

# Multi-objective Phylogenetic Algorithm: Solving Multi-objective Decomposable Deceptive Problems

Jean Paulo Martins, Antonio Helson Mineiro Soares,  
Danilo Vasconcellos Vargas, and Alexandre Cláudio Botazzo Delbem

Institute of Mathematics and Computer Science,  
University of São Paulo, São Carlos, SP, Brazil  
{jean,ahms,vargas,acbd}@icmc.usp.br

**Abstract.** In general, Multi-objective Evolutionary Algorithms do not guarantee find solutions in the *Pareto-optimal set*. We propose a new approach for solving decomposable deceptive multi-objective problems that can find all solutions of the *Pareto-optimal set*. Basically, the proposed approach starts by decomposing the problem into subproblems and, then, combining the found solutions. The resultant approach is a Multi-objective Estimation of Distribution Algorithm for solving relatively complex multi-objective decomposable problems, using a probabilistic model based on a phylogenetic tree. The results show that, for the tested problem, the algorithm can efficiently find all the solutions of the *Pareto-optimal set*, with better scaling than the hierarchical Bayesian Optimization Algorithm and other algorithms of the state of art.

## 1 Introduction

Techniques of search and optimization based in the Theory of Evolution as the Evolutionary Algorithms (EA) are distinguished in the solution of complex problems of optimization that possess some objective functions [3,7]. In such problems, in general, the objectives are conflicting, and there is not only one optimal solution that satisfies equally every objective, thus a set of solutions should be chose to attend the objectives of the problem.

Another important and relatively recent research area on EAs is focused on the developing of Estimation of Distribution Algorithms (EDAs) [17,18]. The main idea of the EDAs is the creation of a model that represents the relations between variables in a population (Building Blocks – BBs), through this model the disruption of BBs can be avoided, allowing the generation of better solutions and fast convergence.

In literature some EDAs are found to solve multi-objective problems as the meCGA [22] and the mohBOA [19], but these has in the construction of their models an impediment, resulting in a great number of functions evaluations. On the other hand, faster EDAs in general construct poorer models limiting their performance for large-scale multimodal problems. In fact, there is a trade-off between model quality and running time to construct the model itself.

Fortunately, several methods to reconstruct phylogenies (models describing the relationship among taxa as, for example, the species features) were developed in the last century. The phylogeny literature [9,21] shows that one of these methods, the Neighbor Joining (NJ) [20,23], is an adequate trade-off between quality and efficiency. This paper proposes a Multi-objective Estimation of Distribution Algorithm (MOEDA) based on phylogenetic models, using the NJ as a method to guarantee solutions in the *Pareto-optimal front* with reduced number of functions evaluations. This MOEDA is the multi-objective variation of the  $\Phi$ GA (Phylo-Genetic Algorithm) [15,25,26] called  $\text{mo}\Phi\text{GA}$ .

The remaining of the paper is organized as follows. Section 2 reviews fundamental concepts on Multi-objective Evolutionary Algorithms (MOEAs). Section 3 introduces the MOEDAs. Section 4 describes the Multi-objective Phylogenetic Algorithm ( $\text{mo}\Phi\text{GA}$ ). Section 5 shows tests and results within  $\text{mo}\Phi\text{GA}$  and Sect. 7 concludes the paper.

## 2 Multi-objective Evolutionary Algorithms

Multi-objective Evolutionary Algorithms is a well established field within Evolutionary Computation that deal with problems with multiple objectives (MOP - Multi-objective Optimization Problem). In such problems, the solutions are defined in relation to a set of objectives, so that each objective contributes to the solution's quality of the MOP.

