Using a gradient based method to seed an EMO algorithm

Alfredo G. Hernandez-Diaz[a], Carlos A. Coello[⋆b], Fatima Perez[c], Rafael Caballero[⋆⋆c], and Julian Molina[c]

[a] Department of Quantitative Methods, Pablo de Olavide University, Seville (Spain)
[b] Centro de Investigacion y de Estudios Avanzados, Mexico D.F., Mexico
[c] Department of Applied Economics (Mathematics), University of Malaga, Malaga, Spain

Abstract. In the field of single-objective optimization, hybrid variants of gradient based methods and evolutionary algorithms have been shown to performance better than the pure evolutionary method. This same idea has been used with Evolutionary Multiobjective Optimization (**EMO**), obtaining also very promising results. In most of the cases, gradient information is used as part of the mutation operator, in order to move every generated point to the exact Pareto front. This means that gradient information is used along the whole process, and then consumes computational resources also along the whole process. On the other hand, in our approach we will use gradient information only at the beginning of the process, and will show that quality of the results is not decreased while computational cost is. We will use a steepest descent method to generate some efficient points to be used to seed an EMO method. The main goal will be generating some efficient points in the exact front using the less evaluations as possible, and let the EMO method use these points to spread along the whole Pareto front. In our approach, we will solve box-constrained continuous problems, gradients will be approximated using quadratic regressions, and NSGA-II [3] and a method based on Rough Sets theory [14] will be used.

Key words: gradient based method, multi-objective programming, Evolutionary Multi-Objective Optimization, quadratic approximation, Rough Sets

1 Introduction

EMO have shown great success on many complex problems, although some weak points can be identified within this type of methods: a lot of function evaluations are required to ensure convergency to the exact Pareto front. EMO methods are stochastic algorithms, and a small number of samplings in the decision space are not enough to ensure convergency.

⋆ The second author acknowledges support from CONACYT project no. 45683-Y.
⋆⋆ Fourth and fifth authors acknowledges support from Andalusian Regional Government and Spanish Ministry of Educacion y Ciencia

On the other hand, the classical (exact) methods for (multi-objective) optimization (gradient based methods) consume just a few number of evaluations, but can be trapped in local optima and require a lot of assumptions about the problem: continuity, differentiability, explicit mathematical formulation, etc.

Also, it is well known that, under proper assumptions, Newton's method is quadratically convergent, but its efficiency is reduced by its expensive computational cost, especially, for the middle-large scale problems. The key point is to evaluate the gradient and the Hessian efficiently, and two different approaches can be found:

- **Use analytical derivatives** The first option is manually obtaining analytic derivatives of each function and evaluate them. But this is only possible if an explicit mathematical formulation is available (although it is likely to result in the most exact methods), and this is the main weakness of this approach as many interesting problems could not be solve: simulation based problems, design problems, etc. On the other hand, it is an error-prone activity, because if the formulation is complicated, obtaining analytical derivatives can be a hard task.
- **Use estimated derivatives** In this category we can find the Newton-like methods, where derivatives are estimated in some efficient way. These methods don't require explicit formulae of derivatives but, on the other hand, consume some more evaluations in order to compute the estimation.

As one of the main strengths of an EMO methods is that an explicit mathematical formulation is not required, our goal in this work will be **use estimated derivatives** but consuming the less evaluations as possible (using them only at the beginning) while maintaining a high quality on the results. On the other hand, instead of using it along the whole process (consuming too many evaluations) we will just use it at the beginning to seed the EMO method. This way, the main role of this gradient based method will be driving the EMO method directly to the exact Pareto front and then let it spread along the rest of the Pareto front.

2 Related work

Some attempts have been done in the last years to get benefits of both approaches (classical and evolutionary) through hybrid methods. The main idea is use the EMO method to guide the search to the right region (global search) and use gradient information to find the accurate optimum quickly using its fast convergency (local search).

In [1], on each generation, for several randomly selected solutions in the population, they convert the MOP problem into a single-objective problem through the use of the ε-Constraint method (see for example [4]) and solve it with a **Newton-like** method, the Sequential Quadratic Programming (SQP) method, in order to improve this solution. They obtain very good results in quality, but consume quite a lot of evaluations in some cases.

