A Study on Multimemetic Estimation of Distribution Algorithms

Rafael Nogueras and Carlos Cotta

Dept. Lenguajes y Ciencias de la Computación, Universidad de Málaga,
ETSI Informática, Campus de Teatinos, 29071 Málaga, Spain
ccottap@lcc.uma.es

Abstract. Multimemetic algorithms (MMAs) are memetic algorithms in which memes (interpreted as non-genetic expressions of problem-solving strategies) are explicitly represented and evolved alongside genotypes. This process is commonly approached using the standard genetic procedures of recombination and mutation to manipulate directly information at the memetic level. We consider an alternative approach based on the use of estimation of distribution algorithms to carry on this self-adaptive memetic optimization process. We study the application of different EDAs to this end, and provide an extensive experimental evaluation. It is shown that elitism is essential to achieve top performance, and that elitist versions of multimemetic EDAs using bivariate probabilistic models are capable of outperforming genetic MMAs.

1 Introduction

Memetic algorithms [8, 14] can be regarded as a pragmatic integration of ideas from population-based global search techniques and trajectory-based local search techniques [12]. One of the central tenets in the paradigm is the notion of *meme*, famously defined by Richard Dawkins as units of imitation [5]. Within this optimization context, memes translate to computational problem-solving procedures. While this definition is broad enough to encompass a wide variety of techniques, it is typically the case that memes are assimilated to local-search procedures. Even more so, these procedures are often fixed or pre-defined and therefore the MA can be regarded as operating with static implicit memes. This said, the explicit management of memes is a topic that has been around for some time now, and can be found in, e.g., multimemetic algorithms (MMAs) [9], in which each solution carries memes indicating how self-improvement is going to be conducted. Such memes are subject to evolution using the standard genetic procedures of recombination and mutation, thus conforming a self-adaptive search approach.

In this work we are going to consider the use of estimation of distribution algorithms (EDAs) [10, 11, 17] as the underlying search engine for multimemetic optimization. While the use of local search procedures in combination with EDAs is a widely-known approach to inject problem-dependent knowledge and improve the efficiency of the optimization process –see, e.g., [16,18,19,22]–the use of EDAs for self-adaptive memetic optimization has been less explored. The contribution of this work is taking some steps in this direction, studying different approaches for the application of EDAs to multimemetic optimization, and providing an extensive empirical evaluation of their performance.

2 Multimemetic EDAs

As mentioned above, the core idea of MMAs is the explicit treatment of memes within the evolutionary process. Hence, we shall firstly describe the representation of memes, before getting into the deployment of EDAs in this context.

2.1 Meme Representation and Application

Memes, conceived as non-genetic expressions of problem-solving strategies, can be represented in many ways depending on the level of abstraction and problem dependance considered. In this work we follow some ideas posed by Smith [20] in the context of pseudoboolean function optimization. Therein, memes are expressed as pattern-based rewriting rules [*condition*→*action*] as follows: let $[C \to A]$ be a meme, where $C, A \in \Sigma^r$ with $\Sigma = \{0, 1, \#\}$ and $r \in \mathbb{N}$ being some constant. In this ternary alphabet '#' represents a wildcard symbol; let $g_1 \cdots g_n$ be a genotype; a meme $[c_1 \cdots c_r \to a_1 \cdots a_r]$ could be applied on any substring of the genotype into which the condition fits, i.e., for which $g_i \cdots g_{i+r-1} = c_1 \cdots c_r$ (for this purpose, wildcard symbols in the condition are assumed to match any symbol in the genotype). If the meme were to be applied on position i, its action would be to implant the action $a_1 \cdots a_r$ in that part of the genotype, i.e., letting $g_i \cdots g_{i+r-1} \leftarrow a_1 \cdots a_r$ (in this case, wildcard symbols in the action are taken as don't-change symbols, that is, keeping unchanged the corresponding symbol in the genotype – we depart here from [20] in which wildcards in the action represented the binary complement of the original gene). The genotype is scanned in a randomized order to check for potential application sites of the meme so as to avoid positional bias. If a match is found the meme is applied and the resulting neighboring genotype is evaluated. A parameter w determining the maximal number of meme applications per individual is used to keep the total cost of the process under control. The best neighbor generated throughout the precess is kept if it is better than the current genotype.