Conflicting objectives are common in a MOP. In these cases, an increase for a particular objective is often limited by the decline in value of another objective. As such, in accordance with [5] there is basically two ways for solving multi-objective problems :

1. Preference-based methods;
2. Generating methods.

The *Preference-based methods* are normally formalized as the application of weights to the objective in some way, yielding a single-objective function. Each weight indicates the importance of the objective for the whole problem, thus, we have a composite function as (1), where  $x$  represents a solution to the whole problem and  $\alpha_i$  is the weight of every single-objective function  $f_i(x)$ .

$$g(x) = \sum_{i=1}^M \alpha_i f_i(x). \quad (1)$$

In that way, the multi-objective problem can be solved as a mono-objective problem. However, this approach requires knowledge of the weights for each objective, which is often not available or known. Moreover, those type of methods are not designed to find a family of solutions.

The *Generating methods* use a population of individuals to evolve multiple objectives at the same time. Using various strategies such as applying a changing selection criterion, choosing individuals based on their pareto optimality, among

others. These type of algorithms commonly aim at finding the *Pareto-optimal set*, since it allows the obtention of multiple optimal solutions concurrently without the need to balance goals.

The *Pareto-optimal front* from a given problem is defined as a set of all *Pareto-optimal* solutions. A solution is considered *Pareto-optimal* according to the concept of *dominance* [5], where a solution  $x_k$  dominates a solution  $x_l$  if:

1. The solution  $x_k$  is no worse (say the operator  $\prec$  denotes worse and  $\succ$  denotes better) than  $x_l$  in all objectives, or  $f_i(x_k) \not\prec f_i(x_l)$ , for  $i = 1, \dots, k$  objectives,
2. The solution  $x_k$  is strictly better than  $x_l$  in at least one objective, or  $f_{\bar{i}}(x_k) \succ f_{\bar{i}}(x_l)$ , for at least one  $\bar{i} \in \{1, \dots, k\}$ .

So, the Pareto-optimal set consists of non-dominated solutions from the solution space, while the Pareto-optimal front is the edge, at the objective space, composed by all non-dominated solutions.

Several MOEAs have been proposed based on the concept of dominance and applied to real problems (for example the Non-dominated Sorting (NSGA) [6] and the Strength Pareto EA (SPEA) [27]). However, some studies show problems regarding the scalability and time-convergence of these algorithms [5].

In Sect. 3 an alternative for the use of MOEAs is described, although there are also limitations on the scalability of this new approach [22][11], it presents significant advantages as the proximity of *Pareto-optimal front* and the number of evaluations.

### 3 Multi-objective Estimation of Distribution Algorithms

The Multi-objective Estimation of Distribution Algorithms differ from other MOEAs for not limiting themselves to raw information from a population. They use the selected individuals from a population as samples of an unknown probabilistic distribution and generate probabilistic models from these samples. The models should approximate the probabilistic distribution of the values of the variables or sub-sets of variables (strong correlation among variables, Building Blocks – BBs). In this way, a model is also a representation of the selected population itself (the used samples) and of other possible solutions that are related to the model but were not among the selected individuals [17,18,22].

The common reproduction operators are not used by MOEDAs since new individuals can be generated by directly sampling from the model. Moreover, if such model can identify the correct BBs of a problem, the combination of improbable values of variables can be avoided. An MOEDA that adequately estimates BBs can solve relatively large-scale complex problems.

However, as more representative is the model (more BBs it correctly estimates), more computationally complex is the algorithm to construct it. Thus, there is a trade-off between the efficiency of the algorithm for construction of a model and the accuracy of the model [17]. As a consequence, MOEDAs are

directly affected by the algorithm efficiency and the accuracy of the model they generate. Thus, an MOEDA with an adequate commitment between efficiency and accuracy would be a relevant contribution. Furthermore, several classes of important problems involving multimodal objective functions, large-scale instances, and solutions in real-time could be adequately solved using such MOEDA.

This paper proposes an MOEDA based on a method that has not been used in MOEDAs, the phylogenetic reconstruction [9]. This method enables an adjustable trade-off between the accuracy of the generated model (Phylogenetic Tree) and the computational efficiency to construct it.

