

Extending the GA-EDA Hybrid Algorithm to Study Diversification and Intensification in GAs and EDAs

V. Robles¹, J.M. Peña¹, M.S. Pérez¹, P. Herrero², and Ó. Cubo¹

¹ Departamento de Arquitectura y Tecnología de Sistemas Informáticos,
Universidad Politécnica de Madrid, Madrid, Spain
{vrobles, jmpena, mperez, ocubo}@fi.upm.es

² Departamento de Lenguajes y Sistemas, Universidad Politécnica de Madrid, Madrid, Spain
pherrero@fi.upm.es

Abstract. Hybrid metaheuristics have received considerable interest in recent years. Since several years ago, a wide variety of hybrid approaches have been proposed in the literature including the new GA-EDA approach. We have design and implemented an extension to this GA-EDA approach, based on statistical significance tests. This approach had allowed us to make an study of the balance of diversification (exploration) and intensification (exploitation) in Genetic Algorithms and Estimation of Distribution Algorithms.

1 Introduction

Over the last years, interest in hybrid metaheuristics has risen considerably among researchers. The best results found for many practical or academic optimization problems are obtained by hybrid algorithms. Combination of algorithms such as descent local search [15], simulated annealing [10], tabu search [6] and evolutionary algorithms have provided very powerful search algorithms.

Two competing goals govern the design of a metaheuristic [19]: exploration and exploitation. Exploration is needed to ensure every part of the search space is searched thoroughly in order to provide a reliable estimate of the global optimum. Exploitation is important since the refinement of the current solution will often produce a better solution. Population-based heuristics (where genetic algorithms [9] and estimation of distribution algorithms [12] are found) are powerful in the exploration of the search space, and weak in the exploitation of the solutions found.

With the development of our new approach, GA-EDA, a hybrid algorithm based on genetic algorithms (GAs) and estimation of distribution algorithms (EDAs), we aim to improve the explorations power of both techniques.

This hybrid algorithm has been tested on combinatorial optimization problems (with *discrete* variables) as well as *real-valued* variable problems. Results of several experiments show that the combination of these algorithms is extremely promising and competitive.

This paper is organized in the following way: First, we will focus on different taxonomies of hybrid algorithms found in the literature; in section 3, the GA-EDA approach is reviewed with a complete performance study presented in section 4. Finally we close with our conclusions and further future work.

2 Taxonomy of Hybrid Algorithms

General taxonomies provides a mechanism to allow comparison of hybrid algorithms in a qualitative way and classifying new hybrid approaches. This section highlights some of the most important hybrid taxonomies.

[2] describes three different forms of hybridization:

- *Component Exchange Among Metaheuristics.*

One of the most popular hybridization methods is the use of trajectory methods such as Local Search, Tabu Search, in population-based algorithms. These solutions combine the advantages of population based methods, which are better on diversification, and trajectory methods, which are better on intensification. For example [7] incorporates local search in a genetic framework.

- *Cooperative Search* [1,4,21].

The second hybridization approach consists of a search performed with various algorithms that, typically, execute in parallel and exchange information about states, solutions, sub-problems or other characteristics.

- *Integrating Metaheuristics and Systematic Methods* .

This approach has produced very effective algorithms. For instance [5] integrates metaheuristics and Constraint Programming.

A complementary taxonomy can be found in [19] which defines a hierarchical classification.

- *LRH (Low-level Relay Hybrid).*

A given metaheuristic is embedded into a single-solution metaheuristic. For instance in [14] a LRH hybrid combines simulated annealing with local search.

- *LCH (Low-level Co-evolutionary Hybrid).*

Algorithms consist in population based heuristics coupled with local search heuristics. The population based algorithms will try to optimize globally and the local search will try to optimize locally.

- *HRH (High-level Relay Hybrid).*

The metaheuristics are executed in a sequence, one after another, each using the output of the previous as its input. In [13] annealing is used to improve the population obtained by a GA.

- *HCH (High-level Co-evolutionary Hybrid).*

Several algorithm perform a search in parallel and cooperate in order to find the optimum. This approach is similar to the previous *cooperative search*. The use of parallel EDAs in a island model [18] is an of this.

The hybrid algorithm GA-EDA, can be classified as *cooperative search* in Blum and Roli's taxonomy. In Talbi's classification GA-EDA is heterogeneous; global because the algorithm search the whole state space, and general because both algorithms solve the same problem.