2.2 EDA Approaches

The underlying idea in the MMA model considered is to have genetic and memetic information linked within a single individual, i.e., each individual carries a genotype and a meme. Once an individual is generated, its genotype is evaluated and the meme is subsequently applied in order to improve it. Whereas these individuals are generated by means of evolutionary operators –such as recombination and mutation– in standard genetic MMAs, multimemetic EDAs approach this by probabilistic sampling of a certain distribution which is evolved during the run. Let us focus on how this is done.

EDAs try to learn the joint probability distribution $p(\boldsymbol{x})$ representing the most promising individuals at each generation. Such generations comprise a cycle of (1) sampling $p(\boldsymbol{x})$ to obtain a population *pop*, selecting the most promising individuals pop' from *pop*, and updating $p(\boldsymbol{x})$ using pop'. Different EDAs can be considered depending on the way they model the joint probability distribution $p(\boldsymbol{x})$. In this work, we have considered the following ones:

- *Univariate models*: variables are assumed to be independent and hence the joint probability distribution $p(\boldsymbol{x})$ is factorized as

  $$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i).$$

  The simplest such EDA is UMDA [13], in which $p(x_i)$ is estimated as

  $$p(x_i = v) = \frac{1}{k} \sum_{j=1}^{k} \delta(pop'_{ji}, v),$$

  where $k = |pop'|$, pop'_{ji} is the value of the i-th variable of the j-th individual in pop', and $\delta(\cdot,\cdot)$ is Kronecker delta ($\delta(a,b) = 1$ if $a = b$ and $\delta(a,b) = 0$ otherwise). A generalization of UMDA is PBIL [2], an algorithm in which the probabilistic model is updated using a linear combination of its current value and the new value learnt from the sample, i.e.,

  $$p'(x_i = v) = (1-\eta)p(x_i = v) + \eta \frac{1}{k} \sum_{j=1}^{k} \delta(pop'_{ji}, v)$$

  for some learning rate parameter η ($0 < \eta \leqslant 1$). Note that PBIL reduces to UMDA for $\eta = 1$.
- *Bivariate models*: these models can capture low order dependencies by assuming relations between pairs of variables. More precisely, in the models considered here $p(\boldsymbol{x})$ is factorized as

  $$p(\boldsymbol{x}) = p(x_{i_1}) \prod_{j=2}^{n} p(x_{i_j} | x_{i_{a(j)}}),$$

  where $i_1 \cdots i_n$ is a permutation of the indices $1 \cdots n$, and $a(j) < j$ is the permutation index of the variable which x_{i_j} depends on. The particular EDAs considered within this class are based on MIMIC [3] and COMIT [1]. In the first case, we assume $a(j) = j - 1$ (i.e., each variable depends on the previous one in the permutation) and the permutation is built by picking i_1 as the variable with the lowest entropy $H(X_k)$ in the selected sample pop', and then picking i_j ($j > 1$) as the variable (among those not yet selected) that minimizes the conditional entropy $H(X_k|X_{i_{j-1}})$. Along this line, we build a COMIT-based approach by not restricting $a(j) = j-1$, thus picking

i_j as the variable that minimizes $H(X_k|X_{i_s}, s < j)$. Thus, while in the first case we have a linear dependence structure, in this second case we have a tree dependence structure. Note finally that in these bivariate multimemetic EDAs we compute separate models for both genotypes and memes.

In all cases, the probability estimation for model updating includes Laplace correction [4] in order to prevent premature convergence, always allowing a non-zero rate of exploration. Furthermore, we also consider for each of the EDAs presented an elitist counterpart[1], in which the new population is created by truncation selection from the union of of the selected sample in the previous step and the sample extracted from the current model. As shown by [7], Laplace-corrected elitist EDAs can converge to a population containing the global optimum.

3 Experimental Analysis

In order to analyze the performance of the multimemetic EDAs described in previous section we have considered a collection of pseudoboolean optimization problems. These are described in Sect. 3.1; subsequently we shall analyze the results in Sect. 3.2.

3.1 Benchmark and Settings

The test suite comprises four different problems defined on binary strings, namely Deb's trap function [6], Watson et al.'s hierarchically consistent test problems (HIFF and HXOR) [21] and Boolean satisfiability. These are described below.