## 4 Multi-objective Phylogenetic Algorithm

The Multi-objective Phylo-Genetic Algorithm (mo $\Phi$ GA) is an MOEDA based on the  $\Phi$ GA, extending its features to find the *Pareto-optimal* set, i.e., solving multi-objective problems. It uses a phylogenetic tree reconstruction (NJ) method to generate a phylogenetic tree for each problem separately. And from the analysis of each tree through the uncertainty variation [15,26], the mo $\Phi$ GA can exactly determine the BBs of a separable deceptive multi-objective problem. This algorithm is synthesized in the diagram from Fig.1, where every branch is associated to one objective ( $Obj_1, Obj_2, \dots, Obj_M$ ).

The mo $\Phi$ GA starts with a population of size  $P$  with random individuals and applies  $\Theta$  tournament selections defined by (2).

$$\Theta = \Phi P \quad (2)$$

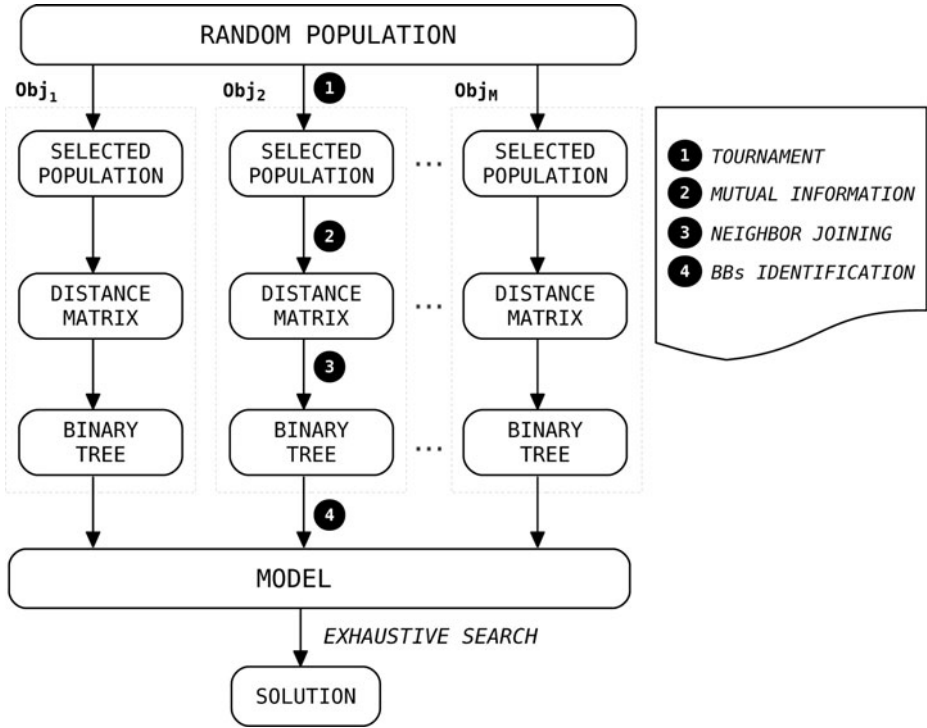
Where  $\Phi$  is a variable already used in previous  $\Phi$ GAs. And it can adjust a trade-off between number of avaliations and the running time of the algorithm [15,25,26].

For a selected population (of a respective objective), a method calculates a distance matrix (Sect. 4.2), required by the NJ. The mo $\Phi$ GA uses Mutual Information (Sect. 4.1) to calculate the distance matrix. Then, NJ is applied to create a phylogenetic tree for each objective separately. And the same procedure is applied to all objectives.

Then the BBs are identified based on an uncertainty criteria of the NJ method. That is, considering the difference between the average and minimum distance in the matrix  $M$  (see Section 4.2 for details of the matrix  $M$ ), if we call the average minus the minimum distance in the  $M$  in a specific  $t$  iteration  $\alpha_t$ , the relative rate it varies can be approximate by the following function fraction:

$$\frac{\alpha_t - \alpha_{t-1}}{\alpha_t}. \quad (3)$$

Disconsidering the points with  $\alpha_t$  near 0, the highest point  $t^*$  of this fraction represent the stopping criteria for the NJ. Therefore, every internal node created by the NJ after the point  $t^*$  should be discarded and the remaining subtrees will form the BBs [26],[15].



