

# Implementation of the GISMOO algorithm in the ParadisEO framework

Florian Mazière<sup>1</sup>, Caroline Gagné<sup>1</sup>, Pierre Delisle<sup>2</sup>, Arnaud Zinflou<sup>3</sup>, and Michaël Krajecki<sup>2</sup>

<sup>1</sup> Université du Québec à Chicoutimi, 555 boul. de l'Université, Chicoutimi, Québec, Canada, G7H 2B1  
[florian.maziere1][caroline.gagne]@uqac.ca

<sup>2</sup> CReSTIC, Université de Reims Champagne-Ardenne, Reims, 51687, France  
[pierre.delisle][michael.krajecki]@univ-reims.fr

<sup>3</sup> Institut de recherche d'Hydro-Québec - IREQ, 1800, Lionel-Boulet, Varennes, QC, Canada J3X 1S1  
zinflou.arnaud@ireq.ca

## 1 Abstract

Most industrial problems tend to optimize several objectives at once. This kind of problem does not usually have a single optimal solution, but rather a set of best trade-offs. To solve these problems, the resolution algorithms, such as metaheuristics, have to adjust their way of doing things in order to find a number of solutions : the Pareto-optimal set. In some recent work [1] a new evolutionary algorithm which allows solving multiobjective problems was proposed : the GISMOO algorithm. This algorithm looks like a genetic algorithm with an elitist replacement, but its main distinctive feature is that it contains an immune phase, whereby half of the offspring population is generated by methods coming from clonal selection algorithms [2]. This immune phase allow us to obtain a set of solutions which are better distributed on the Pareto front. Moreover, it is important to note that GISMOO, unlike other evolutionary algorithms, requires few parameters to calibrate. Thus, it is easy and fast to readjust this algorithm to different problems. Finally, whether the problems are continuous [3] [1] or combinatorial [4] [5], GISMOO outperforms the reference algorithm NSGA-II [6]. Indeed, the classical metrics used during the set of experiments show a better convergence toward the Pareto front and an ability to find more diversified solutions.

In order to make GISMOO accessible to as many people as possible and to get a version which is easily adaptable to any kind of problem, the use of a framework seems to be the suitable solution. The framework that has been chosen is ParadisEO [7], a white-box object-oriented framework for the design of metaheuristics on different platforms. These metaheuristics can be either based on a single solution or several solutions such as evolutionary algorithms. In addition, this framework contains a module for solving multiobjective problems and another to exploit parallel architectures in order to implement parallel and distributed metaheuristics. In its basic version, ParadisEO proposes the MOGA, NSGA-II and SPEA-II algorithms but not yet the GISMOO algorithm, although the latter was proved more efficient. This is why it is interesting to propose an implementation of GISMOO in Paradiseo, whereby users could quickly optimize their own problems by the GISMOO algorithm. To do this, users only have to define the problem in the framework, especially determining its objective functions. Furthermore, they could use the crossover operator and mutations operators already defined in ParadisEO or create new operators more specific to their problem.

## 2 Numerical experiment and results

To exemplify the efficiency of the GISMOO implementation in ParadisEO, the quadratic assignation problem (MQAP) [8] has been chosen. Schematically, a set of facilities have to be assigned to various locations. For each pair of facilities, there are different kinds of flow, and the flow cost depends on the distance between the two facilities. The aim of this problem is to minimize the total cost of each flow. In this way, each kind of flow corresponds to a specific objective function of the problem. Like its uniojective version, this problem is NP-hard [8] and so requires the use of approximate methods to optimize problem instances with many facilities and locations. Moreover,

the MQAP is a well-known multiobjective problem, and some test instances are freely available [9]. The true Pareto front is available for some test instances which allow us to validate the quality of the solutions obtained by the implemented algorithm. During the set of experiments, GISMOO was directly compared with algorithms already defined in ParadisEO, such as NSGA-II or SPEA-II. Since this is multiobjective optimization, several elements have to be evaluated. Thus, three metrics have been used. The first metric, called the convergence metric  $\Upsilon$  [6], allows us to assess the convergence of a solution set toward the Pareto front or its approximation. The second metric  $\mathcal{C}$  [10] compares the convergence of two sets of solutions obtained. More specifically, it indicates the proportion of solutions dominated by another solution set. The last metric, the diversity metric  $\delta$  [6], allows us to evaluate the spread of solutions contained in a set. The results obtained on twenty test instances confirm that the GISMOO algorithm outperforms the others according to all three metrics computed. Indeed, the first two metrics show a better convergence of the algorithm toward the Pareto front. In addition, according to metric  $\delta$ , the solutions obtained by GISMOO are better distributed than those obtained by competing algorithms. This spread is an important consideration in offering a wide range of choices to the decision maker. More details on the set of experiments and results will be provided in an extended article.

## References

1. A. Zinflou, C. Gagne, and M. Gravel. Solving multiple-objective optimization problems using GISMOO algorithm. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 239–244.
2. L. N. de Castro and F. J. Von Zuben. Learning and optimization using the clonal selection principle. *Evolutionary Computation, IEEE Transactions on*, 6(3):239–251, 2002.
3. A. Zinflou, C. Gagné, and M. Gravel. Gismoo vs genetic and differential evolution algorithms in multiobjective optimization. *Proceedings of the 9th Metaheuristics International Conference (MIC 2011)*, pages pp S1–52–10, 2011.
4. C. Gagné and A. Zinflou. An hybrid algorithm for the industrial car sequencing problem. *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI 2012), P, Brisbane, Australia, 2012*.
5. C. Gagné and A. Zinflou. Solving multi-objective quadratic assignment problems using hybrid genetic/immune strategy. *Proceedings of the 10th edition of the Metaheuristics International Conference (MIC 2013), Singapore, 2013*.
6. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
7. S. Cahon, N. Melab, and E.-G. Talbi. Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics*, 10(3):357–380, 2004.
8. J. Knowles and D. Corne. Instance generators and test suites for the multiobjective quadratic assignment problem. In *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 2003, Proceedings*, number 2632 in LNCS, pages 295–310. Springer, 2003.
9. J. Knowles. MQAP instances. <http://www.cs.man.ac.uk/~jknowles/mQAP/>.
10. E Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications, 1999.