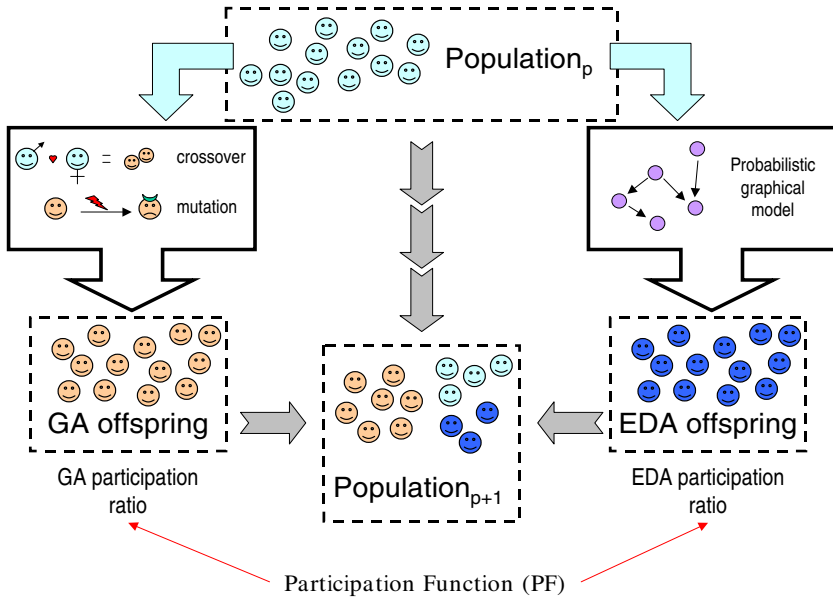


Fig. 1. Hybrid Evolutionary Algorithm Schema

3 Hybrid GA-EDA Algorithm

Hybrid GA-EDA are new algorithms based on both techniques [16,17]. The original objective is to get benefits from both approaches. The main difference from these two evolutionary strategies is how new individuals are generated. These new individuals generated on each generation are called *offspring*. Our new approach generates two groups of offspring individuals, one generated by the GA mechanism and the other by EDA one. On one hand, GAs use crossover and mutation operators as a mechanism to create new individuals from the best individuals of the previous generation. On the other hand, EDAs build a probabilistic model with the best individuals and then sample the model to generate new ones.

$Population_{p+1}$ is composed by the best overall individuals from (i) the past population ($Population_p$), (ii) the GA-evolved offspring, and (iii) EDA-evolved offspring.

The individuals are selected based on their fitness function. This evolutionary schema is quite similar to Steady State GA in which individuals from one population, with better fitness than new individual from the offspring, survive in the next one. In this case we have two offspring pools. Figure 1 shows how this model works.

3.1 Participation Functions

On this approach an additional parameter appears, this parameter has been called *Participation Function* (PF). PF provides a ratio of how many individuals are generated by

each mechanism. In other words, the size of GA and EDA offspring sets. The size of these sets also represents how each of these mechanisms participates on the evolution of the population. These ratios are only a proportion for the number of new individuals each method generates, it is not a proportion of individuals in the next population, which is defined by the quality of each particular individual. If a method were better than the other in terms of how it combines the individuals there would be more individuals from this offspring set than the other.

Several alternatives to these Participation Functions were taken into account in previous experiments, being some of them: the Constant Ratio ($x\%$ EDA / $y\%$ GA), the Alternative Ratio (*ALT*), the Incremental Ratio (EDA++ and GA++) or the Dynamic Ratio (*DYNAMIC*). More information about them could be found in [16,17] From all of these alternatives, maybe could be useful to highlight the last one (*DYNAMIC*), which has a mechanism that increases the participation ratio for the method that happens to generate best individuals. This function evaluates each generation considering the possibility to change the participation criterion as defined by the ratio array.

The *DYNAMIC* algorithm starts with 50%/50% ratio distribution between the two methods. On each generation the best offspring individuals from each method are compared and the winning method gets a 5% of the ratio of the opposite method (scaled by the amount of relative difference between the methods, *diff* variable). This mechanism provides a contest-based dynamic function in which methods are competing to get higher ratios as they generate better individuals.

4 The New Range Based Participation Function

In this section we present a new participation function that is based on the first steps of the Mann-Whitney non-parametric test. In this test there is no hypothesis that the initial samples should follow a normal distribution, which is important in this environment.

The new Range Based Participation Function begins by assembling the fitness from GA and EDA populations into a single set of size $N = n_{GA} + n_{EDA}$. These measures are then rank-ordered from lowest (*rank1*) to highest (*rankN*), with tied ranks included where appropriate.

