Multi-Pareto-Ranking Evolutionary Algorithm

Wahabou Abdou, Christelle Bloch, Damien Charlet, and François Spies

University of Franche-Comté / FEMTO-ST Institute
1 Cours Leprince-Ringuet, 25201 Montbéliard, France
{firstname.lastname}@univ-fcomte.fr

Abstract. This paper proposes a new multi-objective genetic algorithm, called GAME, to solve constrained optimization problems. GAME uses an elitist archive, but it ranks the population in several Pareto fronts. Then, three types of fitness assignment methods are defined: the fitness of individuals depends on the front they belong to. The crowding distance is also used to preserve diversity. Selection is based on two steps: a Pareto front is first selected, before choosing an individual among the solutions it contains. The probability to choose a given front is computed using three parameters which are tuned using the design of experiments. The influence of the number of Pareto fronts is studied experimentally. Finally GAME's performance is assessed and compared with three other algorithms according to the conditions of the CEC 2009 competition.

Keywords: Multiobjective optimization, genetic algorithm, Pareto ranking, multiple fronts

1 Introduction

Many multi-objective evolutionary algorithms (MOEAs) use an elitist archive based on Pareto domination. The main difficulty of these MOEAs often lies in maintaining diversity. Thus, various solutions have been proposed as the crowding distance [6], metrics based on diversity [18] or specific definitions of fitness [8]. In the same way, the algorithm proposed in this paper, named Genetic Algorithm with Multiple parEto sets (GAME), rests on the concept of Pareto ranking. But it uses several Pareto fronts to rank individuals. Besides, the formula used to compute the fitness of individuals varies according to the set they belong to. Finally selection operators (for reproduction and for replacement) work in two steps: selecting one of the Pareto sets and selecting one individual in this set. The probability that each Pareto set is selected both depends on its rank and its size. Individuals belonging to the selected Pareto set are then compared using a fitness-based tournament. Defining various kinds of fitness depending on the non-domination rank preserves both quality and diversity. Section 2 quickly describes previous work related to this subject : some classical MOEAs based on Pareto archives, tuning parameters and managing constraints in such algorithms. Section 3 details the proposed algorithm itself. Section 4 first explains how the parameters used to define the selection probability of each Pareto front

are tuned using the design of experiments. This section then experimentally studies the benefit obtained by using several Pareto fronts. Finally that section presents the experiments performed to validate this approach, particularly the algorithms compared with it in accordance with the CEC2009 competition conditions, and the assessment of GAME efficiency. Section 5 concludes by highlighting and gives the main prospects of this work.

2 Multi-objective evolutionary algorithms (MOEAs) in literature

2.1 Pareto-based MOEAs

Among several evolutionary algorithms which archive non-dominated solutions, we can cite Strength Pareto Evolutionary Algorithm 2 (SPEA2) [18] or a newer method such as the Fast Pareto Genetic Algorithm (FastPGA) [8].

SPEA2 is an elitist multi-objective evolutionary algorithm that relies on Pareto-based and archiving techniques. SPEA2 assigns a strength to each S solution of the current population and the archive. This strength represents the number of solutions dominated by S. The fitness of each solution equals the summation of the strength of solutions that it is dominated by. When two solutions have the same fitness value, they are discriminated according to a metric based on the density (an adaptation of the k-th nearest neighbor method) [13].

FastPGA divides the solutions of the current population into two fronts (non-dominated and dominated sets) using the Pareto dominance principles. The fitness of non-dominated solutions is calculated using the crowding distance proposed by Deb et al. [5]. For solutions in the dominated front, FastPGA uses an extension of SPEA2's fitness assignment method. Indeed, the fitness of each dominated solution equals the summation of the strength values of all the solutions it dominates minus the summation of the strength values of all the solutions by which it is dominated. This reduces the chances that two solutions may have the same value of fitness. FastPGA also uses a specific mechanism to regulate population size over generations.

2.2 Tuning of MOEA parameters

Evolutionary algorithms include several parameters the tuning of which is generally difficult. Their values might significantly impact the quality of the solutions provided by the algorithm. A high mutation probability for instance would favor diversity and limit the exploitation of zones where good solutions may have been found. The algorithm would then nearly behave like a random search strategy. Such parameters are often tuned empirically. But it is difficult to find values which are both suitable and robust (i.e. achieving good enough performance without being too sensitive to the tackled optimization problems). This approach often requires a great number of executions and the knowledge of a human decision-maker to select appropriate values. That is why, in recent years,

several approaches have been proposed to design and/or tune EAs statistically in order to better guarantee their performance and/or robustness [9] [2].

