DOI: 10.1007/978-3-642-28942-2_41

Initialization Procedures for Multiobjective Evolutionary Approaches to the Segmentation Issue

José L. Guerrero, Antonio Berlanga, and José Manuel Molina

University Carlos III of Madrid, Computer Science Department
Group of Applied Artificial Intelligence
Avda. Universidad Carlos III, 22, Colmenarejo, Spain
jguerrer@inf.uc3m.es aberlan@ia.uc3m.es molina@ia.uc3m.es

Abstract. Evolutionary algorithms have been applied to a wide variety of domains with successful results, supported by the increase of computational resources. One of such domains is segmentation, the representation of a given curve by means of a series of linear models minimizing the representation error. This work analyzes the impact of the initialization method on the performance of a multiobjective evolutionary algorithm for this segmentation domain, comparing a random initialization with two different approaches introducing domain knowledge: a hybrid approach based on the application of a local search method and a novel method based on the analysis of the Pareto Front structure.

Keywords: Initialization, Segmentation, Evolution Strategies, Multiobjective Optimization, Memetic algorithms.

1 Introduction

Evolutionary algorithms are a versatile tool to deal with a huge variety of problem domains [1], being convergence speed one of their known issues. Hybridization of intelligent systems [6] to overcome their handicaps is one of the most important trends in artificial intelligence, with applications in fields as diverse as robotics, dimensionality reduction, reasoning methods or multiobjective optimization [5]. Hybrid applications of evolutionary algorithms combine their exploratory capabilities with the information exploitation provided by local search procedures, being also known as memetic algorithms [12].

The initialization of the population in evolutionary algorithms has been considered as a key operator for the final quality of the result and the computational cost required to obtain that result [2]. The most popular initialization process consists in a random initialization of the values for the chosen representation. This procedure aims to maximize the coverage of the search space and, thus, the exploration capabilities of the algorithm. Different alternatives have been considered to improve this behavior, such as novel general alternative initialization strategies [14], reuse of previous solutions [15] or the introduction of specific domain information [3]. In general, default initialization processes are designed to provide a reasonable performance over a wide series of problems, but the injection of additional information may help to improve that performance over specific problem instances (following the no-free-lunch theorem [18]).

Segmentation problems are based on the division of a given curve in a set of n segments (being each of these segments represented by a linear model, which points to another common naming convention for this process: piecewise linear representation, PLR) minimizing the representation error. This issue has been faced from several perspectives, such as time series segmentation [11] or polygonal approximation [16], leading to different algorithms (some of which are closely related). Regarding evolutionary algorithms, they have been applied to this issue using different specific operators and focuses [19,9]. Multi-objective evolutionary algorithms optimize several objectives that may be in conflict with each other at the same time [4], and have been also applied to segmentation issues [7], dealing with the number of segments and representation error of given solutions to provide a Pareto Front of non-dominated solutions.

This paper analyzes the effect of different initialization methods on a multi-objective approach for the segmentation issue. The default random initialization method is compared with different alternatives, based on local search information exploitation (according to similar hybridization principles to the ones used by memetic algorithms, but applied to population initialization) or on the analysis of the Pareto front and its relationship to the input variables.

The work is organized as follows: the second section introduces the formalization of the segmentation issue as a multiobjective problem, along with an evolutionary approach to solve it. The third section covers three different initialization procedures, highlighting their characteristics and differences, while the fourth presents the used dataset, along with the results obtained for the techniques covered in the previous section. Finally, the conclusions obtained from these results are presented, along with possible future lines.

2 Segmentation Formalization

A segmentation process divides a series of data into a certain number of individual segments according to a model (or set of models) minimizing the representation error. This work is focused on PLR segmentation (or polygonal approximation) which uses linear models for this approach. This process can also be seen as the search of the individual points which minimize the overall approximation error. These points are usually called *dominant points.* A possible formalization for this process is presented in equation 1.

$$\begin{gathered} S(T) = \{B_m\} \to B_m = \{\boldsymbol{x}_k\} j \in [k_{min} ... k_{max}] m \in [1 ... seg_{num}] \\ \overset{min}{\underset{max}{}} f_{quality}(\{B_m\}) \end{gathered} \tag{1}$$

where T is the original data, *S(T)* is the segmentation process, B_m is a given resultant segment from that process and $f_{quality}$ is the used quality function.