In [6] they use a multilevel subdivision technique that subdivides the search space, and perform local search in each subspace. This local search is based on a similar derivation of a single descent direction used in [5]. Again, **exact derivatives** are used, and some problems can be found if the objectives have different ranges, because the largest direction of simultanous descent will be biased towards the objective with the largest range.

In [7], they analytically describe the complete set of non-dominated simultaneously improving directions using the **exact gradient** of each objective functions, and this set is consider as a multi-objective gradient. In order to use this information, at the end of a generation a set of candidate solutions is determined. The gradient-based local search operator is then applied with each of these candidate solutions as a starting point. Its performance, although so good with 2-objective problems, is not so good on problems with more than 2 objectives, as explained in the paper. On the other hand, they find problems when moving a solution in the boundary of the feasible region, and the number of evaluations consumed is also high.

In [8], they use **exact derivatives**, and try to answer a key question: what is the best way to integrate the use of gradient techniques in the cycle of an EMO method? They propose an adaptive resource-allocation scheme that uses three gradient techniques: a conjugate gradients algorithm is applied to a randomly chosen objective, an alternating-objective repeated line-search and a combined-objectives repeated line-search. During optimization, the effectivity of the gradient techniques is monitored and the available computational resources are redistributed to allow the (currently) most effective operator to spend the most resources. Results quality is so high, but again quite a lot of evaluations are consumed and exact derivatives formulae is required.

In [9], two methods for unconstrained multi-optimization problems are used as a mutation operator in a state-of-the-art EMO algorithm. These operators require gradient information which is **estimated** using finite difference method and using a stochastic perturbation technique requiring few function evaluations. Results are so promising, but still the number of evaluations is high as the gradient based operator is used along the whole process.

In [10], they design a population-based **estimation** of the multi-objective gradient, although a complete algorithm is not described in this paper. Also, no experimentation is provided, because their aim is to give an indication of the power of using directional information.

In [11], the Multiobjective Steepest Descent Method (MSDM) define the degree of improvement in each objective function when a solution is moved in a direction as the inner product of the direction and the steepest descent direction (using **exact derivatives**) of respective objective function. MSDM finds the direction that maximizes the minimum degree of improvement of all objective functions by solving a quadratic programming problem and moves the solution in that direction. When a solution is on a feasible region boundary, it incorporates the boundary information into the quadratic programming problem

to exclude infeasible directions. MSDM is computationally expensive since a quadratic programming problem has to be solved to find a single direction.

3 Definitions and basic concepts

We consider multiobjective optimization problems (MOP) of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(x), f_2(x), \ldots, f_p(x)\} \\ \text{subject to} & x \in X \subseteq \mathbb{R}^n, \end{array} \tag{1}$$

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, for $x \in \mathbb{R}^n$, a direction $v \in \mathbb{R}^n$ is a *descent direction* if:

$$\nabla f(x)^t v < 0 \tag{2}$$

A generalized gradient method can be summarized in the following equation:

$$x^{k+1} = x^k + \alpha^k v^k$$

where v^k is a descent direction and α^k is the step size. One of the most commonly used choice for the descent direction is the following (*steepest descent*):

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k)$$

Choosing the optimum step size α^k is desirable, but it may be computationally expensive. Hence, some other set of rules, which have good properties, e.g., convergence, are more efficient. One of the most efficient is the Armijo's rule:

- Let $\beta \in (0, 1)$ be a prespecified value, let v be a descent direction and let x be the current point. The condition to accept t (the step size) is:

$$f(x + tv) \leq f(x) + \beta t \nabla f(x)^t v$$

where we start with $t = 1$ and while this condition is not satisfied we set $t := t/2$. The choice of β can be critical, as the bigger the value of β, the bigger the steps we can implement at the beginning. But, the bigger the value of β, the more evaluations that can be consumed if too many reductions of t must be done to achieve the condition.

4 Gradient Based method for Multi-Objective Optimization

The goal now is trying to adapt some of the principles of single-objective optimization to obtain a number of efficient points of the MOP problem. The main idea is based on the Fritz-John's optimality condition for MOP problems (see for example [11]).