2.3 Constraint-handling in MOEAs

Some optimization problems include constraints associated with the search space (set of variables) or the objective space (set of objective functions). Only points that satisfy these constraints are considered as feasible solutions of the optimization problem. Michalewicz and Schoenauer [12] define the types of approaches to handle such constraints in EAs. There are methods that preserve solution feasibility whenever possible. They use some operators (crossover and mutation) which only create solutions belonging to the set of feasible solutions. There also exists methods that include a penalty function in the calculation of the fitness of each solution. This allows to weaken the fitness of solutions that violate constraints. Another approach to handle constraints combine evolutionary algorithms and other optimization methods. They may repeatly use EAs. For instance, Surry et al. [15] rank individuals according to both objective values and constraint violations, which amounts to adding some objectives to the problem.

The genetic algorithm proposed in this paper uses a penalty-based approach to manage the constraints (see section 3.1). In addition, this algorithm uses both the computation of several Pareto fronts like NSGA-II and a definition of fitness which is different for each front like SPEA 2. The proposed solution combines the advantages of these two famous algorithms. Nevertheless contrary to NSGA-II structuring the population in several Pareto fronts is not only used for the replacement selection, it also impacts the recombination process. These characteristics have led to the implementation of a new selection strategy that takes into account the multiplicity of fronts and the fitness functions that depend on rank of the front. The benefit related to the number of fronts and the tuning of this new selection process parameters have been studied statistically.

The following section describes the proposed algorithm while particularly detailing its main characteristics.

3 A multiple-Pareto-ranking genetic algorithm.

The proposed algorithm includes a wide-ranking process and a specific method of fitness assignment. The main goal of these operations is both to ensure a good exploration of the space search and the convergence toward the optimal solution(s). This algorithm is named Genetic Algorithm with Multiple parEto sets (GAME). It rests on a classical global scheme and is quite like other MOEAs developed in literature. Initial population is generated randomly. Individuals represent solutions which are made of several integer or float variables, using binary simulated encoding. This representation is quite classical [14]: each variable is represented as a binary chain. The chain length (number of bits) depends on the wished precision. For instance, when representing a variable that takes values in the interval [0; 10], with a precision equals to 10^{-2}, requires 10 bits. The chains

0000000000 and 1111111111 will respectively represent 0 and 10. 1,23 will be encoded as 0001111110.

Crossover and mutation operators are also commonly used in this kind of representation. They are 2-point crossover and bitflip mutation [5]. GAME also uses parallel , based on master-slave model [4].

Besides, GAME differs from well-known MOEAs in several ways. These differences deal with the following steps: the ranking of individuals, based on Pareto dominance (3.1), fitness assignment (3.2), and the associated selection strategies for both recombination and replacement (3.3).

3.1 Ranking

GAME uses the objective values computed at the evaluation step to rank individuals. This ranking is based on the principle of non-dominated sorting (Pareto dominance). Pareto solutions are those for which improvement in one objective implies the worsening of at least one other objective. All the non-dominated individuals are added to the first Pareto front PF_1. This process is repeated successively three times with the remaining subset of individuals to build the second, the third and the fourth Pareto fronts (PF_2, PF_3 and PF_4). Finally, the remaining subset solutions are gathered in the set of dominated solutions PF_5.

When the problem has some constraints, they are handled during the construction of the Pareto fronts. GAME uses a penality-based method to take constraints intoaccount. In order to preserve the primacy of solutions which respect the limits, those which violate them are banned in the first Pareto front. However, when PF_1 is empty, solutions that do not satisfy the constraints can be accepted. This situation often occurs in the first generations. But it is quickly corrected as the solutions which respect the constraints have a higher probability of reproduction and thus transmit their characters to their offspring.

3.2 Fitness assignment

Since it is not obvious to compare two solutions in the same Pareto front (improvement in an objective is always to the detriment of at least one other), a fitness value is assigned to each solution.