Depending on the definition of this quality function, the objective may be to minimize or maximize this function. The quality of a segmentation process is traditionally determined by the following criteria [11]:

1. Minimizing the overall representation error (*total_error*)
2. Minimizing the number of segments such that the representation error is less than a certain value (*max_segment_error*)
3. Minimizing the number of segments so that the total representation error does not exceed *total_error*

The previous considerations introduce several interesting facts. First of all, the quality of a segmentation process depends on several objectives in conflict: minimizing the number of required segments while minimizing the representation error. Secondly, the configuration of such processes can be problem dependent due to the required parameters (*total_error* and *max_segment_error*) and, thus, a general technique may be hard to apply to a set of problem instances with consistent results. The consideration for the measurement of several objectives in conflict to test the quality of a segmentation process was faced in [8] with the use of multiobjective quality indicators [20]. Given the multiobjective nature of this process, in [7] the segmentation issue was formalized according to equation 2, which explicitly presents this nature.

$$S(T) = \{B_m\} \rightarrow B_m = \{\boldsymbol{x}_k\} j \in [k_{min} ... k_{max}] m \in [1 ... seg_{num}]$$
$$\begin{cases} d(S(T), T) \leq total_{error} \\ \forall m, d(f_{ap}(B_m), B_m) \leq max\_segment\_error \end{cases} \tag{2}$$

where *d(x, y)* is a distance error function between segments x and y and $f_{ap}(x)$, which is the approximation function result over segment x. According to the traditional criteria, *total_error* and *max_segment_error* represent certain constrains which are required by certain segmentation algorithms in order to determine whether they must be stopped [11]. It must be noted that these characteristics may change abruptly among different problem instances, even though they may be used to constrain the search process of the evolutionary approach. Since these parameters are not strictly required and their choice is neither trivial nor problem independent, they have been excluded from the presented approach to provide a more focused discussion.

In [7] the proposed codification is based on integer values representing the *dominant points* of the segmentation process (the edges of the segments which the data is divided into). This representation was introduced in order to preserve the importance of the different dominant points obtained during the evolutionary process, but also implied an increase over the size of the search space when compared to a more commonly used genetic codification [19]. This codification formalizes a representation where each chromosome has a size equal to the length of the data being analyzed, and each gene represents whether that particular position is considered a *dominant point*. This codification is the one followed in the current approach, due to its reduced search space.

Several error functions may be used as well. Two different fitness functions used in evolutionary approaches to segmentation are the maximum error function (equation 3) and the integral squared error function (equation 4). This last option is the one followed in this work.

$$E_\infty(\alpha) = \underset{1 \leq i \leq n}{max}\, e_i(\alpha) \tag{3}$$

$$E_2(\alpha) = \sum_{i=1}^{n} [e_i(\alpha)]^2 \tag{4}$$

The remaining operators are chosen according to standard values: bit-flip mutation, 1-point crossover and binary tournament selection. The crossover probability used is 0.9 and the mutation probability is $1/length$. These operators are not problem specific (nor their associated probabilities), differing from some of the single objective approaches available. The general multiobjective algorithm chosen is SPEA2 [21], which introduces an archive to keep track of the best solutions found during the evolutionary cycles of the different generations. In this problem it is crucial to preserve non-dominated solutions found at different points of the evolutionary cycle, leading to the choice of this algorithm along with an archive size equal to the length of the problem instance being solved, in order to be able to, ideally, store one non-dominated solution for each of the different possible representations regarding their number of dominant points, while the chosen population size will be 100 individuals.

3 Population Initialization

Convergence speed is a constant issue in evolutionary computation, and it has been approached with modifications in the different involved processes: crossover, mutation, selection, etc. Initialization procedures have received a reduced amount of interest from the research community, generally assuming that the overall impact over the performance of the algorithm is lower. Most genetic algorithms use a default bitstring uniform initialization procedure, assigning values of 0 or 1 to every bit for each individual in the population, obtaining a uniform population regarding the binary space, which also exhibits the maximal bit-wise diversity [10]. However, early research showed that this may not be the optimal initialization procedure for specific domains, such as inverse problems in Structural Mechanics [17], where the solutions were known to contain far more 0's than 1's.