– Given a point $x \in X$, a necessary condition to be Pareto optimal solution is the existence of $\boldsymbol{\lambda} \geq 0$ such that:

$$\sum_{i=1}^{p} \lambda_i \nabla f_i(x) = 0$$

For a bi-objective optimization problem, this condition means that for any Pareto optimal solution, we can find some $\lambda \geq 0$ such that $\nabla f_1(x) = -\lambda \nabla f_2(x)$. This is, for any Pareto optimal point, gradients of both objective functions are parallel but in the opposite direction. It means that if we are placed in the minimum of one of the objectives (for example the minimum of f_1, a Pareto optimal solution) and follow the direction of $\nabla f_2(x)$, we will keep in the Pareto front. This is shown graphically in Figure 1.

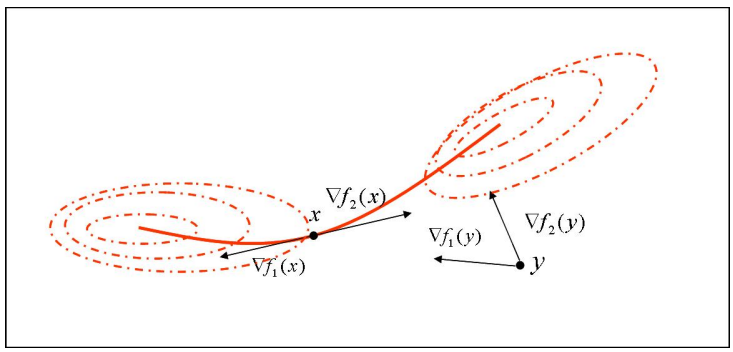

Fig. 1. Pareto front on a bi-objective problem

This idea was used in [12], where they link $p+1$ local searches (more precisely, tabu searches). The first local search starts from an arbitrary point and attempts to find the optimal solution to the problem with the single objective f_1. Let x_1 be the last point visited at the end of this search. Then, a local search is applied again to find the best solution to the problem with the single objective f_2 using x_1 as the initial solution. This process is repeated until all the single-objective problems associated with the p objectives have been solved. At this point, they solve again the problem with the first objective f_1 starting from x_p, to finish a cycle around the efficient set. This phase yields the p efficient points that approximate the best solutions to the single-objective problems that result from ignoring all but one objective function, and additional efficient solutions may be found during this phase because all visited points are checked for inclusion in the approximation of the Pareto front, as probably most of the intermediate points will lie on the Pareto front. This way, they obtain an initial set of efficient points to be used as an initial population for the EMO method developed in [12].

In this work, we are going to use the same idea, link $p+1$ single objective local searches, but using a single-objective gradient based method instead of a tabu

search. Next subsection is devoted to show the main features on this gradient based local search.

4.1 Single-Objective Gradient based method

As local search, we are going to use an steepest descent method, this is, given the current point x_k, the next point will be computed as follows:

$$x^{k+1} = x^k - t \cdot \widetilde{\nabla} f(x^k)$$

where $\widetilde{\nabla} f(x^k)$ is an estimation of $\nabla f(x^k)$, and the step length (t) will be computed following an Armijo's rule with $\beta = 0.1$ and starting with the value of $t = 1$. The reason to choose a low value for β is the fact that small steps are also interesting for us while we are on the Pareto front, as we are checking every intermediate solution for being included in the final approximation. This is, we are not only interested in the final point of each search, but also in the intermediate points. To estimate the gradient of a function f, we will use a quadratic approximation:

$$f(x) \approx \beta_0 + \sum_{i=1}^{n} \beta_i^1 \cdot x_i + \sum_{i=1}^{n} \sum_{j=i}^{n} \beta_{i,j}^2 \cdot x_i \cdot x_j$$