The computation of the value of fitness depends on the Pareto rank of the solution. GAME distinguishes three types of fitness: one for the best solutions (the archive of non-dominated solutions and the first Pareto front), the second for the solutions of the second Pareto front and the third one for the other fronts. These fitness values are only used within the same front. The fitness of two individuals of different fronts cannot be compared. In this case, the comparison will only focus on the ranks of their respective fronts.

Fitness for the archive and the first Pareto front solutions. During the execution of GAME, the first Pareto front (PF_1) contains the best solutions of the current population. The archive of non-dominated solutions contains all the

best solutions evaluated since the first generation. Note that a solution of PF_1 can dominate some solutions in the archive. These two sets represent solutions that are closest to the optimal solutions (PF^*). While trying to get closer to PF^*, the algorithm must preserve diversity in the best solutions set. GAME assigns to the archived solutions and PF_1 a fitness based on the *crowding distance* [6] which is a diversity indicator. This indicator provides information on the distribution of solutions in the front. To avoid bias due to the sizes of ranges of the objective functions, GAME uses a normalized crowding distance.

Fitness for the second Pareto front solutions. The solutions of the second Pareto front (PF_2) are the second best set of solutions in a population. GAME favors solutions of the PF_2 that are closest to PF_1. The convergence indicator used to measure the proximity of solutions of the two fronts is the Generational distance. Using this indicator as fitness enables the solutions of PF_2 that are closest to PF_1 to have a greater probability to be selected for recombination and/or replacement.

Fitness for the remaining solutions For solutions that do not belong to any of the three previous sets (archive of non-dominated solutions, first and second Pareto fronts), a new indicator is used to calculate the values of fitness. This is the *gain* (see Equation 1).

The gain of an $\overrightarrow{x_i}$ solution with respect to $\overrightarrow{x_j}$ in accordance with the f_k objective function is called $gain(\overrightarrow{x_i}, \overrightarrow{x_j}, k)$. It expresses the improvement provided by $\overrightarrow{x_i}$ in comparison to $\overrightarrow{x_j}$ in the selected function. The gain varies from -1 to 1. A negative value indicates that the first solution is worse than the second one according to f_k. A null gain means that both solutions are equivalent.

$$gain(\overrightarrow{x_i}, \overrightarrow{x_j}, k) = \frac{\lambda(f_k(\overrightarrow{x_i}) - f_k(\overrightarrow{x_j}))}{Max(f_k(\overrightarrow{x_i}), f_k(\overrightarrow{x_j}))} \tag{1}$$

where λ is a coefficient: 1 for a maximization problem; -1 for a minimization.

The fitness of each individual is the sum of gains (for all objectives) compared to other solutions in the same front (see Equation 2).

$$fitness(\overrightarrow{x_i}) = \sum_{j=1}^{|PF|} \sum_{k=1}^{m} gain(\overrightarrow{x_i}, \overrightarrow{x_j}, k) \tag{2}$$

where $|PF|$ is the Pareto front size and m the number of objectives functions.

3.3 Selection operator

To ensure both exploration and exploitation, a kind of fitness is associated to each Pareto front. This is actually a selection probability of the front. Indeed, GAME selects individuals in two steps. The first one consist in selecting a Pareto front within the five built. Then, a solution is chosen from those belonging to the front using a binary tournament selection.

The selection of Pareto fronts is based on a biased roulette wheel where the part of each front depends on both its rank and its cardinality. The probability of selection associated with each front is given by equation 3.

$$P(PF_i) = \frac{\delta(PF_i) * |PF_i|}{\sum_{i=1}^{n}[\delta(PF_i) * |PF_i|]} \tag{3}$$

where:
$P(PF_i)$ is the probability of selection associated with the i^{th} Pareto front.
$|PF_i|$ is the cardinality of PF_i.
n is the total number of Pareto fronts. $\delta(PF_i)$ indicates a priority level associated with each front (see Equation 4).

$$\delta(PF_i) = a(n - i) + b \tag{4}$$

The coefficients a and b must be carefully chosen to preserve diversity within the sets of solutions (successive populations and archive), while improving individuals over generations. These parameters may differ for recombination and replacement. Therefore, the coefficients associated with recombination selection are named a and b, whereas those used in replacement selection are a' and b' (see Section 4.1).

Using both $\delta(PF_i)$ and the cardinality ensures the selection of the best solutions, prevents the dominated solutions from having very low values of fitness and preserves the diversity of the successive populations.

