continues being distant into the horizon, in the same way that we still lack of methods to understand the original and peculiar things of each society.

As future work is to continue working with various tools and algorithms that allow us to improve data mining and this allowed us to knowledge based on information extracted from databases (information that can not be extracted directly and that features not visible to the naked eye) to improve many existing systems and create developments that take into account factors that so far can not be displayed using other tools.

Applied the models proposed in several areas for example establishing the need for FDS to be increasingly proactive in order to adapt to the greatest extent possible so changing the behaviour presented by fraudsters or in singers of Mexican Society and determine the possible "New Musical Idols or Bands" where only 27% record their second album, this for different genders according theirs profiles, the principal problem is the confidentially of this information and its use for this propose.

9. References

Amaral, L. and Ottino, J. (2004) Complex systems and networks: Challenges and opportunities for chemical and biological engineers. *Chemical Engineering Scientist*, 59:1653–1666.

Basu, S.; Bilenko, M. and Mooney R. (2004). A probabilistic framework for semi-supervised clustering. *Proceedings of the Tenth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*. Seattle, WA : ACM, press, pp. 59-68.

Breiman, L.; Friedman, J. and Olshen, R. (1984). Classification and Regression Trees, *Wadsworth International Group*. Belmont, CA.

Bullnheimer, B.; Hartl, R. and Strauss C. (1997). A New Rank Based Version of the Ant System: A Computational Study, *Technical report*, Institute of Management Science, University of Vienna, Austria, 1997.

Chandola, V.; Banerjee A. and Kumar, V. (2009). Anomaly detection: A survey. *Journal of Computing Surveys*. ACM. pp. 1-58.

Chaudhuri, S.; Das, G.; Hristidis, V. and Weikum, G. (2006). Probabilistic information retrieval approach for ranking of database query results, *ACM Trans. Database Syst.*, 31(3), pp. 1134-1168

Cruz, L.; Gómez, C.; Aguirre, M.; Schaeffer, S.; Turrubiates, T.; Ortega, R. and Fraire, H.(2008). NAS algorithm for semantic query routing systems in complex networks. *In DCAI*, volume 50 of Advances in Soft Computing, pages 284–292. Springer.

Deb, K. (2001). Multi-objective optimization using evolutionary algorithms, Book Chichester, Uk : John Wiley and Sons.

Dehuri, S. and Cho, S.B. (2009). Multi-criterion Pareto based particle swarm optimized polynomial neural network for classification: A review and state-of-the-art, *Journal of Computer Science Review*. pp. 19-40.

Dorigo, M. (1991). Positive Feedback as a Search Strategy. *Technical Report*. No. 91-016. Politecnico Di Milano, Italy.

Felder, R. and Silverman, L. (1988). Learning and Teaching Styles In Engineering Education, *Journal of Engineering Education*. North Carolina State University and Institute for the Study of Advanced Development.. 78(7), pp. 674_681.

Fenet, S. and Solnon, C. (2003) Searching for Maximum Cliques with Ant Colony Optimization, *EvoWorkshops 2003*, LNCS 2611, 236–245.

Gama, J. and Medes, P. (2005) Learning decision trees from dynamic data streams. *Journal of Universal Computer Science*.

Gambardella, L.M. and Dorigo M.(1995). Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. *Proceedings of ML-95, Twelfth International Conference on Machine Learning, Tahoe City*, CA, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, pp. 252-260.

Goldberg, D.(1989). Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison Wesley*. ISBN: 0-201-15767-5..

González, J.J.; Pazos, R.; Gelbukh, A.; Sidorov, G.; Fraire, H. and Cruz, I. (2007). Prepositions and Conjunctions in a Natural Language Interfaces to Databases, *Lecture Notes in Computer Science*, Vol. 4743, pp. 173-182.

Grosan, C. and Abraham, A.(2007) Hybrid evolutionary algorithms: methodologies, architectures, and reviews. Hybrid evolutionary algorithms. Book auth. Grosan C., Abraham A. and Ishibuchi H.. - Berlin : Springer Verlag-Heidelberg.

Han, J. and Kamber, M. (2006). Data Mining, Concepts and Techniques. *Morgan Kaufmann Publishers is an imprint of Elsevie*r. ISBN 13: 978-1-55860-901-3, ISBN 10: 1-55860-901-6.

Holland J. (1975). Adaptation in Natural and Artificial Systems An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. *The University of Michigan Press*.

Hristidis, V. and Papakonstantinou, Y. (2002). Discover: Keyword Search in Relational Databases, *VLDB '02: Proc. of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, pp. 670-681.

Iskandar, D.; Pehcevski, J.; Thom, J. and Tahaghoghi, S. (2007). Social Media Retrieval using Image Features and Structured Text, *In N. Fuhr, M. Lalmas, and A. Trotman (eds).*

Janikowo, C. (2008) Fuzzy decision trees: Issues and methods. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics. 28-1. 1.14.

Juszczak, P.; Adams, N.; Hand, D.; Whitrow, C. and Weston, D. (2008). Off-the-peg and bespoke classifiers for fraud detection. *Computational Statistics & Data Analysi* –Vol. 52. pp. 4521-4532.