General approaches have to provide a trade-off between the improved initial population obtained and the cost of the process. Such a discussion is carried out in [14], where opposition-based and quasi-random [13] initialization methods are compared, highlighting the computational issues and dimensionality effectiveness. In [15] the reuse of previous solutions in terms of population initialization is considered for the application of evolutionary algorithms to dynamic environments, but the established principles can be used for static environments where an approximation to the solution is known (or can be calculated, as in the local search based method compared in this work). Finally, domain specific approaches introduce characteristics from the faced problem in order to include a seeding in the initial population which can improve the overall results. In [3] such an approach is studied for the timetabling problem, where heuristic individuals go through some randomization process in order to generate the initial population, presenting a discussion of the diversity effect of such a process over the final outcome of the algorithm.

Three different initialization procedures will be compared for the presented problem: default (bitstring uniform), uniform (in terms of Pareto front) and local search. Default initialization assigns a 50% chance of becoming a *dominant point* to each point in the original data. According to that probability, this method generates an initial population which, in the number of segments objective function, is centered around 1/2 of the number of original elements in the data. Being this objective also closely related to the representation error, this generates a poor diversity on the number of segments (or, similarly, the number dominant points), which also implies a poor diversity on the covered range of approximation errors.

Even though default initialization produces the maximal bit-wise diversity, a poor one is obtained in the resultant Pareto front. Since multiobjective optimization seeks the Optimal Pareto Set in the variable space and its associated Optimal Pareto Front in the objective space (the set of solutions where one solution objective function value cannot be improved unless another objection function value is degraded [4]), this may not be the optimal strategy. Uniform initialization tries to ensure the diversity of the front obtained. To achieve this task, each individual is generated according to a number of random dominant points, which are then included into the chromosome at random gene positions. This generates a population which is spread along the dominant points objective, obtaining as well a good diversity over the representation error objective function. Related to the initialization approaches presented at the beginning of this section, this approach is general (in terms of exploiting the Pareto front diversity in the initial population) but uses a domain specific procedure to produce the front with a very low computational cost.

Local search initialization is a heuristic seeding approach using bottom-up segmentation [11] to introduce good individuals into the initial population, a technique which is claimed to obtain comparatively better results than other offline alternatives. This algorithm creates the finest possible approximation of the time series, dividing it into n-1 (where n is the number of points in the time series) segments of length value 2. Afterwards, the cost of merging each pair of adjacent segments is calculated and, if the merge with the lowest cost has an error bellow the user defined value, the segments are merged. The process continues until no pair of adjacent segments can be merged with an acceptable error value. It is important to notice that in every step of the algorithm the costs of the adjacent segments to the merged one in the previous step must be updated.