Once they have been sorted out in this fashion, the rankings are then returned to the population, GA or EDA, to which they belong and substituted for the fitness measures that gave rise to them.

The effect of replacing raw measures with ranks is two-fold. The first is that it brings us to focus only on the ordinal relationships among the raw measures (“greater than”, “less than” and “equal to”) with no illusion or pretense that these raw measures derive from an equal-interval scale. The second is that it transforms the data array into a kind of closed system, many of whose properties can then be known by dint of sheer logic.

Let be,

$T_{GA} = \text{the_sum_of_the_}n_{GA}\text{-ranks_in_group_GA}$

$T_{EDA} = \text{the_sum_of_the_}n_{EDA}\text{-ranks_in_group_EDA}$

Now, we would like to know if GA and EDA do not differ with respect to their effectiveness. If this were true, then the raw measures within fitness in GA and EDA

would be about the same, on balance, and the rankings that derive from them would be evenly mixed within fitness in GA and EDA, like cards in a well shuffled deck.

So if this were true, we would expect the separate averages of the GA ranks and the EDA ranks each to approximate the same overall mean value. This entails that the rank-sums of the two groups, T_{GA} and T_{EDA} , would approximate the values,

$$Mean_{GA} = n_{GA}(N + 1)/2$$

$$Mean_{EDA} = n_{EDA}(N + 1)/2$$

Thus we know that:

- The observed value of T_{GA} belongs to a sampling distribution whose mean is equal to $Mean_{GA}$.
- The observed value of T_{EDA} belongs to a sampling distribution whose mean is equal to $Mean_{EDA}$.

Finishing, the effectiveness of GA and EDA will be,

$$Effect_{GA} = T_{GA}/Mean_{GA}$$

$$Effect_{EDA} = T_{EDA}/Mean_{EDA}$$

Thus, the percentages for the next generation will be,

$$Perc_{GA} = Effect_{GA}/Effect_{GA} + Effect_{EDA}$$

$$Perc_{EDA} = Effect_{EDA}/Effect_{GA} + Effect_{EDA}$$

5 Behavior Analysis of DYNAMIC vs. RANGE Participation Functions

The experiments to compare the behavior of DYNAMIC and RANGE Participation Functions have been performed considering five continuous problems:

- ① *Branin* RCOS function
- ② *Griewank* function
- ③ *Rastrigin* function
- ④ *Schweifel's* problem [8,20]
- ⑤ A continuous version of the *MaxBit* problem

The hybrid algorithm is composed of the simplest versions of both GA and EDA components. In this sense a real string (real-coded vector) has been used to code all the problems. GA uses *Roulette Wheel* selector, one-point crossover, flip mutation (in this case selecting a random gene, with probability 0.01 and generating a new value using an uniform random distribution) and uniform initializer. EDA uses the continuous version of the Univariate Marginal Distribution Algorithm (UMDA_c) [11]. The overall algorithms generate an offspring twice the size of the population. Depending on the ratios provided by the Participation Function, this offspring is then distributed between the two methods. The composition of the new population is defined by a deterministic method, selecting the best fitness scores from the previous population and both offspring sets. The stopping criteria is quite straightforward, we stop when the difference of the

Table 1. Branin

	GA	EDA	DYNAMIC	RANGE
Average fitness	0.4016	0.3987	0.4000	0.3990
Average generation number	19	19	19	19

sum of the fitness values of all individuals in two successive generations is smaller than a predefined value.

After having executed ten consecutive times the experiments, the average of the best fitness values and the average of the number of generations are calculated. Several population sizes have been tested, but in this paper we only present the most representative size. All these experiments have been performed in an 8-nodes cluster of bi-processors with Intel Xeon 2.4GHz with 1GB of RAM and Gigabit network running Linux 2.4.

With the aim of making a good comparison among the results achieved by all the presented algorithms, we have done the Mann-Whitney statistical test to compare them. The fitness values of the best solutions found in the search are used for this purpose.

It is important to highlight that the results presented in this paper depend on the individual representation used for each of the problems.

5.1 Branin RCOS Function

Definition. This problem is a two-variable continuous problem with three global minimum and no local minimum. The problem is defined as follows [3]:

$$f_B(x_1, x_2) = \left(x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10$$

$$-5 < x_1 < 10$$

$$0 < x_2 < 15$$

The global optimum for this problem is 0.397887 with the following values $(x_1, x_2) = (-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475)$.