Deb's 4-bit fully deceptive function (TRAP henceforth) is defined as

$$f_{trap}(b_1 \cdots b_4) = \begin{cases} 0.6 - 0.2 \cdot u(b_1 \cdots b_4) & \text{if } u(b_1 \cdots b_4) < 4 \\ 1 & \text{if } u(b_1 \cdots b_4) = 4 \end{cases} \tag{1}$$

where $u(s_1 \cdots s_i) = \sum_j s_j$ is the unitation (number of 1s in a binary string) function. This function is used as the basic block to build a higher-order problem by concatenating k such blocks, and defining the fitness of a $4k$-bit string as the sum of the fitness contribution of each block. In our experiments we have considered $k = 32$ subproblems (i.e., 128-bit strings, $opt = 32$).

As to the hierarchically consistent test problems, they are recursive epistatic functions defined for binary strings of 2^k bits by means of two auxiliary functions $f : \{0, 1, \times\} \to \{0, 1\}$ (used to score the contribution of building blocks), and $t : \{0, 1, \times\} \to \{0, 1, \bullet\}$ (used to capture the interaction of building blocks), where '$\bullet$' is used as a *null* value. In the case of the Hierarchical if-and-only-if (HIFF) function f and t are defined as:

$$f(a, b) = \begin{cases} 1 & a = b \neq \bullet \\ 0 & \text{otherwise} \end{cases} \qquad t(a, b) = \begin{cases} a & a = b \\ \bullet & \text{otherwise} \end{cases}$$

[1] Note that the original definition of COMIT was already intrinsically elitist. Here we consider both an elitist and a non-elitist version of this approach.

We use these two functions as follows:

$$\mathrm{HIFF}_k(b_1 \cdots b_n) = \sum_{i=1}^{n/2} f(b_{2i-1}, b_{2i}) + 2 \cdot \mathrm{HIFF}_{k-1}(b'_1, \cdots, b'_{n/2}) \qquad (2)$$

where $b'_i = t(b_{2i-1}, b_{2i})$ and $\mathrm{HIFF}_0(\cdot) = 1$. The Hierarchical XOR (HXOR) works similarly but changing f so as to provide a fitness contribution of 1 when $a = 1$ and $b = 0$ or vice versa, and having in that case $t(a, b) = a$ (and $t(a, b) = \times$ otherwise). We have considered $k = 7$ (i.e., 128-bit strings, $opt = 576$)

Finally, the Boolean satisfiability problem is a classical NP-complete problem which amounts to finding a truth assignment to n variables such that a certain Boolean formula Φ is satisfied. We consider this formula is expressed in conjunctive normal form with $n = 128$ variables and $k = 3$ variables per clause; this problem is known to have an easy-hard-easy phase transition when varying the ratio m/n. The difficulty peak is located around $m = 4.3n$. We use a problem generator approach in this case, generating a different satisfiable instance with this critical clauses/variable ratio ($opt = m = 550$) in each run of the MMA.

We consider multimemetic EDAs as described in Sect. 2, with a population size of $\mu = 128$ individuals. Selection is done by truncation, keeping the best 50% individuals to update the probabilistic model, and a learning rate $\eta = 0.1$ is used in PBIL. The memes are expressed as rules of length $r = 3$ and we consider $w = 1$ (one rewriting is done and kept if the solution is improved). In all cases the cost of applying a meme is accounted as a fractional evaluation (i.e., as the fraction of the fitness function that needs being recomputed due to genotypic changes) and added to the total number of evaluations. A run is terminated upon reaching 50,000 evaluations, and 20 runs are performed for each problem and algorithm. To gauge the results, we also include in the experimentation an equivalent evolutionary MMA –termed sMMA henceforth– using the same population size ($\mu = 128$) and a generational reproductive plan with binary tournament for parent selection, one-point crossover ($p_X = 1.0$), bit-flip mutation ($p_M = 1/\ell$, where $\ell = 128$ is the number of bits), local-search (conducted using the meme linked to the individual) and replacement of the worst parent (an inherently elitist strategy, following the model presented in [15] – previous experiments with a non-elitist sMMA yielded globally inferior results, and so did a MA using a fixed bit flip meme). In this sMMA, offspring inherit the meme of the best parent, which is then subject to mutation with probability p_M.