Jin, R. and Agrawal, G. (2003) Efficient decision tree construction on streaming data. *In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 571 576.

Kou, Y.; Sirwongwattana, C. and Huang, S. (2004). Survey of fraud detection techniques. *IEEE International Conference on Networking, Sensing and Control*. Taipei : IEEE press, pp. 749-754. ISSN: 1810-7869. Print ISBN: 0-7803-8193-9.

Liu, L.; XiaoLong, J. and Kwock, C. (2005). Autonomy oriented computing — from problem solving to complex system modeling. *In Springer Science* + *Business Media Inc*, pages 27–54.

Llora, X. and Wilson, S. (2004). Mixed Decision Trees: Minimizing Knowledge representation bias in LCS. *Genetic and Evolutionary Computation. GECCO*. Lecture Notes in Computer Science –Vol. 3103/2204. pp. 797 809.

Lozano, M. and García, C. (2010). Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *In Journal of Computers and Operations Research*. pp. 481-497.

Lu, G. and Areibi, S.(2004). An Island-Based GA Implementation for VLSI Standard-Cell Placement. *In GECCO 2004*. K. Deb et al. (Eds.), LNCS 3103, pp. 1138–1150, Springer-Verlag, 2004.

Manning, C.; Raghavan, P. and Schütze, H. (2009). An Introduction to Information Retrieval, *Cambridge University Press*, Cambridge, England, pag.195.

Márquez, M. Y. (2009). Determinación de perfiles de generación de RSD por tipología familiar a través de minería de datos: Estudios de casos en tres comunidades de Mexicali, B. C. *Tesis Doctoral*. UABC.

Michlmayr, E. (2007). Ant Algorithms for Self-Organization in Social Networks. *PhD thesis*, Vienna University of Technology.

Mihail, M.; Saberi, A. and Tetali, P.(2004). Random walks with lookahead in power law random graphs. Internet Mathematics, 3, 2004.

Mitchell, T. (1997). Machine Learning. McGraw Hill.

Noy, A.; Raban, D. and Ravid, G. (2006). Testing Social Theories in CMC through Gaming and Simulation. *Journal of Simulation and Gaming*, 37(2), pp. 174-194.

OECD (2008) Household Behaviour and the Environment.

Ochoa, A.; Hernández, A.; Cruz, L.; Ponce, J.; Montes, F.; Li, L. and Janacek, L. (2010) New Achievements in Evolutionary Computation, *Book edited by: Peter Korosec*, ISBN 978-953-307-053-7, pp. 318, INTECH, Croatia, downloaded from SCIYO.COM

Ochoa, A.; Sehr, M.; Sarchimelia, M.; Meriam, G. et al. (2006). Italianitá: Discovering a Pygmalion effect on Italian Communities Using Data Mining. *In Proceedings of CORE'2006*.

Oren, N. (2002). Improving the effectiveness of Information Retrieval with Genetic Programming, *MSc research report*, University of the Witwatersrand, South Africa.

Ortega, R.(2009) Estudio de las Propiedades Topológicas en Redes Complejas con Diferente Distribución del Grado y su Aplicación en la Búsqueda de Recursos Distribuidos. *PhD thesis*, Instituto Politécnico Nacional, México.

Ornelas, F.; Padilla, A.; Padilla F.; Ponce de León E. and Ochoa, A. (2009) Genetic Algorithm using Migration and Modified GSX as Support. *Artificial Intelliigence & Applications, Book Edited by: A. Gelbukh*, ISBN 978-607-95367-0-1, pp. 21-28, SMIA, Mexico.

Pedrycz, W. and Sosnowski (2005). Genetically optimized fuzzy decision trees. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*. pp. 633- 641.

Perez, J.; Muguerza, J.; Arbelaitz, O.; Gurrutxaga, I. and Martin, J. (2007). Combining multiple class distribution modified subsampled in a single tree. *Pattern Recognition Letters.* pp. 414-422.

Ponce, J.; Hernández, A.; Ochoa, A.; Padilla, F.;Padilla A.; Álvarez, F. and Ponce de León, E. (2009a). *Data Mining in Web Applications. Data Mining and Knowledge Discovery in Real Life Applications book*. Edited by Julio Ponce and Adem Karahoca. ISBN 978-3-902613-53-0, 436 pages

Ponce , J.; Padilla, F.; Ochoa, A.; Padilla, A.; Ponce de León, E. and Quezada, F. (2009b). Ant Colony Algorithm for Clustering through of Cliques, *Artificial Intelligence & Applications, A. Gelbukh (Ed.),*ISBN: 978-607-95367-0-1, pp. 29-34, November 2009, Mexico.

Quinlan, J. (1993). C4.5: Programs for Machine Learning. *Morgan Kaufmann,* San Mateo, CA.

Raidl, G. (2006). A unified view on hybrid metaheuristics. *In Proceedings of Hybrid Metaheuristics, Third International Workshop.* Berlin : Springer Verlag, pp. 1-12.

Rokach, L. and Maimon, O.(2005) Top-down induction of decision trees Classifiers - a survey. *IEEE Transactions on Systems, Man and Cybernetics*. Reviews - Vol. 35-4. pp. 476-487. ISSN: 1094-6977.