**Fig. 1.** Basic steps of the mo $\Phi$ GA for separable deceptive problems

Sequentially, each objective is combined as clusters (excluding equal clusters of variables), leading to a small set of BBs. Finally, an exhaustive search is applied at each BB from the set of BBs to find the *Pareto-optimal* for each BB. The combination of the *Pareto-optimal* solutions at each obtained BB in the tests was able to compose not just some or the solutions with different values of fitness, but all the possible combinations of solutions in the *Pareto-optimal front*.

This type of result was possible, because mo $\Phi$ GA focuses on solving the linkage learning (the interdependence between the variables). This contrasts with the results of other usual multi-objective Population-based Algorithms, which are able to find most but not all solutions. This may also shed light to a different focus on tackling these types of problems, which is to focus on decomposing them and not on searching solely for the solutions. The solution of the decomposition, as showed by this article, may make the searching process easier.

The complete description of the algorithm are explained in the sections as follows, Sect. 4.1 describe how the distance matrix is built, necessary for the NJ (reconstruction of phylogenetic trees method). Which is presented in Sect. 4.2.

### 4.1 Distance Metric

In order to construct the distance matrix  $D$ , a metric capable of extracting the correlation between variables is desirable. Mutual Information [14] is a measure of similarity between two variables which has been successfully applied to the development of EDAs. Equation (4) describe it, where  $X$  and  $Y$  are two random variables,  $p_{xy}(X, Y)$  is the joint probability of  $(X = x, Y = y)$ ; and  $p_x(X)$  and  $p_y(Y)$  are the marginal probability of  $X = x$  and  $Y = y$ , respectively.

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p_{xy}(X, Y) \log \frac{p_{xy}(X, Y)}{p_x(X)p_y(Y)}. \quad (4)$$

However, the Mutual Information is not a metric, since it does not satisfy strictly the triangle inequality. Thus (5), as described in [14] has been widely used in the literature [1,2,8,13] specially because it satisfies the triangle inequality, non-negativity, indiscernibility and symmetric properties. In (5),  $H(X, Y)$  is the entropy of the pair  $(X, Y)$ .

$$D(X, Y) = H(X, Y) - I(X, Y). \quad (5)$$

The metric (5) is also an universal metric, in the sense that if any non-trivial distance measure places  $X$  and  $Y$  close, then it will also be judged close according to  $D(X, Y)$  [14].

Once the phylogenetic tree is constructed, the subtrees that represents the correlated variables (these subtrees are called clados in Biology), and consequently the Building Blocks, need to be identified, and this identification is an empirical procedure that depends on *ad hoc* knowledge about the problem. This problem was solved for the use of the NJ by the  $\Phi$ GA and explained at [26,15,25].

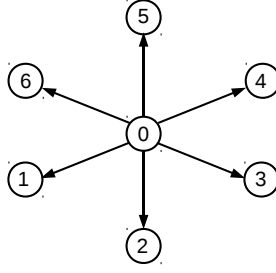
### 4.2 Neighbor Joining

In this context, the Neighbor Joining can be viewed as a hierarchical clustering method [4,12,16] that uses a distance matrix  $D$  to create a phylogenetic tree [20,23]. The distance matrix is used as information about the relationship between variables and the tree is built in such way that more related variables must be closer in the tree.

The algorithm starts with all variables connected to an unique internal node, labeled "0", composing a star tree of  $n + 1$  nodes, where  $n$  is the number of variables, Fig. 2 illustrates a star tree.