3.2 Experimental Results

Full numerical results are provided in Table 1. Firstly, notice that elitist versions of multimemetic EDAs (denoted by a subscript e) perform in general much better than their non-elitist counterparts. Furthermore, while the latter are in most cases inferior to the sMMA, elitist multimemetic EDAs provide top performance in all problems, and are also globally superior to non-memetic EDAs (results not shown). This is further investigated in Table 2. As it can be seen, the superiority

Table 1. Results (20 runs) of the different MMAs on TRAP, HIFF, HXOR and SAT. The median ($\tilde{x}$), mean ($\bar{x}$) and standard error of the mean ($\sigma_{\bar{x}}$) are shown.

	TRAP		HIFF		HXOR		SAT	
	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$
sMMA	31.4	30.0 ± 0.5	408.0	427.6 ± 13.9	360.0	360.2 ± 4.4	547.0	546.6 ± 0.4
UMDA	20.0	20.5 ± 0.2	363.0	385.9 ± 19.5	320.5	324.6 ± 5.2	548.0	548.0 ± 0.3
UMDA$_e$	23.9	25.2 ± 0.8	576.0	517.5 ± 16.9	348.0	341.6 ± 6.9	548.0	548.3 ± 0.2
PBIL	19.2	19.2 ± 0.0	276.5	275.1 ± 3.4	270.5	271.0 ± 2.4	543.0	543.8 ± 0.5
PBIL$_e$	24.6	25.7 ± 0.7	441.5	441.7 ± 12.1	259.0	258.6 ± 2.2	548.5	548.3 ± 0.3
MIMIC	20.3	20.2 ± 0.1	328.5	330.3 ± 5.0	310.0	312.9 ± 2.5	546.0	545.7 ± 0.4
MIMIC$_e$	31.6	31.3 ± 0.2	472.0	493.6 ± 16.3	393.5	397.6 ± 4.7	548.0	548.2 ± 0.2
COMIT	21.0	21.0 ± 0.2	337.5	342.9 ± 5.9	330.0	328.1 ± 3.7	548.0	548.0 ± 0.2
COMIT$_e$	32.0	32.0 ± 0.0	424.0	443.8 ± 12.2	408.0	419.4 ± 7.8	548.0	548.0 ± 0.3

Table 2. Statistical comparison among the different multimemetic EDAs using Wilcoxon ranksum ($\alpha = 0.05$). For each problem/EDA three symbols are provided, respectively indicating how the algorithm compares with its (non-)elitist counterpart, with sMMA, and with the algorithm with the highest median for the corresponding problem (which is marked with a star ⋆ in this third position). A white/black circle (○/●) indicates the algorithm labeled in the column has a worse/better median with statistical significance. A '=' sign indicates no statistically-significant difference.

	UMDA	UMDA$_e$	PBIL	PBIL$_e$	MIMIC	MIMIC$_e$	COMIT	COMIT$_e$
TRAP	○○○	●○○	○○○	●○○	○○○	●=○	○○○	●●⋆
HIFF	○○○	●●⋆	○○○	●=○	○○○	●●=	○○○	●=○
HXOR	○○○	●○○	●○○	○○○	○○○	●●○	○○○	●●⋆
SAT	=●=	=●=	○○○	●●⋆	○=○	●●=	=●=	=●=

of elitist algorithms over non-elitist ones is statistically significant in all cases, except in PBIL for HXOR and UMDA and COMIT for SAT. Moreover, the superiority of elitist algorithms over sMMA (mid-symbol in each entry of Table 2) is also statistically significant in most cases. Among the different elitist EDAs, COMIT$_e$ seems to provide the best overall results, being the top algorithm in TRAP (see Fig. 1 for an illustration of fitness evolution on this problem) and HXOR, and being indistinguishable from PBIL$_e$ in SAT. It is interesting to notice the good performance of UMDA$_e$ on this problem. This can be due to the fact that the optimal solution in this case is a homogeneous string (all 0s or all 1s), a structure which is easily captured by memes; as soon as the simpler nature of UMDA's probabilistic model directs the search towards a state with predominance of either symbol, the memes can facilitate reaching the optimum. This hypothesis is supported by the comparatively worse results of univariate algorithms on the HXOR problem, in which the optimal solution contains a 50%-50% mixture of 1s and 0s, placed in precise locations (so as to make any half of the solution be maximally dissimilar from the other half).