Ruiz, R.; Guzman, A. and Martinez J. (1999) Enfoque Lógico Combinatorio al Reconocimiento de Patrones. Instituto Politecnico Nacional, 1999.

Safavian, S. and Landgrebe, D. (1991) A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man and Cybernetics*. pp. 660 674.)

Sánchez, D.; Vila, M.; Cerda, L. and Serrano, J. (2009). Association rules applied to credit card fraud detection. *Expert Systems with Applications: An International Journal* – Vol. 36, pp. 3630-3640. ISSN:0957-4174.

Sayyadian, M.; Shakery, A.; Doan, A. and Zhai, C. (2004). Toward Entity Retrieval over Structured and Text Data, *In Proc. of ACM SIGIR 2004 Workshop on Information Retrieval and Databases*.

Shih, Y. and Loh, W. (1997) Split Selection Methods for Classification trees. Statistica Sinica, pp. 815-840.

Shou Chih, C., Hsing Kuo, P. and Yuh Jye, L.(2005) Model Trees for Classification of hybrid data types. *In Intelligent Data Engineering and Automated Learning - IDEAL: 6th International Conference*. pp. 32-39.

Stolfo, S.; Fan, D.; Lee, W.; Prodromidis, A. and Chan, P. (1997). Credit Card Fraud Detection Using Metalearning: *Issues and Initial Results. AAAI Workshop AI Methods in Fraund and Risk Management*. Columbia : AAAI Press, pp. 83-90.

Stolfo, S.; Fan, D.; Lee, W.; Prodromidis, A. and Chan P. (2000). Cost-Based Modeling for Fraud and Intrusion Detection: Results from the JAM Project. *DARPA Information Survivability Conference & Exposition* – Vol. 2. Hilton Head : IEEE Press, pp. 130-144. ISBN: 0-7695-0490-6.

Stützle, T. and Hoos, H. H.(1996). Improving the Ant System: A detailed report on the MAXMIN Ant System. *Technical report AIDA-96-12*, FG Intellektik, FB Informatik, TU Darmstadt.

Tempich, C.; Staab, S. and Wranik, A. (2004). REMINDIN': Semantic Query Routing in Peer-to-Peer Networks based on Social Metaphers, *In 13th World Wide Web Conference (WWW)*.

Tzung–Pei, H.; Wen-Yang, L.; Shu-Min, L. and Jiann-Horng L. (2007). Dynamically Adjusting Migration Rates for Multi-Population Genetic Algorithms. *Journal of Advanced Computational and Intelligent Informatics.* Vol. 11, No. 4, pp. 410-417.

Utgoff, P. (1989) Incremental induction of decision trees. Machine Learning. pp. 161-186.

Utgoff, P.; Berkman, N. and Clouse, J. (1997). Decision tree induction based on efficient tree Restructuring. *Machine Learning.* 5-44.

Utgoff, P. and Brodley, C. (1990). An Incremental Method for Finding multivariate splits for decision trees. *In Proc.7th International Conference on Machine Learning*. pp. 58-65.

Utgoff, P. and Brodley, C. (1995). Multivariate decision trees. *Machine Learning.* pp. 45-77.

Vallez, M. and Pedraza-Jimenez, R. (2007). Natural Language Processing in Textual Information Retrieval and Related Topics, [on line]. "Hipertext.net", num. 5, 2007. <http://www.hipertext.net> [Consulted: 07/15/10]. ISSN 1695-5498

Vanichsetakul, N. and Wei-Yin, L. (1988). Tree-Structured Classification via Generalized Discriminant analysis. *Journal of the American Statistical Association,* pp. 715 728.

Wang, X.; Nauck, D. and Spott, M.(2007). Intelligent data analysis with fuzzy decision trees. Soft Computing - A Fusion of Foundations, Methodologies and Applications. pp. 439 457.

Whitley, D.; Rana, S. and Heckendorn (1998). The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. *Journal of Computing and Information Technology,* Vol. 7, pp. 33-47, Colorado State University.

Whitrow, C.; Hand, D.; Juszczak, P.; Weston, D. and Adams, N. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Journal of Data Mining and Knowledge Discovery*. pp. 30-55.

Wu, C.-J.; Yang, K.-H. and Ho.(2006). AntSearch: An ant search algorithm in unstructured peer-to-peer networks. *In ISCC,* pages 429–434.

ISBN 978-953-307-154-1
Hard cover, 442 pages
Publisher InTech
Published online 21, January, 2011
Published in print edition January, 2011

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

Project Ranking-Based Portfolio Selection Using Evolutionary Multiobjective Optimization of a Vector Proxy Impact Measure

S. Samantha Bastiani[1], Laura Cruz[1], Eduardo Fernández[2], Claudia Gómez[1], Victoria Ruiz[1]

[1]Madero Institute of Technology, Ciudad Madero, México
[2]Autonomous University of Sinaloa, Culiacan, México

cggs71@hotmail.com, lcruzreyes@prodigy.net.mx, eddyf@uas.edu.mx, b_shulamith@hotmail.com, victoria.rzrz@gmail.com

Abstract

Selecting project portfolios Decision-Maker usually starts with limited information about projects and portfolios. One of the challenges involved in analyzing, searching and selecting the best portfolio is having a method to evaluate the impact of every project and portfolio in order to compare them.