The number of parameters (N) to adjust such an approximation for a function with n variables is: $N = 1 + n + \frac{n(n+1)}{2} = \frac{n^2+3n+2}{2}$. N represents the minimum number of points needed to adjust such an approximation. For a problem with 30 variables, for example, at least 496 will be needed. In order to generate these N points efficiently, we used Latin-Hypercubes [13], which is a method that guarantees a good distribution of the initial population in a multidimensional space, as it is required in order to better fit the function with this quadratic approximation. A Latin cube is a selection of one point from each row and column of a square matrix representing different ranges of each variable. This way, we obtain a set of points, where, in each variable, there is exactly one point per column or range of values. Once these points are generated and evaluated, we compute the values of each parameter solving the corresponding system of equations using a pseudo-inverse (due to its complexity when N is increased). This system of equations can be formulated using matrices: $X \cdot B = Y$, where:

$$X = \left(\begin{array}{c|c|c} 1 & (x_i^1) & (x_i^1 \cdot x_j^1) \\ 1 & (x_i^2) & (x_i^2 \cdot x_j^2) \\ \vdots & \vdots & \vdots \\ 1 & (x_i^N) & (x_i^N \cdot x_j^N) \end{array}\right) B = \begin{pmatrix} \beta_0 \\ \beta_i^1 \\ \vdots \\ \beta_{i,j}^2 \end{pmatrix} Y = \begin{pmatrix} f(\boldsymbol{x_1}) \\ f(\boldsymbol{x_2}) \\ \vdots \\ f(\boldsymbol{x_N}) \end{pmatrix}$$

Finally, we assumed the following stopping conditions:

1. The step is too small: $t \cdot \|\nabla f(x_k)\| < 0.01$, or
2. The improvement is too small: $|f(x_{k+1}) - f(x_k)| < 0.001$

The complete method is summarized in Algorithm 1.

Algorithm 1 Multi-Objective Gradient Based method: MGBM

```
Generate a set InitPop with N initial points using Latin-Hypercubes.
Send each point in InitPop to the list of effic. sol: PF
Use the set InitPop to adjust a quadratic approximation of each objective function.
for each solution in PF do
  for each objective function f_i (repeating the first one) do
    x_0 =last point visited or efficient solution
    while stopping conditions = FALSE do
      Obtain x_{k+1} through the single-objective gradient based method using objective f_i
      Send x_{k+1} to PF.
    end while
  end for
end for
```

5 Hybridization and Preliminary Results

In order to show some preliminary results, we have used this Multi-Objective Gradient Based method (MGBM) to seed an EMO method based on Rough Set Theory. This EMO method was used in [14] in cooperation with a Differential Evolution method and showed some interesting properties to be hybridized: if some (close to the real) efficient solutions are provided, this Rough Sets method is able to spread along the whole front using few evaluations.

Rough Sets theory is a new mathematical approach to imperfect knowledge. The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians and mathematicians. Recently, it also became a crucial issue for computer scientists, particularly in the area of artificial intelligence (AI). Rough sets theory was proposed by Pawlak [16], and presents another attempt to this problem. Rough sets theory has been used by many researchers and practitioners all over the world and has been adopted in many interesting applications. The rough sets approach seems to be of fundamental importance to AI and cognitive sciences, especially in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, inductive reasoning and pattern recognition. Basic ideas of rough set theory and its extensions, as well as many interesting applications, can be found in books (see [17]), special issues of journals (see [18]), proceedings of international conferences, and in the internet (see `www.roughsets.org`).

For MOP problems, this approach tries to approximate the Pareto front using a Rough Sets grid. To do this, they use an initial approximation of the Pareto front (provided by any other method) and implement a grid in order to get more information about the front that will let it improve this initial approximation. To create this grid, as an input it requires M feasible points divided in two sets: the nondominated points (ES) and the dominated ones (DS). Using these two sets a grid is created to describe the set ES in order to intensify the search on it. But it describes the Pareto front in decision variable space and then this information can be easily used to generate more efficient points and then improve

this initial approximation. In our case, this initial sets, the nondominated points (ES) and the dominated ones (DS), will be provided by the MGBM. To test the performance of the MGBM and the MGBM-RS method we used two test problems from the **ZDT** set [15]: ZDT1 and ZDT2. We first run the MGBM method and let the RS phase complete the approximation till 2000 evaluations are consumed. In Figure 2, we show the initial approximation (MGBM) as well as the final results (MGBM+RS).