Let us now turn our attention to the effect that the utilization of Laplace correction has on the algorithms. While its use is fundamental within the geno-

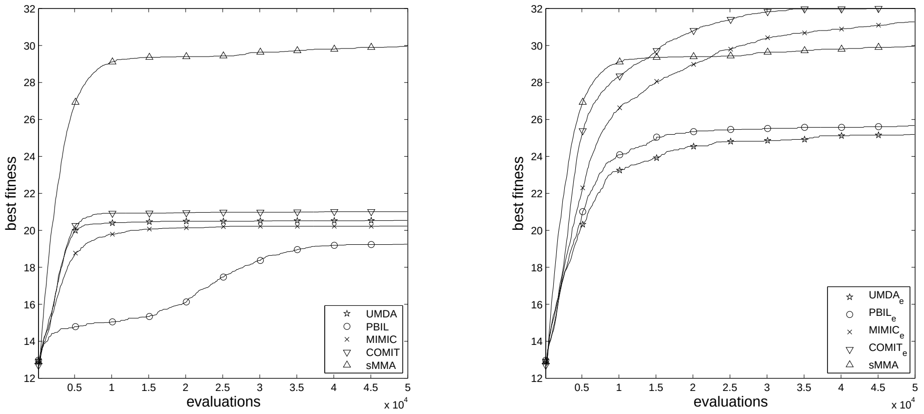

Fig. 1. Evolution of best fitness in TRAP for non-elitist multimemetic EDAs (left) and elitist ones (right). The results of sMMA are included in both figures.

Table 3. Results (20 runs) of the different MMAs on TRAP, HIFF, HXOR and SAT, without Laplace correction in meme probability estimation. The median ($\tilde{x}$), mean ($\bar{x}$) and standard error of the mean ($\sigma_{\bar{x}}$) are shown.

	TRAP		HIFF		HXOR		SAT	
	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$	$\tilde{x}$	$\bar{x} \pm \sigma_{\bar{x}}$
UMDA$_e^*$	23.8	25.2 ± 0.8	576.0	516.9 ± 17.3	342.0	342.8 ± 7.6	549.0	548.4 ± 0.3
PBIL$_e^*$	24.7	25.7 ± 0.7	443.0	451.5 ± 15.9	262.5	259.9 ± 2.2	549.0	548.6 ± 0.3
MIMIC$_e^*$	32.0	31.7 ± 0.2	464.0	473.6 ± 12.6	412.0	417.0 ± 5.3	548.5	548.5 ± 0.2
COMIT$_e^*$	32.0	32.0 ± 0.0	464.0	480.0 ± 13.9	416.0	427.6 ± 10.0	548.0	548.4 ± 0.3

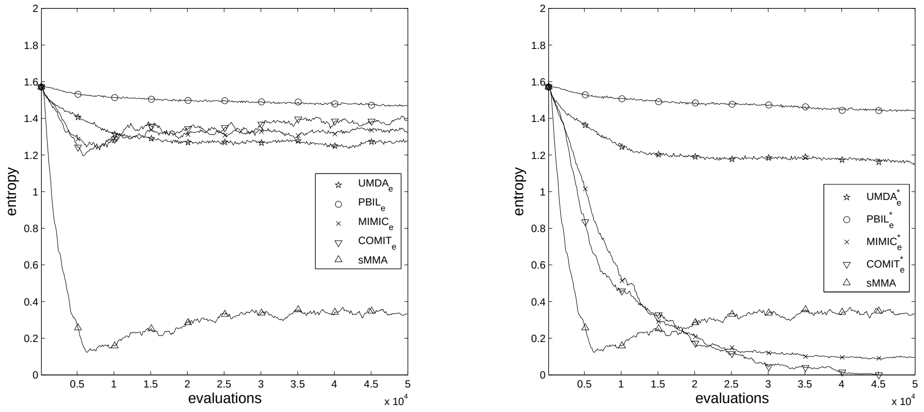

Fig. 2. Evolution of meme diversity in TRAP for multimemetic EDAs using Laplace correction in the probabilistic modeling of memes (left) and without using such correction (right).