This paper develops a model for composing public-oriented project portfolios. Information concerning the quality of the projects is in the form of a project-ranking, which can be obtained by the application of a proper multi-criteria method; however the ranking does not assume an appropriate evaluation. A best portfolio is primarily found through a multi-objective optimization that regards the impact indicators that reflect the quality of the projects in the portfolio and competent portfolios' cardinalities. Overall good solutions are obtained by developing an evolutionary method, which is found to perform well in some test examples.

Keywords: Project portfolio selection; Multi-objective optimization; Multi-criteria analysis

1. Introduction

Project portfolio selection is one of the most difficult, yet most important decision-making problems faced by many organizations in government and business sectors. To carry out the project selection, the decision maker usually starts with limited information about projects and portfolios. His/her time is often the most critical scarce resource. In multiple situations the decision maker feels more comfortable employing simple decision procedures, because of lack of available information, lack of time, aversion to more elaborated decision methods, and even because of his/her fondness for established organizational practices. Cooper et al. ([1]) argues about popularity of scoring and ranking methods in R&D project management in most American enterprises.

Methods of scoring and ranking are used by most of the government organizations that fund R&D projects. Usually, methods for scoring, ranking or evaluating projects contain some way of aggregating multi-criteria descriptions of projects (e.g. [2]). Validity of these methods depends on how accurately ranking and scores reflect decision maker preferences over portfolios. In fact, the portfolio's score should be a value function on the portfolio set, but this requires a proper elicitation of decision maker preferences inside the portfolio's space.

Ranking is also used in problems where a "Participatory budgeting" is involved. "Participatory budgeting" can be defined as a public space in which government and society agree on how to adapt priorities of citizenship to public policy agenda. The utility of these participatory exercises is that the government obtains information about priorities of the participating social sectors, and might identify programs with a consensual benefit.

Ranking of public actions given by the participants is an expression of their preferences on projects, not on portfolios. Let us assume that a method of integrating the individual ranking on a collective order is applied, as the Borda score or a procedure based on the exploitation of collective fuzzy preference relations (e.g. [3, 4, 5]). With the obtained group order, the decision maker has more information about social preferences regarding the different actions to budget. The decision maker should use that information to find the best portfolio.

This paper proposes a new model for project portfolio selection, which makes use of the ranking of a set of projects according to the preferences of a decision maker. The model is formulated by a set of indirect indicators that reflect the impact of the portfolio in terms of the number of projects and the positions they occupy in the ranking. This paper is structured as follows: the background is briefly described in the second section. It is also shown the algorithm which lead to optimize the proposed impact model. Section 3 presents such model, followed by a description of the solution algorithm (Section 4). Finally in Section 5 we give empirical evidence that supports our results and some conclusions are given in Section 6.

2. Background

A project is a temporary process, singular and unrepeatable, pursuing a specific set of objectives ([6]). In this paper, it is not considered that the projects can be divided into smaller units, such as assignment or activities.

A portfolio consists of a set of projects that can be performed in the same period of time ([6]). Due to that, projects in the same portfolio can share the available resources of the funding organization. Therefore, it is not sufficient to compare the projects individually; the decision maker should compare project groups to identify which portfolio makes a major contribution to the objectives of the organization.

Selecting portfolios integrated by properly selected projects is one of the most important decision problems for public and private institutions [7, 8]. The Decision Maker (DM) (a person or a collective entity) is in charge for selecting a set of projects to be supported ([9]).

2.1. Related works

Gabriel et al. ([10]) proposed an additive function as a portfolio's score. This function aggregates the rank of projects. The simplest model is to assign project priorities in correspondence to the project rank (the highest priority is assigned to the best ranked project and so on). The portfolio's score is the sum of priorities associated with its projects. 0-1 mathematical programming is used to maximize the score.

Mavrotas et al. ([11]) proposed an additive function depending on a project's augmented score. This augmented score is built according to the project's specific rank. The augmented score of a project A holds that no combination of projects with worse ranking positions and a lower total cost can have a score bigger than A. The augmented score is obtained by solving a knapsack problem for each project. The portfolio's score is the sum of its projects' augmented scores.

Validity of these methods depends on how accurately the ranking scores reflect the decision maker preferences over portfolios. In fact, the portfolio's score should be a value function on the portfolio set, but this requires a proper elicitation of the decision maker preferences in the portfolio's space. In order to illustrate limitations of those methods, consider the following example: Let us suppose a 20-project strict ranking; priority 20 is assigned to the best project; 19 is assigned to the second one; 1 is assigned to the worst ranked project. Considering a score given by the sum of priorities, the portfolio containing the best and the worst projects (score = 21) should be indifferent to the portfolio containing the second best project and the second-to-last one (score = 21). The DM could hardly agree with such a proposition.