This problem is considered easy not only because of the number of variables, but the small chance to miss the basin of the global minimum in a global optimization procedure. This is due to the probability to reach the global optimum using local optimization methods, started with a small number of random points is quite high.

Results. Branin is a very simple problem where in few generations (approx 19) all the algorithms converge. This problem was solved using a population size of 300 individuals.

As it is possible to appreciate in the table 2, EDA gets better results than GA. However, the hybrid algorithm with the RANGE Participation Function obtains significant better results than GA, EDA and the DYNAMIC Participation Function.

Table 2. Statistical Significance Tests for Branin

Mann-Whitney Test	p-value
EDA better GA	0.1126
RANGE better DYNAMIC	0.2315
RANGE better EDA	0.2697
RANGE better GA	0.0790

5.2 Rastrigin Function

Definition. It is a scalable, continuous, and multimodal function that must be minimized. It's the result of modulating n -dimensional sphere function with $a \cdot \cos(\omega x_i)$.

$$f_{Ra5}(\mathbf{x}) = a \cdot n + \sum_{i=1}^n (x_i^2 - a \cdot \cos(\omega \cdot x_i))$$

$$a = 10; \omega = 2\pi; n = 5$$

$$-5.12 < x_i < 5.12$$

The global minimum for this problem can be found in the solution $x_i = 0, i = 1, \dots, n$ with a fitness value of 0.

Results. This problem was solved using a population size of 1000 individuals.

Table 3. Rastrigin

	GA	EDA	DYNAMIC	RANGE
Average fitness	0.11656	4.10471	0.00013	0.00005
Average generation number	27	28	28	28

Although Rastrigin function has no lineal dependency among the variables, the performance of EDAs (with the UMDA approach) is very poor. Nearby the optimum value there are many local optima and EDAs seems to be very sensitive to this characteristic.

Table 4. Statistical Significance Tests for Rastrigin

Mann-Whitney Test	p-value
EDA better GA	0.1126
RANGE better DYNAMIC	0.1601
RANGE better EDA	0
RANGE better GA	0

The table 4 presents the Mann Whitney significance tests for this problem. In this case, EDA is better than GA with a p-value of 0.1126 and the RANGE Participation Function is significantly better than the DYNAMIC Participation Function with a p-value of 0.1601. Moreover, RANGE is better than GAs and EDAs with p-values equal to 0.

Table 5. Schwefel

	GA	EDA	DYNAMIC	RANGE
Average fitness	3.129	1852.836	0.120	0.037
Average generation number	31	26	34	33

5.3 Schwefel's Problem

Definition. Schwefel's function is a continuous multimodal function. It is interesting because it is a separable problem, it means that searching along the coordinate axes gives optimal values for each of the components because function gradient is oriented along the axes. As in the previous case global optimum is surrounded by several local optimum in the neighborhood.

$$f_{S10}(\mathbf{x}) = \sum_{i=1}^n x_i \cdot \sin(\sqrt{|x_i|})$$

$$n = 10$$

$$-500 < x_i < 500$$

$$f_{S10}(\mathbf{x}^*) = \min(f_{S10}(\mathbf{x}))$$

The global minimum for this problem can be found in the solution $x_i = 420.9687$, $i = 1, \dots, n$ with a fitness value of 0.

Results. This problem has been solved with a population of 2000 individuals.

GAs perform very good in this problem because of the separability of the component optimal values. Genetic combination tries to preserve good gene values when generating new individuals. Although (see Table 6) GA is much better than EDA and, one more time, RANGE outperforms DYNAMIC, GAs and EDAs.

Table 6. Statistical Significance Tests for Schwefel

Mann-Whitney Test	p-value
EDA better GA	1
GA better EDA	0
RANGE better DYNAMIC	0.0001
RANGE better EDA	0
RANGE better GA	0

5.4 The MaxBit Continuous Problem

Definition. This problem is a redefinition of the binary MaxBit problem previously presented. The aim is to maximize:

$$f_{M12}(\mathbf{x}) = \frac{\sum_{i=1}^n x_i}{n}$$

$$x_i \in \{0, 1\}; n = 12$$

In the continuous domain this problem is more complex, as the optimum value of the function is located on the boundary of the search space.

Results. This problem has been solved with a population of 250 individuals.

In this experiment (see Table 6), GA is much better than EDA and, one more time, RANGE outperforms DYNAMIC, GAs and EDAs.