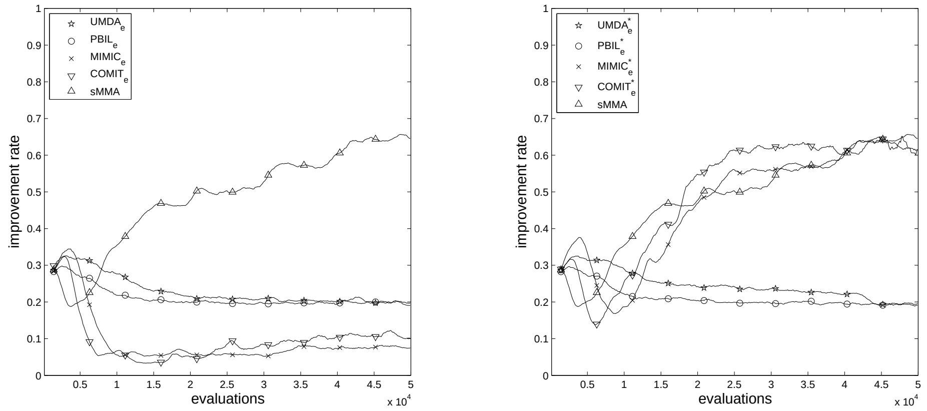

Fig. 3. Evolution of meme success (percentage of meme applications that result in an improvement) in TRAP for multimemetic EDAs using Laplace correction in the probabilistic modeling of memes (left) and without using such correction (right).

typic model (experiments without this correction indicate quick convergence of the probabilistic model to suboptimal states within a few dozens generations), its use at the memetic level seems more questionable. As it can be seen in Fig. 2 (left), the entropy of memes generated remains very high in all EDAs, unlike the sMMA which stabilizes at a low entropy level. This indicates that the EDAs are facing difficulties to focus the search of adequate memes, either by the limitations of the underlying probabilistic model or by the exploratory disturbance of the Laplace correction. We have therefore performed experiments with the elitist EDAs deactivating this correction in meme modeling. Not surprisingly, this has a larger influence in the bivariate models, which are now capable to converging to particular states, than in univariate models, which remain incapable of grasping the structure of memes in many cases – see Fig. 2 (right). This also has in general a positive influence in performance as shown in Table 3. While the univariate models perform slightly better, the difference is not statistically significant. In the case of the bivariate models, there is a statistically significant difference in favor of MIMIC$^*_e$ (the superscript * denoting deactivation of Laplace correction) for TRAP and HXOR, and in favor of COMIT$^*_e$ for HIFF. It is interesting to notice how meme success (the percentage of meme applications that result in an improvement) is higher in these variants – see Fig. 3 – supporting the hypothesis that the search is more focused in this case.

4 Conclusions

We have studied the use of EDAs in a multimemetic context. They appear to be a promising approach to this kind of self-adaptive memetic optimization due to their well-known advantages, namely their requiring less parameterization effort than their genetic counterparts and their amenability to model combinatorial structures such as memes. Indeed, the experimentation with multimemetic EDAs has provided encouraging results: when endowed with elitism, multimemetic EDAs are markedly superior to a MMA manipulating genes and memes using genetic operators. We have also observed that the memetic search is more focused when no Laplace correction is used in meme modeling. Of course, this may need further investigation in other contexts in which memes are represented in a different way (and indeed in the general context of EDA optimization, since the use of this technique is not widespread). This is not the only line for future research: on one hand, the use of more complex probabilistic graphical models such as Bayesian networks is an appealing option. As a matter of fact, it may be conceivable that the structure used to model the probability distribution over the memetic space be different than its genotypic counterpart. This could pave the way to a fully decoupled evolutionary model in which genotypes and memes evolve in different populations, subject to separate selection processes, and interacting via some strategy for genotype-meme pairing and application, in the line of coevolutionary memetic algorithms [20].

Acknowledgements This work is partially supported by MICINN project ANYSELF (TIN2011-28627-C04-01), by Junta de Andalucía project DNEMESIS (P10-TIC-6083) and by Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech.