It is necessary to compare impact of possible portfolios in order to find the best one. The information provided by the simple project ranking is very poor for portfolio optimization purposes. Hence, some proxy impact measures should be defined. This problem was approached by [12] under the assumption that "the portfolio impact on a decision maker mind is determined by the number of supported projects and their particular rank". If project A is clearly better ranked than B, then A is admitted to have "more social impact" than B.

The DM should consider this information from the ranking. The appropriateness of a portfolio is not only defined by the quality of the included projects, but also by the amount of contained projects. The purpose is to build a good portfolio by increasing the number of supported projects and controlling the possible disagreements regarding decision maker preferences, which are assumed as incorporated in input ranking. For Fernandez and Olmedo ([12]), a discrepancy is the fact that given a pair of projects (A,B) (being B worse ranked than A), B belongs to portfolio and A does not. Different categories of discrepancy are defined according to the relative rank of the concerning projects. Some discrepancies might be acceptable between the information provided by the ranking and the decisions concerning the approval (hence supporting) of projects, whenever this fact increases the number of projects in the portfolio. However, this inclusion should be controlled because the admission of unnecessary discrepancies is equivalent to underestimating the ranking information. A multi-objective optimization problem is solved by using NSGA-II, in which the objective functions are the number of supported projects and the number of discrepancies (separately in several functions, in regard to the importance of each kind of discrepancy) ([12]).

Main drawback: a portfolio quality measure model based solely on discrepancies and in the number of supported projects is highly questionable; more information is required about project impacts. If decision maker thinks in terms of priority and relatively important projects, their numbers and ranks should be considered.

2.2. NSGA-II (Non-dominated Sorting Genetic Algorithm-II)

The research problem implicates the use of techniques of multi-objective optimization, particularly multi-objective evolutionary algorithms (MOEAs). One advantage of these algorithms is its capacity of handling problems with an exponential complexity. Other advantage is their ability to generate an approximation to the Pareto optimal set in a single run instead of having to perform many runs as in conventional multi-objective optimization. Several works have reported successful results with this kind of algorithms ([12]).

One of the most used algorithms for solving multi-objective problems is the NSGA-II (Non-dominated Sorting Genetic Algorithm), which has gained much popularity solving problems efficiently. It is shown in Figure 1.

Fig 1. Structure of the algorithm NSGA-II.

```
$R_t= P_t \cup Q_t$
F= fast-nondominated-sort ($R_t$)
until $|P_{t+1}|< N$
  crowding-distance-assignment ($F_i$)
$P_{t+1}=P_{t+1} \cup F_i$
Sort ($P_{t+1}, \geq_n$)
$P_{t+1}=P_{t+1}[0:N]$
$Q_{t+1}$=make-new-pop($P_{t+1}$)
t=t+1
```

The procedure "Fast non-dominated sorting" (shown in Figure 2), optimizes the algorithm NSGA-II.

Finally this algorithm has a diversity indicator whose evaluation is shown in Figure 3. This indicator favors solutions in less populated regions of the search space; these solutions will be advantaged by the selection mechanism ([13]).

Fig 2. Structure of the Fast-nondominated-sort procedure

```
for each $p \in$ P
for each $q \in$ P
        if($p < q$) then
            $S_p= S_p \cup \{q\}$
        else if($q < p$) then
            $n_p = n_p +1$
    if $n_p$=0 then
            $\mathcal{F}_1=\mathcal{F}_1 \cup \{p\}$
  $i$=1
  while $\mathcal{F}_i \neq 0$
        $\mathcal{H} = 0$
        for each $p \in \mathcal{F}_i$
        for each $q \in \mathcal{S}p$
                $n_q= n_q -1$
        if $n_q = 0$ then $\mathcal{H} = \mathcal{H}\{q\}$
       $i=i+1$
       $\mathcal{F}_i = \mathcal{H}$
```

Fig 3. Structure of the algorithm of the Crowding-distance-assignment (I).

```
$l$= |I|
for each i, set I $[i]_{distance}$=0
for each objective m
          I=sort (I, m)
             for i =2 to($l$-1)
I $[i]_{distance}$ = I $[i]_{distance}$ + (I [i +1]).m –
I [i-1].m)
```

3. The proposed model

The new model overcomes the idea proposed in [12, 14]. In this model the optimization is performed over indicators, which positively provide indirect information on the impact of the portfolio. This model handles three categories for projects: priority, satisfactoriness and acceptability, besides incorporating a ranking in descending order.

Once data has been established, solution sets are evaluated through a set of indicators of impact that form the model proposed in this paper. The following functions are defined:

$$I_1(\vec{x}) = \sum_{i=1}^{n} x_i F(i,1) \qquad (1)$$

$$F(i,1) = \begin{cases} 1 & \text{if } i \in G_1 \\ 0 & \text{Otherwise} \end{cases}$$

where the binary variable x_i indicates whether the ith project belongs to the portfolio or does not. That is, $x_i = 1$ if the ith project belongs to portfolio; otherwise $x_i = 0$. Note that Function I_1 counts how many projects belonging to the priority category (Group 1) are contained in the portfolio.

$$I_2(\vec{x}) = \sum_{i=1}^{n} x_i (n-i) F(i,1) \qquad (2)$$

where x_i (n-i) is a value that reflects the rank order of the supported ith project. I_2 increases with the rank ordering of the supported projects of the priority category. This function measures (in proxy way) how good the supported priority projects are.

$$I_3(\vec{x}) = \sum_{i=1}^{n} x_i F(i,2) \qquad (3)$$

$$F(i,2) = \begin{cases} 1 & \text{If } i \in G_2 \\ 0 & \text{Otherwise} \end{cases}$$

where the binary variable x_i has the same above meaning. Note that Function I_3 counts how many projects belonging to the satisfactory category (Group 2) are contained in the portfolio.