The utility and the efficiency of the proposed ranking mode and selection process were experimentally validated. This experimentation first studied the tuning of a, b, a' and b' for a given number of Pareto fronts. Secondly, a preliminary test phase focused on the sensitivity of GAME to the used number of Pareto Fronts, for the tuned values of $\delta(PF_i)$.

It is worth noting that besides its specific characteristics, presented in this section, GAME uses a parallel evaluation process based on a master-slave model [4]. The parallelization allows GAME to tackle problems with complex objective functions or with objective functions requiring simulation to assess their values. A variant of GAME, named ESBEA, has been applied, for instance, to a constrained optimization problem containing four objective functions evaluated by simulation, in the field of ad hoc mobile communication networks [1].

In this paper, the performance of GAME was assessed by comparison with previous results in literature. The following sections describe experimental conditions and summarize the main results.

4 Validation and experimental results

The experiments described in the following subsections aim at:

- tuning the probabilities associated with Pareto sets;
- validating the interest to have multiple Pareto fronts;

– assessing the performance of GAME compared with other methods presented in literature.

All experiments were performed on the optimization problems proposed in the CEC 2009 [17] competition, more particularly bi-objective constrained problems, according to the experimental conditions defined in this competition. These problems are detailed in [17].

GAME has been implemented using jMetal [7], a platform for the development, experimentation and study of metaheuristics. A population size equals to 100 and 300 generations were used to perform the 30,000 evaluations required for each execution in the CEC 2009 competition. Crossover rate and mutation rate were respectively equal to values recommended in literature: 0.8 and $\frac{1}{Number\ of\ variables}$ (in order to limit the number of parameters in the parameter study and focus on those involved in the proposed mechanisms).

4.1 Tuning GAME's parameters

A parametric study based on the design of experiments (DOEs) has been performed to tune the priority assigned to each Pareto front built by GAME. DOEs permit to get a lot of information on parameters [1] and study their influence, while carrying out a minimum of tests [3].

The whole campain of tests has been based on JMP (9 Pro version) [2], a statistical software for expert data analysis, to create DOE and to analyze the provided results. This software has been used to create some test tables, in which each line corresponds to a given combination of the factors (in this case, a, b, a' and b' presented in Section 3.3) defining a given test. For each of these tests, two indicators (*IGD* and *Maximum Spread* [5]) have been used to measure the quality of the studied combination of parameters [3].

This parameter tuning was performed with a number of Pareto fronts set at 5, by varying a, b, a' and b' between 0 and 10 (integer values). JMP has generated a partial DOE table consisting of 44 tests to be performed. The benchmark used for these tests was Constrained Problem 1.

Figure 1 shows the results returned by JMP. For IGD and *Maximum Spread* indicators, the first four curves show the variations of these indicators based on the values taken by each factor. The last column gives a desirability function (for each indicator) with values between 0 and 1. A desirability value of 1 indicates that the used parameter values enable to get the optimal level for the considered indicator. In this parameter tuning, the desirability according to the IGD decreases when the IGD reaches high values. That is the inverse of the *Maximum Spread*. The last row of curves shows the levels of desirability with respect to both the IGD and *Maximum Spread* indicators.

Based on these results, the best value of desirability is achieved when $a = 6$, $b = 5$, $a' = 5$ and $b' = 4$.

[1] In DOEs these parameters are called factors

[2] http://www.jmp.com

[3] The measured values are also called responses.