For every leaf node  $i \in N = \{1, \dots, n\}$ , the net divergence,  $R_i$ , is calculated by (6), which is the sum of all distances referent to node  $i$ . From this, a new matrix  $M$  is calculated using  $D$  and net divergences (7), where  $D_{ij}$  is the element at position  $(i, j)$  in  $D$ .

$$R_i = \sum_{j \in N, j \neq i} D_{ij}. \quad (6)$$



**Fig. 2.** Star tree of a problem with 6 variables

$$M_{ij} = D_{ij} - \frac{R_i + R_j}{n - 2}. \quad (7)$$

$M$  is very important for the NJ success since the joining nodes will derive from it. The two nodes,  $i$  and  $j$ , corresponding to the smallest  $M_{ij}$ , are removed from the star tree. Then, we insert a new node  $u$ , connected to  $i$  and  $j$ , beginning the formation of a binary tree.

To calculate the branch length from  $u$  to  $i$  and  $j$ , (8) and (9) are used, where  $S_{ij}$  is the branch length between  $i$  and  $j$ .

$$S_{iu} = \frac{D_{ij}}{2} + \frac{R_i - R_j}{2(n - 2)}, \quad (8)$$

$$S_{ju} = \frac{D_{ij}}{2} + \frac{R_j - R_i}{2(n - 2)}. \quad (9)$$

Afterward, it is calculated the distance from the other variables to  $u$  in order to fulfill a new  $D$  (without columns related to  $i$  and  $j$  and with a new column for  $u$ ). Such calculi is defined by (10), where  $k$  is necessarily a node different from  $u$  in  $D$ .

$$D_{ku} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}, \forall k \neq u. \quad (10)$$

$D$  has decreased its size from  $n \times n$  to  $(n - 1) \times (n - 1)$ . Then, this new  $D$  is used to calculate a new  $M$ . The whole process is repeated until the size of  $D$  reaches 2. Then, the two remaining nodes are joined together creating the last internal node, which is set as the root of the tree. Lastly, the node “0” (that primarily connected every node in a star tree) is removed from the tree.

## 5 Test Problems and Results

The mo $\Phi$ GA was tested with a relatively complex multi-objective function from the literature [22], considering all the test factors: problem size, population size and number of executions. This multi-objective function is a combination of two

objectives and is defined by (11) and (12), both being linearly separable fully deceptive problems with conflicting objectives, where  $x$  represents a solution to the problem,  $m$  is the number of BBs and  $k = 5$  is the trap size.

The *trap5* and the *inv-trap5* are difficult problems, because standard crossing over operators can not solve it, unless the bits of each partition are placed close to each other. Mutation operators are also very inefficient for solving them, requiring  $O(n^5 \log n)$  evaluations [24].

$$\max f_{trap5}(x), \quad x \in \{0, 1\}^{mk}, \quad (11)$$

$$f_{trap5}(x) = \sum_{i=0}^{m-1} trap5(x_{ki} + x_{ki+1} + \dots + x_{ki+k-1}),$$

$$trap5(u) = \begin{cases} 5 & \text{if } u = 5, \\ 4 - u & \text{if } u < 5. \end{cases}$$

$$\max f_{inv-trap5}(x), \quad x \in \{0, 1\}^{mk}, \quad (12)$$

$$f_{inv-trap5}(x) = \sum_{i=0}^{m-1} inv-trap5(x_{ki} + x_{ki+1} + \dots + x_{ki+k-1}),$$

$$inv-trap5(u) = \begin{cases} 5 & \text{if } u = 0, \\ u - 1 & \text{if } u > 0. \end{cases}$$

Considering a binary representation for the solutions, (11) has an optimum global solution consisting uniquely of ones, whereas (12) has an optimum global solution consisting uniquely of zeros. The Fig. 3 represents the relation between them in a string of 5 bits.

The  $\text{mo}\Phi\text{GA}$  was used to find solutions in the *Pareto-optimal front*. To confirm the solutions quality, the entire solution space was identified, enabling the confirmation that the solutions found lies in the *Pareto-optimal front*.