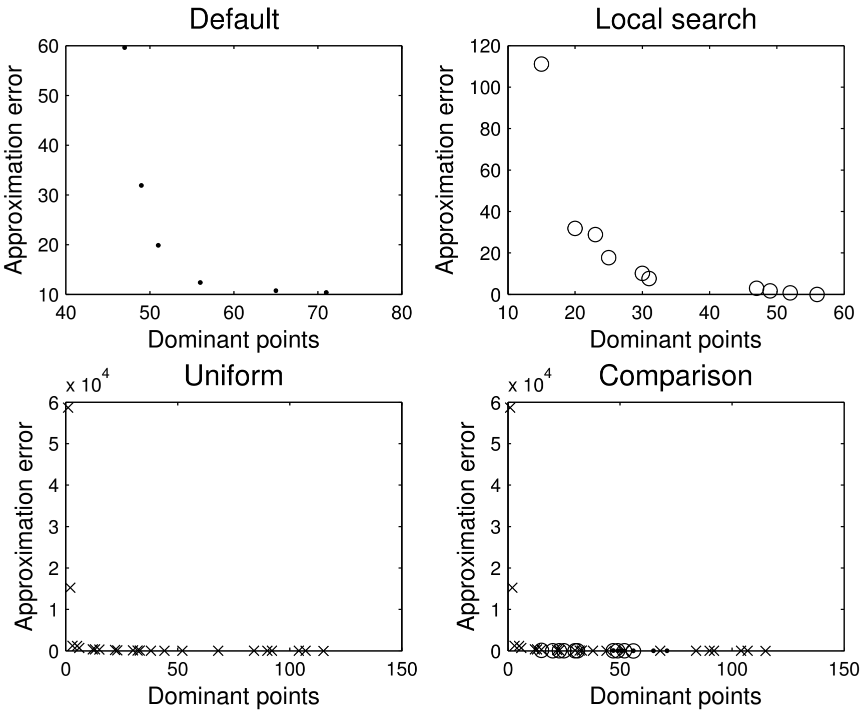


Fig. 1. Initial Pareto front comparison for the three presented methods (leaf curve)

One of the difficulties arising in the application of these single objective procedures is that, in order to obtain a certain number of different individuals to be introduced into the initial population, there is a lack of direct control over the objective functions values. This may require several executions to obtain a single individual which can be introduced into the population, thus increasing the overall computational cost. On the other hand, unlike other presented alternatives in the literature ([3]) different individuals are obtained with the different configuration parameters directly from the heuristic technique, eliminating the requirement for additional randomization processes.

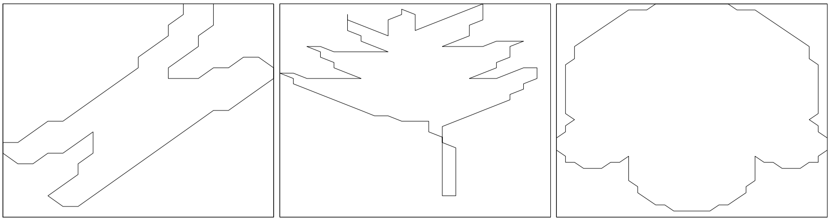
Figure 1 presents the non-dominated solutions obtained in an initial population of 100 individuals generated with the default method and the proposed approach based on the diversity in the objective space, along with a Pareto front composed from ten solutions obtained with different runs of the detailed single-objective algorithm. As expected, the range in the objective functions covered by the default initialization is very limited compared to the one which focuses on objective function diversity. The number of non-dominated individuals generated is clearly inferior to those in the uniform approach as well, obtaining an initial population which, even though it is composed of the same number of individuals, provides the algorithm with less valuable information (Pareto front individuals).

Local search initialization provides individuals which are clearly superior to the ones randomly initialized (by either of the alternative procedures), but their range is limited compared to the ones performed by uniform initialization.

4 Experimental Validation

Along with the performance of the presented initialization methods, current experimental validation will try to determine whether the inclusion of local search individuals in the population generated by either of the alternative methods improves its results. Three commonly used curves from the polygonal approximation domain are introduced into the data set: chromosome, semicircle and leaf, presented in figure 2. For the validation of the performance of the different initialization methods, 30 runs of every configuration have been performed, the unary hypervolume [20] of the resultant Pareto Front calculated for each of the alternatives (both for the initial and final populations), and the difference between the different pairings calculated. Afterwards, a t-test is carried out to determine the statistical significance of the obtained results. The reference front used for the hypervolume computation is obtained with a uniform initialization procedure and a population size of 1000 individuals left to run for 2000 generations.

The representation of the initial population Pareto fronts for the three curves in the dataset are presented in figures 1 (leaf), 3 (chromosome) and 4 (semicircle). Graphically these figures show several interesting facts regarding the proposed initialization: assuming that the heuristic seeding provided by the local search technique provides good solutions in terms of objective functions values and diversity, the comparison with the default process shows that bitstring uniform populations may provide good (figure 3) or very bad solutions (figure 1), being this quality problem dependent (determined by whether the solutions around 50% dominnant points are meaningful or not for the final Pareto front), disencouraging the use of this technique for an unknown problem instance. On the other hand, the initial populations provided by the uniform method exhibit for all the different dataset instances Pareto fronts with a very good diversity over the two objectives, being thus applicable to new unknown instances with a certain guarantee over the quality of the initial population's Pareto front.