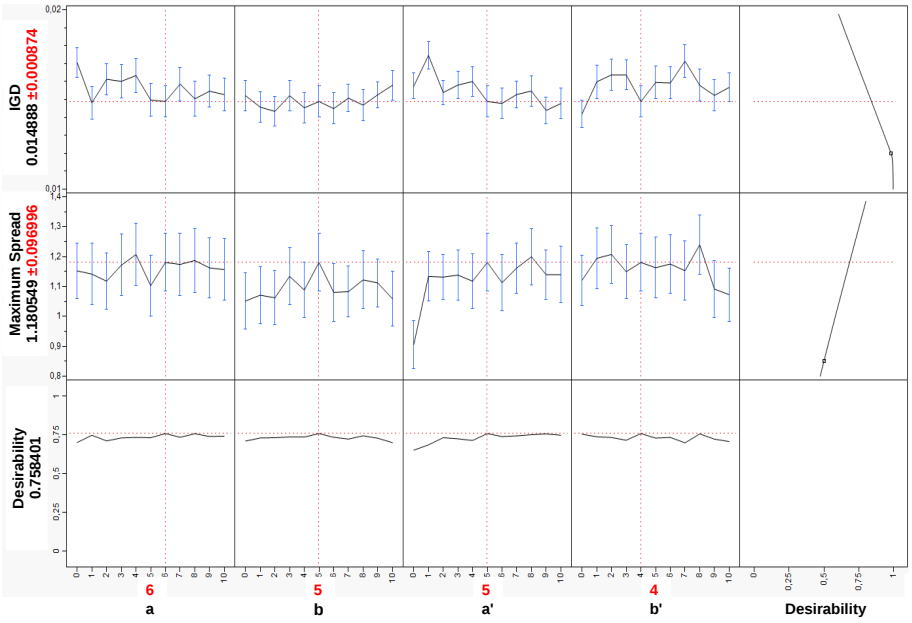

Fig. 1. Tuning the selection coefficients

4.2 Influence of multiple Pareto fronts.

To measure the benefits of multiple Pareto fronts, four versions of GAME are compared, using the values of $\delta(PF_i)$ tuned with JMP. The first version builds two Pareto fronts (like classical algorithms). This version will serve as a reference. The other three versions build 3, 4 and 5 fronts. These values were chosen to study a steady increase of the number of fronts while limiting the duration of experimentations. They are compared to the "2 front" version in the constrained problem 1 of the CEC 2009 competition. The comparison focuses on convergence (IGD) and diversity (*Maximum Spread*) indicators.

Since the generation of the initial population of GAME does not use specific heuristics, individuals are created randomly. Therefore, the best of them often are of fair quality. This results in a relatively high value of the IGD in the first generations (see Figure 2). Fast enough, solutions that are better suited to the problem are produced (by recombination) and eliminate the worst individuals in the archive. This explains the sharp decrease in the IGD in the first 25 generations. Thereafter, the curves of the IGD continue to decline steadily but with a much lower slope. Beyond this behavior common to all scenarios, these results allow us to observe that the increase in the number of fronts improves the quality of final solutions in terms of proximity to the optimum.

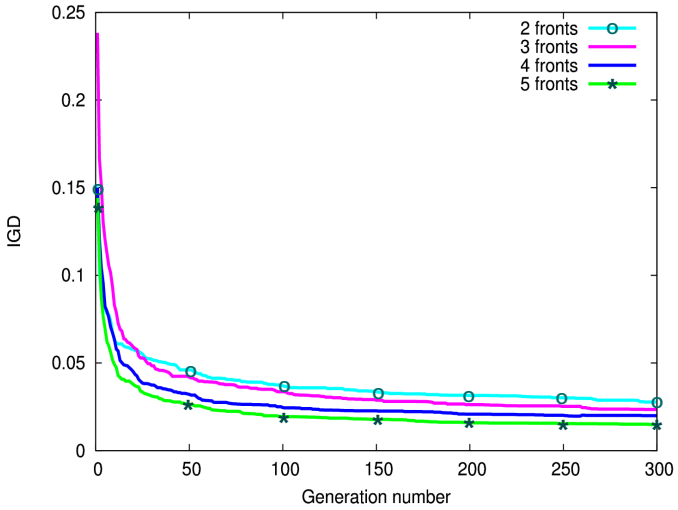

Fig. 2. IGD

During the first few generations, the *Maximum Spread* has relatively high values (see Figure 3). This can be explained by the fact that the archive contains solutions that do not meet constraints. Indeed, the inclusion of such solutions stretches the front beyond the boundaries that are set by the constraints of the problem. The archive built by GAME is then refined, when solutions which respect the constraints and have good values of objective functions are found. This explains the drop in the *Maximum Spread* in the early generations. After the fall phase, the *Maximum Spread* is growing slightly. In addition, the slight variations in the curves of *Maximum Spread* over generations indicates that the archive is updated regularly. New non-dominated solutions are continually being discovered. If initially all scenarios have the same behavior, one can observe that after a few dozen generations, the impact of multiple fronts becomes substantial.