In the MaxBit Continuous problem EDA is slightly better than GA (with p-value equal to 0). However, DYNAMIC and RANGE have the same behavior getting the maximum value for all the problem executions.

6 Intensification and Diversification in GAs and EDAs

One interesting issue is to survey the evolution of the DYNAMIC and RANGE Participation Functions in the series of different experiments. These functions, as we have seen, adjust the participation ratio depending on the quality of the individuals each of the method is providing. This measure has been indirectly used to evaluate the quality of each of the methods across the continuous generations of one algorithm.

In Figure 2 the evolution of the two different participation functions is shown. Being the first one associated to the DYNAMIC participation function and the second one to the RANGE participation function. Moreover we have introduced an additional section at the bottom of the figure with the aim of clarifying the progress of diversification and intensification in the optimization process.

DYNAMIC participation function (Figure 2.a) behaves with smooth variation in the ratios for each of the evolutionary methods. As diversification features are required in

Table 7. MaxBitCont

	GA	EDA	DYNAMIC	RANGE
Average fitness	0.9940	0.9998	1	1
Average generation number	36	40	33	34

Table 8. Statistical Significance Tests for MaxBitCont

Mann-Whitney Test	p-value
EDA better GA	0
RANGE better DYNAMIC	1
DYNAMIC better RANGE	1
RANGE better EDA	0
RANGE better GA	0

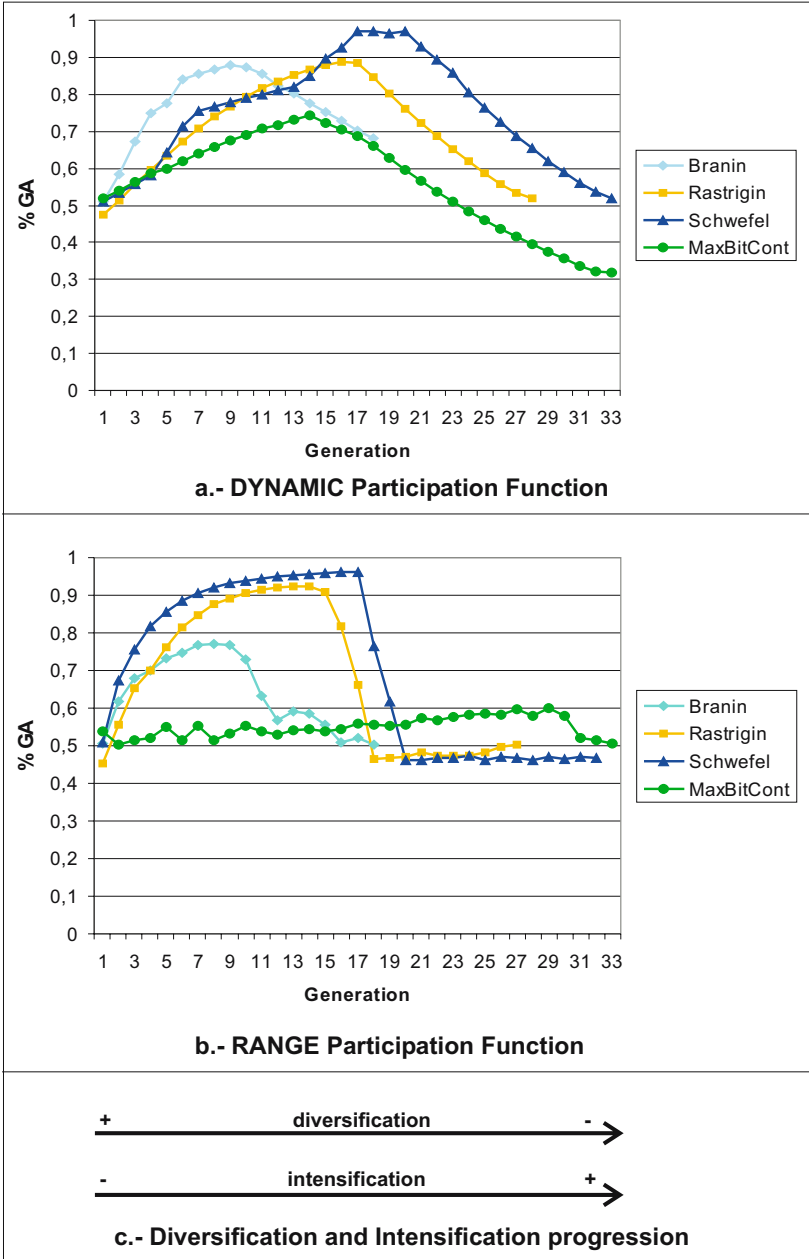


Fig. 2. Evolution of a.- DYNAMIC and b.- RANGE Participation Functions. Progression of Diversification and Intensification during the search

early steps of the process, during the first generations, genetic algorithms perform better, and therefore their participation ration increases. However, in a second stage, EDAs get profit from their better intensification performance and this characteristic causes that the ration of participation is inverted. The shape of this participation function is similar in all the experiments, and the variations are based on the specific nature of the problem itself.