Besides

$$I_4(\vec{x}) = \sum_{i=1}^{n} x_i (n-i) F(i,2) \qquad (4)$$

measures (in proxy way) how good the supported satisfactory projects are.

Similarly we define

$$I_5(\vec{x}) = \sum_{i=1}^{n} x_i F(i,3) \qquad (5)$$

$$F(i,3) = \begin{cases} 1 & \text{If } i \in G_3 \\ 0 & \text{Otherwise} \end{cases}$$

Function I_5 counts how many projects belonging to the acceptable category (Group 3) are contained in the portfolio.

Finally

$$I_6 = \sum_{i=1}^{n} x_i \qquad (6)$$

represents the portfolio cardinality.

We assume that the DM "feels" the potential impact of the portfolio in terms of the numbers of projects for category and the positions they occupy.

The best portfolio should be the best solution of the multi-objective problem:

$$\underset{C \in R_F}{Max}\ (I_1, I_2, I_3, I_4, I_5, I_6) \qquad (7)$$

where R_F is the feasible region determined by budgetary constraints.

In this case the DM, based on his/her preferences, should select the best portfolio.

4. The proposed algorithm

The algorithm developed in this research work is called Evolutionary algorithm for Solving the public Portfolio problem from Ranking Information (ESPRI). It is inspired by the NSGA-II algorithm developed by Deb et al. ([13]), which successfully manages exponential complexity ([12]). ESPRI uses the indicator vector from Equation 7 for evaluating solutions.

To illustrate ESPRI algorithm process, a set of n projects is taken as example, with its respective total budget as well as necessary budget for each project. Previously, such projects were ranked according to decision maker preferences. Heuristically, the projects were separated in three categories: priority, satisfactoriness and acceptability. Once this process is complete, the algorithm generates random portfolios, which form the NSGA-II initial population

Later, the following procedures are applied: fast-non-dominated, crowding distance and genetic operators. Finally, the algorithm shows the found non-dominated solutions for the decision maker. Figure 4 shows ESPRI algorithm.

Fig 4. Structure of the algorithm

```
$R_t= P_t \cup Q_t$
quality assessment: Impact Indicators Model.
F= fast-nondominated-sort ($R_t$)
while: |Pt+1|< N
   crowding-distance-assignment ($F_i$)
Pt+1=Pt+1$\cup$ $F_i$
Sort (Pt+1, $\geq$n)
Pt+1=Pt+1[0:N]
Qt+1=Create-new-pop(Pt+1)
t=t+1
```

5. Computational Experiments

This section describes conducted experiments with the proposed evolutionary algorithm (ESPRI).

The aim of this experiment is to study indicator new model capacity, as well as to compare ESPRI solutions against the state of the art.

5.1. Experimental Environment

The following configuration corresponds to experimental conditions required for tests described in this paper:

1. Software: Operating System, Mac OS X Lion 10.7.5 (11G63b) Java Programming Language, Compiler NetBeans 7.2.1.
2. Hardware: computer equipment, Intel Core i7 2.8 GHz CPU and 4 GB of RAM.
3. Instances: An instance used for this study was taken from the state of the art, reported by Fernández et al. in [12, 14].
4. Performance Measure: In this case the performance is measured through the aforementioned six objectives (Eq. 7).

5.2. An illustrative example

Within the public portfolio problem, an instance for ranking strategy is formed by four attributes: Id, total amount to be distributed, project cost and ranking. The test example is taken from the state of the art, ([12]), which works with an instance that consists of 100 projects. The projects are separated into three categories: priority, satisfactoriness and acceptability, approximately uniform.

For the experiment, ESPRI algorithm was run 20 times; in each run 200 iterations were performed. The experiment reported was held with an instance of 100 projects with a total amount of 2.5 billion to be distributed; this instance can be seen in Table 1.

The ESPRI algorithm was set as: one-point crossover, Mutation probability = 0.5, population size = 200 and Number of Generations = 100.

Note that if the resources were distributed strictly following the ranking order, the resulting portfolio would have 22 projects, all belonging to the priority category.

Table 2 shows a representative sample of the approximation to Pareto frontier, which our proposal might reach. Red marks represent a set of solutions preferred by a decisions maker interested in increasing the number of priority projects that are supported, as well as the total number of projects, but with emphasis on those considered satisfactory (category 2).

One of the best compromise solutions obtained by Fernandez et al. in [12] is shown in Table 3. This solution contains a total of 24 projects, all belonging to the priority category. Compared to it, our red-marked solutions in Table 2 seem to be of greater impact and have equal or greater number of priority projects (24, 25, or 26), and contain much more total projects.