4.3 Performance assessment

The performance of GAME was assessed as if it had taken part in the CEC 2009 competition, in the "Constrained problems" category. This category includes bi-objective problems where the two objectives are to be minimized. Each problem includes 10 real-valued decision variables. GAME was compared to the three algorithms which got the best results in this category, using the same experimental conditions and the same performance criterion:

– for each constrained problem, 30 independent executions limited to 30,000 evaluations must be performed for each algorithm;

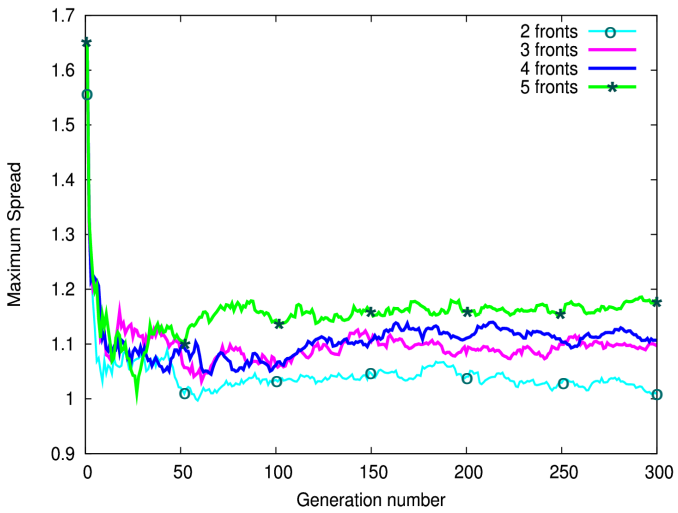

Fig. 3. Maximum spread

– then the mean value of IGD indicator (computed in the sets of final solutions for the 30 independent executions) must be used as comparison criterion. The IGD is to be minimized.

Table 1 provides, for the exhaustive list of bi-objective problems of the competition, the average of the IGD values obtained by the three compared algorithms: DMOEA-DD [11], LiuLi [10] and MTS [16]. These results permitted these algorithms to respectively reach the first rank, the second rank and the third rank during the CEC 2009 competition.

DMOEAD-DD (Dynamical Multiobjective Evolutionary Algorithm - Domain Decomposition) improves DMOEA [19] that used an aggregated *fitness* function including the notion of Pareto dominance, entropy and density (based on *crowding distance*). In the improved variant, authors split the search space into several subsets. DMOEA computes Pareto fronts for each of them. Genetic operators permit information exchange between these subsets.

LiuLi (concatenation of the authors'name : Hai-Lin Liu and Xueqiang Li) splits the search space into sub-areas in order to reduce algorithm complexity. Genetic operations, particularly reproduction, are performed in a single sub-area. Information exchange between areas is based on children, because they may be assigned to other areas.

MTS (Multiple Trajectory Search) is an algorithm based on three local search methods. For each solution, MTS determines the method which corresponds to its neighborhood. This algorithm begins with a large search. The size of neighborhood is progressively reduced until it reaches a given minimal size. Then its size is set to its initial value and the regression re-starts.

Problem	GAME	DMOEADD	LiuLi	MTS
Constrained Problem 1	0.01489	0.01131	0.00085	0.01918
Constrained Problem 2	0.00042	0.0021	0.004203	0.02677
Constrained Problem 3	0.03462	0.056305	0.182905	0.10446
Constrained Problem 4	0.00742	0.00699	0.014232	0.01109
Constrained Problem 5	0.01227	0.01577	0.10973	0.02077
Constrained Problem 6	0.00181	0.01502	0.013948	0.01616
Constrained Problem 7	0.00545	0.01905	0.10446	0.02469

Table 1. Mean values of IGD returned by GAME and the algorithms which participated to CEC2009 competition

Besides, GAME has been tested in the same conditions and the results it provided were added in table 1. This study shows that, globally, GAME reaches solutions which are quite close to the reference front. This results in low IGD values in table 1. Moreover, in a large majority of cases, GAME provides better results than those returned by the three best algorithms in the competition (in the "constrained problems" category).