References

1. Baluja, S., Davies, S.: Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In: 14th International Conference on Machine Learning. pp. 30–38. Morgan Kaufmann (1997)
2. Baluja, S.: Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Tech. Rep. CMU-CS-95-19, Carnegie Mellon University, Pittsburgh, PA, USA (1994)
3. Bonet, J.S.D., Isbell, C.L., Jr., Viola, P.: Mimic: Finding optima by estimating probability densities. In: Mozer, M., Jordan, M., Petsche, T. (eds.) Advances in Neural Information Processing Systems. vol. 9, pp. 424–430. The MIT Press (1996)
4. Cestnik, B.: Estimating probabilities: A crucial task in machine learning. In: Aiello, L. (ed.) 9th European Conference on Artificial Intelligence. pp. 147–149. Pitman, Stockholm, Sweden (1990)
5. Dawkins, R.: The Selfish Gene. Clarendon Press, Oxford (1976)
6. Deb, K., Goldberg, D.E.: Analyzing deception in trap functions. In: Whitley, L.D. (ed.) Second Workshop on Foundations of Genetic Algorithms. pp. 93–108. Morgan Kaufmann, Vail, Colorado, USA (1993)
7. González, C., Lozano, J., Larrañaga, P.: Mathematical modeling of discrete estimation of distribution algorithms. In: Larrañaga, P., Lozano, J.A. (eds.) Estimation of Distribution Algorithms, Genetic Algorithms and Evolutionary Computation, vol. 2, pp. 147–163. Springer US (2002)
8. Hart, W., Krasnogor, N., Smith, J.: Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing, vol. 166, chap. Memetic Evolutionary Algorithms, pp. 3–27. Springer, Berlin Heidelberg (2005)

9. Krasnogor, N., Blackburne, B., Burke, E., Hirst, J.: Multimeme algorithms for protein structure prediction. In: Merelo, J., et al. (eds.) Parallel Problem Solving From Nature VII, Lecture Notes in Computer Science, vol. 2439, pp. 769–778. Springer, Berlin (2002)
10. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms, Genetic Algorithms and Evolutionary Computation, vol. 2. Springer US (2002)
11. Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E.: Towards a New Evolutionary Computation, Studies in Fuzziness and Soft Computing, vol. 192. Springer Berlin Heidelberg (2006)
12. Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Tech. Rep. Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA (1989)
13. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions I. Binary parameters. In: Parallel Problem Solving from Nature – PPSN IV. Lecture Notes in Computer Science, vol. 1141, pp. 178–187. Springer Berlin Heidelberg (1996)
14. Neri, F., Cotta, C., Moscato, P.: Handbook of Memetic Algorithms, Studies in Computational Intelligence, vol. 379. Springer, Berlin Heidelberg (2012)
15. Nogueras, R., Cotta, C.: Analyzing meme propagation in multimemetic algorithms: Initial investigations. In: 2013 Federated Conference on Computer Science and Information Systems. pp. 1013–1019. IEEE Press, Cracow (Poland) (2013)
16. Pelikan, M., Goldberg, D.: Hierarchical BOA solves ising spin glasses and MAXSAT. In: Cantú-Paz, E., et al. (eds.) 2003 Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science, vol. 2724, pp. 1271–1282. Springer Berlin Heidelberg (2003)
17. Pelikan, M., Sastry, K., Cantú-Paz, E.: Scalable Optimization via Probabilistic Modeling, Studies in Computational Intelligence, vol. 33. Springer Berlin Heidelberg (2006)
18. Radetic, E., Pelikan, M., Goldberg, D.E.: Effects of a deterministic hill climber on hBOA. In: 2009 Genetic and Evolutionary Computation Conference. pp. 437–444. ACM, New York, NY, USA (2009)
19. Robles, V., Miguel, P., Larrañaga, P.: Solving the traveling salesman problem with EDAs. In: Larrañaga, P., Lozano, J.A. (eds.) Estimation of Distribution Algorithms, Genetic Algorithms and Evolutionary Computation, vol. 2, pp. 211–229. Springer US (2002)
20. Smith, J.E.: Self-adaptative and coevolving memetic algorithms. In: Neri, F., Cotta, C., Moscato, P. (eds.) Handbook of Memetic Algorithms, Studies in Computational Intelligence, vol. 379, pp. 167–188. Springer, Berlin Heidelberg (2012)
21. Watson, R.A., Pollack, J.B.: Hierarchically consistent test problems for genetic algorithms: Summary and additional results. In: 1999 IEEE Congress on Evolutionary Computation. pp. 292–297. IEEE Press, Washington D.C. (1999)
22. Zhang, Q., Sun, J., Tsang, E., Ford, J.: Estimation of distribution algorithm with 2-opt local search for the quadratic assignment problem. In: Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.) Towards a New Evolutionary Computation, Studies in Fuzziness and Soft Computing, vol. 192, pp. 281–292. Springer Berlin Heidelberg (2006)