For these problems, the MGBM is able to find 32 exact efficient points for the ZDT1 problem and 36 exact efficient points for the ZDT2, using around 750 evaluations. We must note that close to 500 of them are consumed by the Latin-Hypercubes, and then the proper gradient based method is consuming around 250 evaluations. This initial set of efficient solutions lets the second phase (the RS phase) complete a wide and well distributed approximation of the whole Pareto front within 2000 evaluations, being then so competitive for this kind of problems.

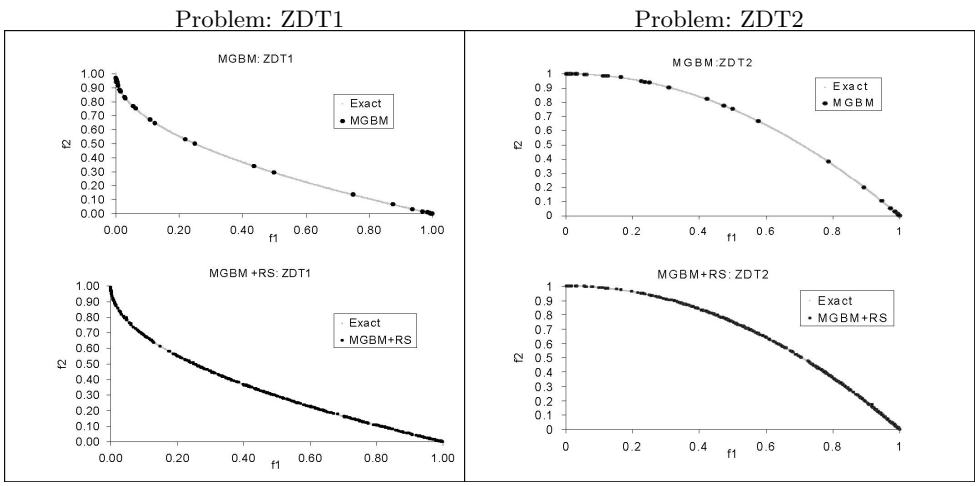

Fig. 2. MGBM and MGBM+RS results for ZDT1 and ZDT2

On the other hand, we have used MGBM to seed the well-known NSGA-II [3], which is a MOEA representative of the state-of-the-art in the area. The seeding procedure is consuming about 1000 evaluations while the NSGA-II is consuming another 1000 evaluations. In order to allow a fair comparison of results, the seeded NSGA-II is compared with NSGA-II with a random initial population and consuming 2000 evaluations. It can be observed in Table 1 that the seeded NSGA-II produced the best values in most cases. We used three standard measures in the literature to compare the performance of both methods: SSC [19] (to be maximized), Unary additive epsilon indicator ($I^1_{\varepsilon+}$) [20] (to be minimized) and Spread (Δ)([2])(to be minimized). Regarding SSC and the unary additive epsilon

indicator, the seeded procedure outperformed NSGA-II in all the cases. Relating the Spread measure, the random NSGA-II outperformed our approach only in two cases. This is certainly remarkable if we consider the fact that the seeding procedure is only focused in convergence aspects. Thus, it was expected that the random NSGA-II would be favored by this performance measure.

Function	**A**lgorithm	**SSC**	$I^1_{\varepsilon+}$	Δ
ZDT1	Newton+NSGA2	**0.9203**	**0.0233**	**0.4571**
ZDT1	NSGA2-2000	0.7604	0.1780	0.8093
ZDT2	Newton+NSGA2	**0.8870**	**0.0104**	**0.4074**
ZDT2	NSGA2-2000	0.6765	0.2727	0.9246
ZDT3	Newton+NSGA2	**0.6849**	**0.1769**	0.7954
ZDT3	NSGA2-2000	0.6752	0.1817	**0.7848**
ZDT4	Newton+NSGA2	**0.9562**	**0.0448**	0.9972
ZDT4	NSGA2-2000	0.9075	0.0915	**0.9291**
ZDT6	Newton+NSGA2	**0.9215**	**0.0291**	1.0198
ZDT6	NSGA2-2000	0.4281	0.4831	**0.9523**

Table 1. Comparison of results for the five test problems.