5 Conclusion

To sum up, GAME is an elitist multi-objective genetic algorithm, based on the building of multiple Pareto fronts. This ranking strategy and the associated 2-step selection provided gains both in terms of proximity with optimal solutions and in terms of diversity in the set of final solutions returned by GAME. Finally, using the experimental conditions of the CEC 2009 competition showed that it would have been quite well ranked in this competition, which constitutes a promising result. In addition, GAME uses a parallel evaluation procedure based on a master-slave model. Although this model allowed the reduction in terms of the duration of the experimentations in previous work in the field of mobile networks [1]. Nevertheless, designing an asynchronous version of GAME would be very interesting. Such a version would no longer rely on an architecture where the master must wait until all the slaves have finished their task before building the next generation. Two other prospects are planned for this work. First, GAME should be applied in various real-life fields to test its sensitivity to the tackled problems more widely. This would permit to prove the genericity of the results provided in this paper. Besides, the authors are working on an adaptive version of GAME. This aims at making it more robust with regards to problem characteristics, rather than using statistically tuned parameters without varying the algorithm behavior during execution.

References

1. Abdou, W., Henriet, A., Bloch, C., Dhoutaut, D., Charlet, D., Spies, F.: Using an evolutionary algorithm to optimize the broadcasting methods in mobile ad hoc networks. Journal of Network and Computer Applications 34(6), 1794 – 1804 (2011)

2. Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M.: Experimental Methods for the Analysis of Optimization Algorithms. Springer, Germany (2010)
3. Box, G.E.P., Hunter, J.S., Hunter, W.G.: Statistics for experimenters. Design, innovation, and discovery. 2nd ed. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons. xvii, 633 p. (2005)
4. Cantú-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Genetic Algorithms and Evolutionary Computation, Kluwer Academic Publishers (2000)
5. Deb, K.: Multi-objective Optimization Using Evolutionary Computation. John Wiley & Sons (2003)
6. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. Evol. Computation 6(2), 182–197 (2002)
7. Durillo, J.J., Nebro, A.J.: jmetal: A java framework for multi-objective optimization. Advances in Engineering Software 42(10), 760 – 771 (2011)
8. Eskandari, H., Geiger, C.D., Lamont, G.: FastPGA: A dynamic population sizing approach for solving expensive multiobjective optimization problems. In: Evolutionary Multi-Criterion Optimization. 4th Int. Conf., EMO 2007. Lecture Notes in Computer Science, vol. 4403, pp. 141–155. Springer (2007)
9. Hippolyte, J.L., Bloch, C., Chatonnay, P., Espanet, C., Chamagne, D., Wimmer, G.: Tuning an evolutionary algorithm with taguchi methods. In: CSTST'08, 5th Int. Conf. on Soft Computing as Transdisciplinary Science and Technology. pp. 265–272. ACM Press (2008)
10. Liu, H.L., Li, X.: The multiobjective evolutionary algorithm based on determined weight and sub-regional search. In: Proc. of 11th conf. on Congress on Evolutionary Computation. pp. 1928–1934. CEC'09, Trondheim, Norway (2009)
11. Liu, M., Zou, X., Chen, Y., Wu, Z.: Performance assessment of dmoea-dd with cec 2009 moea competition test instances. In: IEEE Congress on Evolutionary Computation. pp. 2913–2918 (2009)
12. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evol. Comput. 4, 1–32 (March 1996)
13. Silverman, B.W.: Density estimation: for statistics and data analysis. Chapman and Hall, London (1986)
14. Srinivas, M., Patnaik, L.M.: Genetic algorithms: A survey. IEEE Computer 27(6), 17–26 (1994)
15. Surry, P., Radcliffe, N., Boyd, I.: A multi-objective approach to constrained optimisation of gas supply networks: The comoga method. In: Fogarty, T. (ed.) Evolutionary Computing, Lecture Notes in Computer Science, vol. 993, pp. 166–180. Springer Berlin / Heidelberg (1995)
16. Tseng, L.Y., Chen, C.: Multiple trajectory search for unconstrained/constrained multi-objective optimization. In: Proc. of 11th conf. on Congress on Evolutionary Computation. pp. 1951–1958. CEC'09, IEEE Press, Trondheim, Norway (2009)
17. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition. Tech. rep., School of Computer Science and Electronic Engineering (2009)
18. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. Tech. rep., Computer Engineering and Networks Laboratory (TIK),Swiss Federal Institute of Technology (ETH) Zurich (2001)
19. Zou, X., Chen, Y., Liu, M., Kang, L.: A new evolutionary algorithm for solving many-objective optimization problems. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society 38(5), 1402–1412 (Oct 2008)