Our solutions would be preferred by every decision maker whose preferences are identified with the number of priority needs to attend to, and the total amount of needs (projects) addressed. Table 3 allows comparing the best solution by Fernández et al. ([12]) with our solution in the project space.

The impact indicator model is more flexible. The comparison shows that the proposal of [12] is a rigid model. This does not find several solutions that would have greater social benefit.

Table 4 shows the results of the instance that was used in [12]. As can be seen, the obtained non-dominated solutions seem to be satisfactory for the DM. The solutions obtained by our proposal should be more preferred than the best solution in [12] because this contains less projects and less priority projects.

6. Conclusions

The proposed model of impact indicators of the portfolio can explore the solution space and generate potential best portfolios, besides reasonably modelling decision maker preferences on portfolios under limited information about projects.

It was also proposed an evolutionary algorithm based on the NSGA-II that seems to be capable of obtaining solutions near to the Pareto frontier. The obtained solutions are more satisfactory than those obtained by the state of the art.

The quality of the solutions indicates that the algorithm converges close to the true Pareto frontier where best portfolios lie; this helps the decision maker to analyze his/her own preferences and to clarify his/her decisions.

We have obtained some evidence in favor to our proposal, which allows helps the DM in finding a rational compromise between the quality of the projects in the portfolio and the number of projects approved.

Table 1. Instance of 100 projects.

P	Budget	P	Budget	P	Budget	P	Budget
1	84.00	26	31.25	51	27.50	76	46.50
2	124.50	27	26.50	52	41.25	77	44.00
3	129.75	28	36.25	53	29.50	78	25.75
4	147.75	29	50.00	54	25.25	79	38.25
5	126.00	30	34.75	55	40.00	80	40.75
6	137.25	31	48.25	56	30.75	81	42.75
7	96.00	32	46.00	57	39.00	82	43.00
8	84.75	33	36.75	58	44.50	83	32.25
9	93.00	34	34.00	59	47.50	84	37.75
10	121.50	35	26.00	60	36.00	85	44.75
11	102.75	36	31.75	61	28.50	86	27.00
12	141.75	37	29.75	62	29.00	87	39.50
13	105.75	38	37.25	63	30.25	88	30.00
14	98.25	39	26.75	64	49.50	89	37.50
15	101.25	40	43.75	65	33.00	90	49.00
16	83.25	41	27.25	66	38.50	91	41.75
17	109.50	42	47.00	67	33.50	92	39.25
18	107.25	43	41.00	68	48.50	93	34.50
19	135.00	44	30.50	69	35.00	94	49.75
20	97.50	45	45.25	70	28.75	95	48.00
21	127.50	46	26.25	71	25.50	96	29.25
22	114.00	47	45.50	72	40.25	97	47.75
23	106.50	48	44.25	73	38.75	98	42.25
24	94.50	49	48.75	74	46.75	99	46.25
25	43.50	50	33.25	75	37.00	100	39.75
						Total	5542.00

*P: Rank ordering Identifier

Table 2. Experimental results obtained by ESPRI algorithm.

Objectives						Objectives					
I1	I2	I3	I4	I5	I6	I1	I2	I3	I4	I5	I6
21	1728	18	1022	2	41	20	1677	9	515	7	36
24	1995	15	859	1	40	16	1266	24	1235	10	50
12	974	30	1547	12	54	25	2102	8	464	2	35
13	1067	33	1683	5	51	5	414	24	1277	30	59
22	1817	18	1039	0	40	22	1852	17	957	1	40
19	1541	19	1091	2	40	11	861	22	1159	26	59
23	1856	16	917	2	41	19	1548	14	816	9	42
13	1046	27	1381	11	51	15	1182	13	746	23	51
22	1806	18	1032	1	41	26	2163	9	514	1	36
17	1387	12	624	15	44	14	1145	15	819	18	47
19	1589	8	458	12	39	24	1968	16	941	0	40
23	1869	16	933	1	40	24	1942	10	576	4	38
11	901	25	1242	20	56	16	1327	32	1648	1	49
17	1372	13	762	21	51	10	767	25	1339	25	60
9	708	26	1343	26	61	18	1487	10	493	17	45
14	1181	33	1683	4	51	24	1985	10	567	5	39
6	510	23	1225	30	59	26	2163	10	569	0	36
21	1732	17	980	3	41	22	1821	15	891	3	40
7	548	21	1089	34	62	12	939	21	1116	27	60
4	314	27	1415	27	58	12	993	30	1547	10	52
17	1402	24	1220	12	53	23	1876	18	1007	0	41
21	1743	18	1040	0	39	24	2000	12	689	2	38
19	1544	6	356	18	43	21	1720	18	978	5	44

Table 3. Solutions obtained from the work of Fernández et al. in [13] and of our proposal. These solutions consist of: the cardinality and final chromosome non-dominated solutions for each job.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Initial	22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[13]	24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[14]	40	1	1	0	0	1	0	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0

		51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
Initial	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[13]	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[14]	40	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4. Comparison of experimental results with the objectives evaluated with impact indicator model.