(a) Chromosome curve (b) Leaf curve (c) Semicircle curve

Fig. 2. Curves included in the data set

Table 1. Initial populations comparison

Default		L.S.	Uniform		Unif. + l.s.		Def. + l.s.	
Chromosome curve								
Mean	Std	Mean	Mean	Std	Mean	Std	Mean	Std
4.47E-01	4.39E-02	8.59E-01	9.54E-01	7.52E-03	9.60E-01	6.82E-03	8.59E-01	9.94E-07
Leaf curve								
1,66E-01	3,22E-02	7,45E-01	9,62E-01	1,99E-02	9,63E-01	1,99E-02	7,45E-01	3,39E-16
Semicircle curve								
2,80E-01	5,21E-02	8,08E-01	9,50E-01	2,42E-02	9,51E-01	2,42E-02	8,08E-01	4,52E-16

Table 2. Final populations comparison

Default		Uniform		Unif. + l.s.		Def. + l.s.	
Chromosome curve							
Mean	Std	Mean	Std	Mean	Std	Mean	Std
9,41E-01	2,97E-02	9,67E-01	1,07E-04	9,66E-01	4,70E-03	9,67E-01	1,07E-04
Leaf curve							
7,77E-01	4,76E-02	9,79E-01	3,55E-03	9,76E-01	1,20E-02	9,77E-01	7,58E-03
Semicircle curve							
8,46E-01	4,06E-02	9,75E-01	5,27E-03	9,76E-01	3,08E-03	9,77E-01	3,39E-04

The hypervolume results obtained for the three different curves are presented in tables 1 and 2, while the statistical significance results over those values are presented in table 3. The initial populations comparison does not provide a standard deviation value for the local search initialization, since each of the runs starts with the exact same initial population. In final populations, no results for local search are provided, since, as will be detailed, the populations obtained by local search dominate those created by a default initialization process, providing the same final results (disregarding the stochastic nature of evolutionary approaches) in local search and local search plus default initialization configurations (being these results included under this last heading).

The test results presented in table 3 are obtained from the final populations, since all the differences in the initial ones were statistically significant. The results show that uniform initialization yields better performance of the algorithm compared to any of the remaining alternatives, and also that the addition of local search individuals to its initial population does not improve its results (in the final outcome of the algorithm). However, local search use does improve (for the two harder problem instances, lead and semicircle) the default initialization performance.

The initial populations provided by the different runs of a default initialization procedure become, in general, fully dominated by the individuals introduced by the local search (results in table 1 for local search and local search plus default individuals are the same). The impact of the local search procedures is related