Some parameters used for the experiments are described in Table 1, also are used a tournament of size 16, and 30 runs for each experiment.

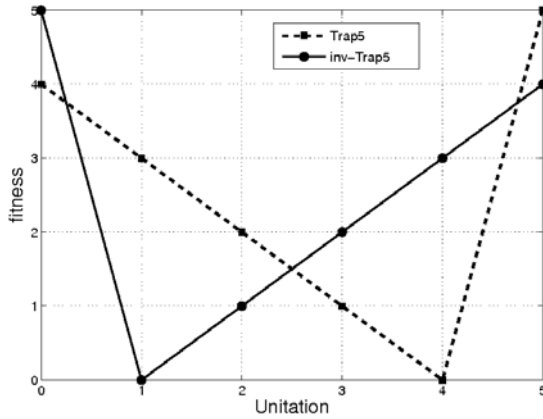
Figures 4 and Fig. 5 presents the space of solutions and the *Pareto-optimal set* found respectively for problems of size 30 and 50.

Those tests evidence the robustness of the algorithm, because it was able to find the entire *Pareto-optimal set* for all the tests presented. Even with the

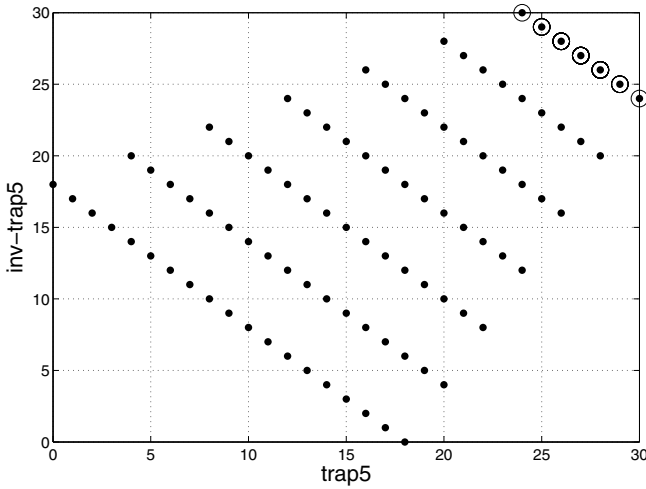
**Table 1.** Population sizes obtained by the Bisection Method [10]