Papers	(I1, I2, I3, I4, I5, I6)
Fernández et al	24, 1995, 0, 0, 0, 24
Our proposal	24, 1995, 15, 859, 1, 40
Our proposal	25, 2051, 11, 644, 3, 39
Our proposal	25, 2079, 11, 609, 1, 37
Our proposal	26, 2151, 10, 577, 1, 37
Our proposal	26, 2163, 9, 514, 1, 36
Our proposal	25, 2102, 8, 464, 2, 35

References

[1] Cooper, R., Edgett, S., Kleinschmidt, E., (2001): "Portfolio Management for New Product Development: Results of an Industry Practices Study", *R&D Management* 31 (4), 361-380.

[2] Henriksen A.D., Traynor A.J. (1999): "A practical R&D project selection scoring tool", IEEE Transactions on Engineering Management 46 (2), 158-170.

[3] Macharis C., Brans, J.P., and Mareschal, B. (1998): "The GDSS PROMETHEE Procedure. A PROMETHEE-GAIA based procedure for group decision support", Journal of Decision Systems, 7, 283-307.

[4] Leyva, J.C., Fernandez, E., (2003). A new method for group decision support based on ELECTRE-III methodology. European Journal of Operational Research, 148 (1), 14-27.

[5] Fernández, E., López, E., Bernal, S., Coello Coello, C. A., and Navarro, J. Evolutionary multiobjective optimization using an outranking-based dominance generalization. Computers & Operations Research, 37(2):390–395.(2010a).

[6] Carazo, A. F., Gómez, T., Molina, J., Hernández-Díaz, A. G., Guerreo, F.

[7] M., and Caballero, R. Solving a comprehensive model for multiobjective project portfolio selection. Computers & Operations Research, 37(4):630–639.(2010).

[8] Castro, M. Development and implementation of a framework for I&D in public organizations. Master´s thesis, Universidad Autónoma de Nuevo León. (2007).

[9] García, R., Hyper-Heuristic for solving social portfolio problem. Master´sThesis, Instituto Tecnológico de Cd. Madero. (2010).

[10] Fernández, E. and Navarro, J.,A genetic search for exploiting a fuzzy preference model of portfolio problems with public projects. Annals OR, 117(191-213):191–213.(2002).

[11] S. Gabriel, Kumar, S., Ordoñez, J., Nasserian, A. (2006): "A multiobjective optimization model for project selection with probabilistic consideration", Socio-Economic Planning Sciences 40 (4), 297-313.

[12] Mavrotas, G., Diakoulaki, D., Koutentsis, A. (2008): "Selection among ranked projects under segmenetation, policy and logical constraints", European Journal of Operational Research 187 (1), 177-192, 2009.

[13] Fernández, E, Olmedo R. Public Project Portfolio Optimization Under A Participatory Paradigm. Applied Computational Intelligence and Soft Computing. Archive Volume 2013, January 2013. Article No. 4

[14] Deb, K., Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester-New York-Weinheim-Brisbane-Singapore-Toronto. (2001)

[15] Fernández E., Luz Flerida Félix, Gustavo Mazcorro., Multi-objective optimization of an outranking model for public resources allocation on competing projects. Int. J. Operational Research, Vol. 5, No. 2, pp. 190-210. (2009).

[16] Coello Coello, C. A., Lamont, G. B., and Van Veldhuizen, D. A. Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation. Springer, 2nd edition. (2007).

[17] Fernández Eduardo R., Navarro Jorge A., Olmedo Rafael A. Modelos y Herramientas Computacionales para el Análisis de Proyectos y la Formación de Carteras de I&D. Revista Iberoamericana de Sistemas, Cibernética e Informática. Volumen 1 - Número 1 – Año 2004, páginas: 59-64.

[18] Fernández, E., López, E., López, F., and Coello Coello, C. A. Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. Information Sciences, 181(1):44–56.(2010b).

[19] Ghasemzadeh, F., Archer, N., and Iyogun, P., A zero-one model for project portfolio selection and scheduling. Journal of the Operational Research Society, 50(7):745–755.(1999).

[20] Nebro Antonio J., Alba Enrique, Luna Francisco. Optimización Multi-Objetivo y Computación Grid. Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga. E.T.S. Ingeniería Informática. Campus de Teatinos, 2004.

[21] Peñuela, C. and Granada, M., Optimización multiobjetivo usando un algoritmo genético y un operador elitista basado en un ordenamiento no dominado (NSGA-II). Scientia Et Technica, 8(35):175–180. (2007).

[22] Roy, B. (1990). “The Outranking Approach and the Foundations of ELECTRE methods”, in Bana e Costa, C.A. (ed.), Reading in multiple criteria decision aid, Springer- Verlag, Berlin , 155-183.

[23] Roy, B. and Slowinski, R., Handling effects of reinforced preference and counter-veto in credibility of outranking. European Journal of Operational Research, 188(1):185–190.(2008).

[24] Tenorio Rodríguez Gilberto Javier. Optimización de carteras formadas por proyectos interdependientes en organizaciones públicas. Tesis para obtener el grado de maestro en ciencia en Ingeniería de sistemas, 2010.