6 Conclusions

In this paper, a Multi-Objective Gradient Based Method to generate some efficient points is proposed. The main aim is consuming the less evaluations as possible and use these solutions to seed and EMO method. For this reason, gradient information is used only as a seeding procedure and it is not invoked through all the resolution, as usually it is done in the literature. With this preliminary results we show how the use of gradient information only at the beginning of the resolution process could reduce the computational cost while quality is not decreased. This is, gradient information could be so useful at the beginning to enhance convergence, but once the EMO method is provided with solutions close (or in) to the Pareto front, the use of gradient information is consuming a lot of evaluations while not providing sensible advantages.

In the future, besides completing a comprehensive set of experiments, we would like to improve the local search, considering a more efficient method such as BFGS, instead of steepest descent.

References

1. Xiaolin Hu, Zhangcan Huang, Zhongfan Wang: Hybridization of the multi-objective Evolutionary algorithms and the gradient-based algorithms. Congress on Evolutionary Computation, 2003. CEC'03. Vol. 2, pp.: 870- 877, (2003).
2. K. Deb, Multi-Objective Optimization using Evolutionary Algorithms. JohnWiley & Sons, Chichester, UK (2001), ISBN 0-471-87339-X.

3. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, (2002).
4. R. E. Steuer: Multiple Criteria Optimization: Theory, Computation, and Application. John Wiley, New York, (1986).
5. S. Schaffler, R. Schultz, and K. Weinzierl: Stochastic method for the solution of unconstrained vector optimization problems. Journal of Optimization Theory and Applications, 114(1), pp.: 209-222, (2002).
6. M. Dellnitz, O. Schütze, and T. Hestermeyer: Covering pareto sets by multilevel subdivision techniques. J. of Optimization Theory and Applications, 124(11), pp:. 13-136, (2005).
7. Peter Bosman, Edwin de Jong: Exploiting Gradient Information in Numerical Multi-Objective Evolutionary Optimization. Proceedings of the 7th annual Conference on Genetic and Evolutionary Computation (GECCO'05), pp.: 755 - 762. ACM Press. (2005)
8. Peter Bosman, Edwin de Jong: Combining gradient techniques for numerical multi-objective evolutionary optimization. Proceedings of the 8th annual Conference on Genetic and Evolutionary Computation (GECCO'06), pp.: 627 - 634. ACM Press. (2006)
9. Pradyumn K. Shukla: On Gradient Based Local Search Methods in Unconstrained Evolutionary Multi-objective Optimization. Proceedings of EMO 2007, LNCS 4403, pp.: 96-110, (2007).
10. Martin Brown and Robert E. Smith: Effective Use of Directional Information in Multi-objective Evolutionary Computation. Proceedings of GECCO 2003, LNCS 2723, pp.: 778-789, (2003).
11. Fliege, J., Svaiter, B.: Steepest Descent Methods for Multicriteria Optimization. Mathematical Methods of Operations Research, 51(3), pp.: 479-494, (2000).
12. Molina, J., Laguna, M., Marti, R. and Caballero, R.: SSPMO: A Scatter Tabu Search Procedure for Non-Linear Multiobjective Optimization.INFORMS Journal on Computing, Vol. 19 (1), pp.: 91-100, (2007).
13. M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2), pp.: 239-245, (1979).
14. Hernandez-Diaz A., Santana-Quintero L., Coello C. ,Caballero R. and Molina J.: A new proposal for multi-objective optimization using differential evolution and Rough Sets Theory. In Thomas Philip Runarson et alt.(editors) Parallel Problem Solving from Nature (PPSN IX) 9th Interantional Conference, pp.: 483-492, (2006).
15. Zitzler E., Deb K., Thiele L.:Comparison of Multiobjective Evolutionary Algorithms:Empirical Results.Evolutionary Computation, Vol. 8(2), pp.: 173-195, (2000).
16. Z. Pawlak: Rough sets. International Journal of Computer and Information Sciences, 11(1), pp.: 341-356, (1982).
17. Z. Pawlak: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, The Netherlands, (1991).
18. T.Y. Lin: Special issue on rough sets. Journal of the Intelligent Automation and Soft Computing, Vol. 2(2), (1996).
19. E. Zitzler and L. Thiele: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE Trans. on Evolutionary Computation Vol. 3(4), pp.: 257–271, (1999).
20. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca: Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation Vol. 7(2), pp.: 117–132, (2003).