<b>Problem size</b>	30	50	100
<b>Population size</b>	3843	7704	22543





**Fig. 3.** Functions  $trap_5$  and  $inv - trap_5$

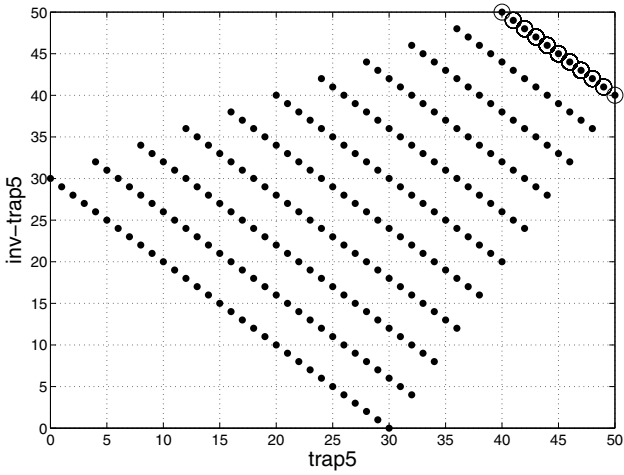


**Fig. 4.** Problem of size 30. The circled points are both the solutions found and the complete *Pareto-optimal front*.

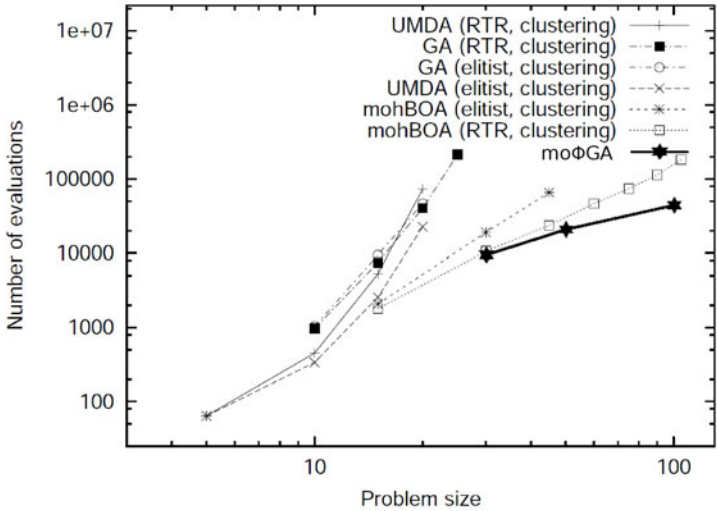
amount of possible solutions growing exponentially with the size of the problem [22].

The  $trap_5$  vs.  $inv-trap_5$  problem was also tested by [19] and our experiments were done using the same parameters. The results obtained by them and the results found by us (applying  $mo\Phi GA$  to problems of size 30, 50 and 100) are represented in the Fig. 6.

By observing Fig. 6 it is possible to perceive that the number of evaluations used by  $mo\Phi GA$  has a lower slope than the algorithms described in [19], where



**Fig. 5.** Problem of size 50. The circled points are both the solutions found and the complete *Pareto-optimal front*.



**Fig. 6.** A comparison of  $\text{trap5}$  vs.  $\text{inv-trap5}$  problem. This image was first published in [19] and edited to include the moΦGA results.

multi-objective hBOA had better results than the others algorithms tested, this can be an indication of the minor complexity of the mo $\Phi$ GA in relation to the other algorithms from the literature.

## 6 Future Work

This article sheds light on an approach of how to solve efficiently relative complex decomposable deceptive problems. The trap-invtrap problem is an important one, because it points out deficiencies of many discrete multi-objective EAs, however, it is not enough, lots of another experiments need to be done before the approach can be completely validated.

Thus, there are important points to be extended in future versions of mo $\Phi$ GA to enable the algorithm to solve many-objective problems, multi-objective problems with different Building Block sizes, problems in continuous search spaces, hierarchical problems and so on.

## 7 Conclusions

A great variety of problems exists in the real world, which naturally demand efficient solutions of multi-objective problems. For such problems the MOEDAs were proven to achieve the most prominent solutions. This paper proposes a MOEDA, the mo $\Phi$ GA, which can adjust the trade-off between running time and functions evaluations, by the variation of parameter  $\Phi$ .

Moreover, in all the problems tested, the mo $\Phi$ GA surpassed the mohBOA in terms of less function evaluations. Specially when the problems in question increased in size. In fact, the mo $\Phi$ GA finds the entire set of solutions that compose the *Pareto-optimal front*, not only some solutions as most of the algorithms from the literature. This is made possible by approaching problems in two clear stages:

1. Decomposition Stage
2. Optimization Stage

The importance of the approach is demonstrated specially well in the proposed algorithm, because even with a simple optimization stage, the mo $\Phi$ GA achieved state of art results. The same approach was validated in other difficult problems by other  $\Phi$ GA algorithms [15,26].

The promising results presented by the mo $\Phi$ GA motivates new variants of it, as for example, an algorithm with a more robust optimization stage, as well as, an extension to deal with continuous global optimization problems and hierarchical deceptive problems.