RANGE participation function (Figure 2.b) presents a similar behavior in general, although (i) there is an abrupt transition between the region in which GAs exploit diversification and the moment in which EDAs are necessary to converge to the optimum value via intensification. (ii) in MaxBitCont problem there are similar proportions of both methods during all the evolution.

7 Conclusions and Future Work

In this contribution a new Participation Function for the hybrid GA-EDA algorithm has been presented. The new function provides a direct adaptability to the results achieved by each of the participating algorithms. This performance seems to fit better at the switching point in which the importance of the diversification decreases and intensification is more required to obtain the optimum value.

Besides, diversification and intensification of both GA and EDA algorithms have been analyzed. This study requires a deeper research to evaluate the theoretical benefits and the quantitative results of these two algorithms according to these concepts.

References

1. V. Bachelet and E. Talbi. Cosearch: A co-evolutionary metaheuristics. In *Proceedings of Congress on Evolutionary Computation CEC2000*, pages 1550–1557, 2000.
2. C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, September 2003.
3. F.K. Branin. A widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development*, pages 504–522, 1972.
4. J. Denzinger and T. Offerman. On cooperation between evolutionary algorithms and other search paradigms. In *Proceedings of Congress on Evolutionary Computation CEC1999*, pages 2317–2324, 1999.
5. F. Focacci, F. Laburthe, and A. Lodi. Local search and constraint programming. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research and Management Science*, Kluwer Academic Publishers, Norwell, MA.
6. F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
7. J. Hao, F. Lardeux, and F. Saubion. A hybrid genetic algorithm for the satisfiability problem. In *Proceedings of the First International Workshop on Heuristics*, Beijing, 2002.
8. F. Herrera and M. Lozano. Gradual distributed real-coded genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(1), 2000.
9. J.H. Holland. *Adaption in natural and artificial systems*. The University of Michigan Press, Ann Harbor, MI, 1975.
10. S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

11. P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. Optimization in continuous domains by learning and simulation of Gaussian networks. In A. S. Wu, editor, *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pages 201–204, 2000.
12. P. Larrañaga and J.A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2001.
13. F.T. Lin, , C.Y. Kao, and C.C. Hsu. Incorporating genetic algorithms into simulated annealing. *Proceedings of the Fourth International Symposium on Artificial Intelligence*, pages 290–297, 1991.
14. O.C. Martin and S.W. Otto. Combining simulated annealing with local search heuristics. *Annals of Operations Research*, 63:57–75, 1996.
15. C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.
16. J.M. Peña, V. Robles, P. Larrañaga, V. Herves, F. Rosales, and M.S. Pérez. GA-EDA: Hybrid evolutionary algorithm using genetic and estimation of distribution algorithms. In *Innovations in Applied Artificial Intelligence: Proceeding of IEA/AIE 2004*, 2004.
17. V. Robles, J.M. Peña, P. Larrañaga, M.S. Pérez, and V. Herves. GA-EDA: A new hybrid cooperative search evolutionary algorithm. In J.A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors, *Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms*. Springer Verlag, 2005. In Press.
18. V. Robles, M.S. Pérez, V. Herves, J.M. Peña, and P. Larrañaga. Parallel stochastic search for protein secondary structure prediction. In *Lecture Notes in Computer Science*, Czestochowa, Poland, 2003.
19. E-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(5):541–564, 2002.
20. A. Törn, M. M. Ali, and S. Viitanen. Stochastic global optimization: Problem classes and solution techniques. *Journal of Global Optimization*, 14, 1999.
21. M. Toulouse, T. Crainic, and B. Sansó. An experimental study of the systemic behavior of cooperative search algorithms. In I. Osman S. Voß, S. Martello and C. Roucairol, editors, *In Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, chapter 26, pages 373–392. Kluwer Academic Publishers, 1999.