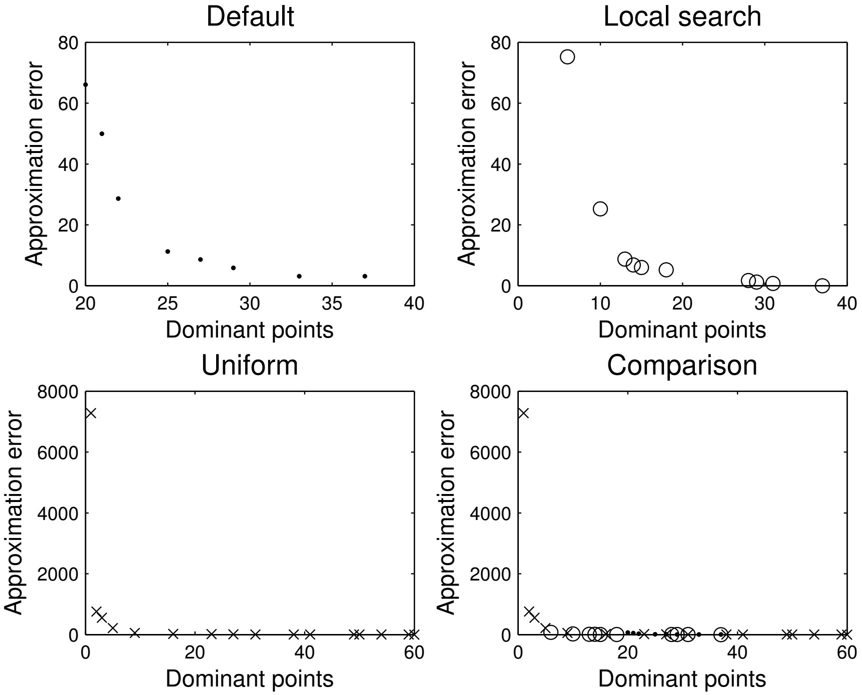


Fig. 3. Initial Pareto front comparison for the chromosome curve

to the quality of its results compared to the optimal Pareto front and the cost of their computation. As presented in table 3, the heuristic seeding does improve the results of the bitstring random initialization process (in two of the three curves in the dataset), but also requires a computational cost to obtain those individuals. As previously explained, obtaining n individuals for this initial population by means of the local search procedure may require more than n executions of this algorithm, and this cost may be even higher if certain diversity is required in those heuristic individuals.

Uniform initialization provides a higher range of objective function values to its individuals (which are graphically represented by the initial and final *"tails"* of the Pareto front), which provides additional non-dominated individuals to the algorithm and allowing it to obtain better final solutions, as seen in

Table 3. Statistical significance test

Curve	Def/l.s.	Def./Unif.	Unif./l.s.	Unif./Unif. + l.s.	Def./Def. + l.s.
Chromosome	No	Yes	Yes	No	No
Leaf	Yes	Yes	Yes	No	Yes
Semicircle	Yes	Yes	Yes	No	Yes

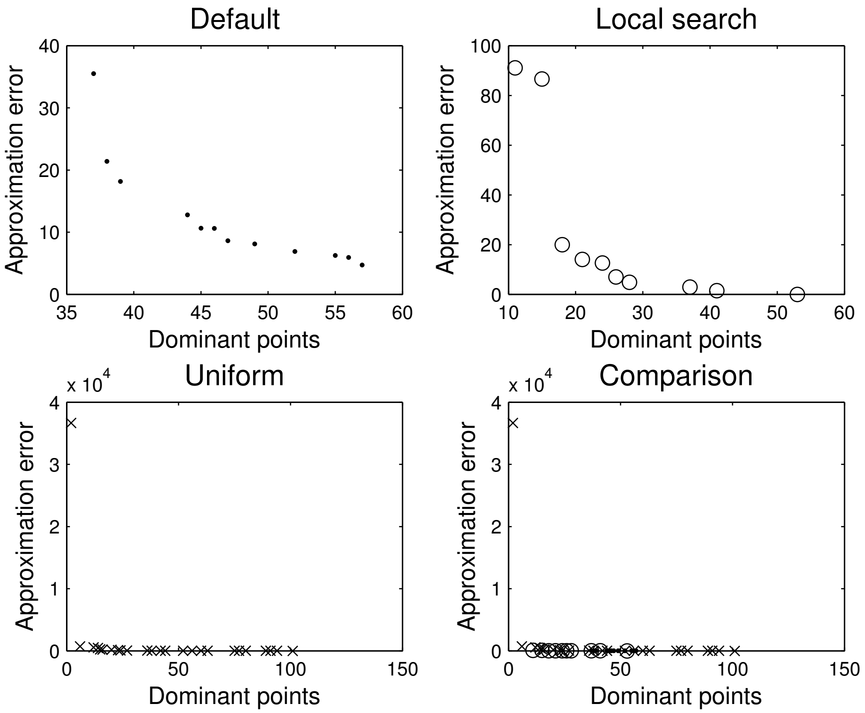


Fig. 4. Initial Pareto front comparison for the semicircle curve

table 3. This shows the importance of the diversity in terms of objective space, which cannot be obtained with the default bitstring random initialization. Even though there is no general technique to be able to obtain this diversity in the objective space for a general problem, the presented technique allows to do so in the segmentation domain with a very low computational cost (similar to that of the default initialization process) being clearly superior to the considered alternatives.