## References

1. Aghagolzadeh, M., Soltanian-Zadeh, H., Araabi, B., Aghagolzadeh, A.: A hierarchical clustering based on mutual information maximization. In: IEEE International Conference on Image Processing, ICIP 2007, vol. 1 (2007)

2. Apornthewan, C., Ballard, D., Lee, J.Y., Lee, J.S., Wu, Z., Zhao, H.: Gene hunting of the Genetic Analysis Workshop 16 rheumatoid arthritis data using rough set theory. In: BMC Proceedings, vol. 3, p. S126. BioMed Central Ltd (2009)
3. Coello, C.A.C., Zacetenco, S.P., Pulido, G.T.: Multiobjective optimization using a micro-genetic algorithm (2001)
4. Day, W., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification* 1(1), 7–24 (1984)
5. Deb, K.: Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation* 7(3), 205–230 (1999)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
7. Deb, K.: Multi-objective optimization using evolutionary algorithms (2001)
8. Dionísio, A., Menezes, R., Mendes, D.A.: Entropy-based independence test. *Non-linear Dynamics* 44(1), 351–357 (2006)
9. Felsenstein, J.: *Inferring Phylogenies*, vol. 266. Sinauer Associates (2003)
10. Harik, G.: Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. Ph.D. thesis, The University of Michigan (1997)
11. Hughes, E.: Evolutionary many-objective optimisation: many once or one many? In: The 2005 IEEE Congress on Evolutionary Computation, vol. 1, pp. 222–227. IEEE, Los Alamitos (2005)
12. Johnson, S.: Hierarchical clustering schemes. *Psychometrika* 32(3), 241–254 (1967)
13. Kraskov, A.: Synchronization and Interdependence Measures and Their Application to the Electroencephalogram of Epilepsy Patients and Clustering of Data. Report Nr. NIC series 24 (2008)
14. Kraskov, A., Stogbauer, H., Andrzejak, R., Grassberger, P.: Hierarchical clustering based on mutual information. Arxiv preprint q-bio/0311039 (2003)
15. de Melo, V.V., Vargas, D.V., Delbem, A.C.B.: Uso de otimização contínua na resolução de problemas binários: um estudo com evolução diferencial e algoritmo filo-genético em problemas deceptivos aditivos.. In: 2ª Escola Luso-Brasileira de Computação Evolutiva (ELBCE), APDIO (2010)
16. Morzy, T., Wojciechowski, M., Zakrzewicz, M.: Pattern-oriented hierarchical clustering. In: Eder, J., Rozman, I., Welzer, T. (eds.) ADBIS 1999. LNCS, vol. 1691, pp. 179–190. Springer, Heidelberg (1999)
17. Pelikan, M., Goldberg, D., Lobo, F.: A survey of optimization by building and using probabilistic models. *Computational optimization and applications* 21(1), 5–20 (2002)
18. Pelikan, M., Sastry, K., Cantu-Paz, E.: Scalable optimization via probabilistic modeling: From algorithms to applications. Springer, Heidelberg (2006)
19. Pelikan, M., Sastry, K., Goldberg, D.: Multiobjective hBOA, clustering, and scalability. In: Genetic And Evolutionary Computation Conference: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 663–670. Association for Computing Machinery, Inc., New York (2005)
20. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution* 4(4), 406 (1987)
21. Salemi, M., Vandamme, A.M.: *The Phylogenetic Handbook: A Practical Approach to DNA and Protein Phylogeny*, vol. 16. Cambridge University Press, Cambridge (2003), <http://doi.wiley.com/10.1002/ajhb.20017>
22. Sastry, K., Goldberg, D., Pelikan, M.: Limits of scalability of multiobjective estimation of distribution algorithms. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 3, pp. 2217–2224. IEEE, Los Alamitos (2005)

23. Studier, J., Keppler, K.: A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution* 5(6), 729 (1988)
24. Thierens, D.: Analysis and design of genetic algorithms. Katholieke Universiteit Leuven, Leuven (1995)
25. Vargas, D.V., Delbem, A.C.B.: Algoritmo filogenético. Tech. rep., Universidade de São Paulo (2009)
26. Vargas, D.V., Delbem, A.C.B., de Melo, V.V.: Algoritmo filo-genético. In: 2<sup>a</sup> Escola Luso-Brasileira de Computação Evolutiva (ELBCE), APDIO (2010)
27. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4), 257–271 (2002)