5 Conclusions

This work has presented the importance of initialization for evolutionary approaches, particularly for the segmentation issue. A common approach to include domain information into an evolutionary approach is to perform a hybridization including some local search step, which involves a considerable computational cost but aims to accelerate the exploitation step of the search, with the possible degradation of exploration capabilities. An overview of different initialization approaches is presented, and a comparison among three different possibilities is carried out: random initialization, a hybrid initialization including individuals obtained by means of a local search based procedure and a uniform approach based on the analysis of the Pareto Front shape in order to obtain an initial population focused on the diversity of individuals in the objective space. This uniform approach yields a performance not only better than the one provided by default initialization, but also superior to the one provided by a local search based initial population. The addition of local search individuals to an initial population generated by the uniform approach showed no statistically significant improvements in the outcome of the algorithm.

The measured improved performance comes from the amount of valuable information contained in the Pareto Front obtained: the increased covered ranges of objective function values by the individuals in the initial population provide a higher number of non-dominated individuals, which allows a better final performance of the algorithm, highlighting the importance of diversity in the objective space rather than the decision variables space. Future lines of this work include the inclusion of local search procedures at different steps of the evolutionary algorithm, additional research on initial population creation methods and the study of the applicability of these techniques to different multiobjective problems.

Acknowledgments. This work was supported in part by Projects CICYT TIN2011-28620-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

References

1. Affenzeller, M., Winkler, S.: Genetic algorithms and genetic programming: modern concepts and practical applications. Chapman & Hall/CRC (2009)
2. Bramlette, M.: Initialization, mutation and selection methods in genetic algorithms for function optimization. In: Proceedings of the Fourth International Conference on Genetic Algorithms, vol. 100, p. 107. Morgan Kaufmann, San Mateo (1991)
3. Burke, E., Newall, J., Weare, R.: Initialization strategies and diversity in evolutionary timetabling. Evolutionary Computation 6(1), 81–103 (1998)
4. Coello, C., Lamont, G., Van Veldhuizen, D.: Evolutionary algorithms for solving multi-objective problems. Springer-Verlag New York Inc. (2007)
5. Corchado, E., Abraham, A., de Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
6. Corchado, E., Graña, M., Wozniak, M.: Editorial: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
7. Guerrero, J., Berlanga, A., García, J., Molina, J.: Piecewise linear representation segmentation as a multiobjective optimization problem. Distributed Computing and Artificial Intelligence, 267–274 (2010)
8. Guerrero, J., García, J., Molina, J.: Piecewise linear representation segmentation in noisy domains with a large number of measurements: the air traffic control domain. International Journal on Artificial Intelligence Tools 20(2), 367–399 (2011)
9. Ho, S., Chen, Y.: An efficient evolutionary algorithm for accurate polygonal approximation. Pattern Recognition 34(12), 2305–2317 (2001)
10. Kallel, L., Schoenauer, M.: Alternative random initialization in genetic algorithms. In: Proceedings of the 7th International Conference on Genetic Algorithms, pp. 268–275 (1997)

11. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. Data mining in time series databases, pp. 1–21 (2003)
12. Krasnogor, N., Smith, J.: A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation 9(5), 474–488 (2005)
13. Maaranen, H., Miettinen, K., Mäkelä, M.: Quasi-random initial population for genetic algorithms*. Computers & Mathematics with Applications 47(12), 1885–1895 (2004)
14. Rahnamayan, S., Tizhoosh, H., Salama, M.: A novel population initialization method for accelerating evolutionary algorithms. Computers & Mathematics with Applications 53(10), 1605–1614 (2007)
15. Ramsey, C., Grefenstette, J.: Case-based initialization of genetic algorithms. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 84–91 (1993)
16. Sarfraz, M.: Linear Capture of Digital Curves. In: Interactive Curve Modeling, pp. 241–265. Springer, London (2008)
17. Schoenauer, M.: Shape representations and evolution schemes. In: Proceedings of the Fifth Annual Conference on Evolutionary Programming, pp. 121–129 (1996)
18. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)
19. Yin, P.Y.: A new method for polygonal approximation using genetic algorithms. Pattern Recognition Letters 19(11), 1017–1026 (1998)
20. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., Da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)
21. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, EUROGEN 2001, Athens, Greece, pp. 95–100 (2001)