A Study on Dominance-Based Local Search Approaches for Multiobjective Combinatorial Optimization

Arnaud Liefooghe, Salma Mesmoudi, Jérémie Humeau, Laetitia Jourdan, El-Ghazali Talbi

▶ **To cite this version:**

Arnaud Liefooghe, Salma Mesmoudi, Jérémie Humeau, Laetitia Jourdan, El-Ghazali Talbi. A Study on Dominance-Based Local Search Approaches for Multiobjective Combinatorial Optimization. SLS 2009 - 2nd International Workshop on Engineering Stochastic Local Search Algorithms: Designing, Implementing and Analyzing Effective Heuristics, Sep 2009, Brussels, Belgium. pp.120-124, 10.1007/978-3-642-03751-1_11 . hal-00763711

HAL Id: hal-00763711
https://hal.science/hal-00763711v1

Submitted on 3 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Study on Dominance-based Local Search Approaches for Multiobjective Combinatorial Optimization

Arnaud Liefooghe, Salma Mesmoudi, Jérémie Humeau,
Laetitia Jourdan, and El-Ghazali Talbi

Laboratoire d'Informatique Fondamentale de Lille, UMR CNRS 8022,
INRIA Lille-Nord Europe, Université Lille 1, Villeneuve d'Ascq, France
Arnaud.Liefooghe@lifl.fr

Abstract. The purpose of the current paper is twofold. First, a unified view of dominance-based multiobjective local search algorithms is proposed. We focus on methods based on the iterative improvement of the nondominated set by means of a neighborhood operator. Next, the effect of current solutions selection and of neighborhood exploration techniques for such purpose is studied. Experiments are conducted on a permutation flowshop scheduling problem in a two- and a three-objective variant.

1 Introduction

The aim of this study is to provide a unified view of dominance-based local search for multiobjective optimization. Contrary to the single-objective case, a Multiobjective Combinatorial Optimization Problem (MCOP) does not yield a unique optimal solution. Instead, a set of compromise solutions, known as efficient solutions must generally be identified. Since they are naturally well-suited to find multiple efficient solutions in a single simulation run, a tremendous number of multiobjective evolutionary algorithms have been proposed over the last two decades [1]. However, local search methods are known to be efficient metaheuristics for single-objective optimization. Local search, also referred to as hill-climbing, descent, iterative improvement, etc., is likely the oldest and simplest metaheuristic [2]. But multiobjective local search principles based on a dominance relation appeared quite recently [1, 3]. Hence, some dominance-based multiobjective local search methods have been proposed in the literature, including the Pareto Archived Evolution Strategy (PAES) [4], the Pareto Local Search (PLS) [5] or the Bicriteria Local Search (BLS) [6]. Such methods generally combine the definition of a neighborhood structure with the use of a population of solutions. They maintain a set of potentially efficient solutions, and iteratively improves this set by exploring part of its neighborhood. Our first purpose is to give a unified view of dominance-based multiobjective local search. We describe the basic components shared by all these algorithms and we introduce a general-purpose model for their design. Afterwards, we concentrate on a subpart of components involved into the unified model in order to study their respective behavior on a multiobjective permutation flowshop scheduling problem.

2 Dominance-based Multiobjective Local Search

Until now, each DMLS algorithm was designed independently of the others, and was implemented as a self-contained method with its own specific components. In the following, we identify the common components shared by all DMLS algorithms and propose a unifying model that takes them into account. Hence, whatever the MCOP to be solved, the common concepts for the design of a DMLS algorithm can be stated as follows: (1) design a representation, (2) design a initialization strategy, (3) design a way of evaluating a solution, (4) design a suitable neighborhood structure, (5) design a way of evaluating a neighboring solution incrementally (if possible), (6) decide a current set selection strategy, (7) decide a neighborhood exploration strategy, (8) decide an archive management strategy, (9) decide a stopping condition. When dealing with any kind of metaheuristics, one may distinguish problem-related and problem-independent components. Hence, the first five issues presented above strongly depend of the MCOP under consideration, whereas the last four ones can be seen as generic components. In addition, three data structures are used to store (*i*) the archive contents, (*ii*) the *current set* of solutions whose neighborhood is to be explored, and (*iii*) the *candidate set* of neighbor solutions that will potentially enter the archive. Problem-related components are assumed to be designed for the MCOP at hand, so that they are not discussed in the paper due to space limitation.

2.1 Problem-independent Issues

Current Set Selection. The first phase of a local search step deals with the selection of a set of solutions from which the neighborhood will be explored. Generally speaking, in the frame of the DMLS model presented in the paper, two strategies can be applied. Firstly, an *exhaustive selection*, where all solutions from the archive are selected. Second, a *partial selection*, where only a subset of solutions is selected. Such a set may be selected at random, or also with respect to a diversity measure. Of course, if some archive members are marked as visited, they must be discarded of the current set selection for obvious efficiency reasons.

Neighborhood Exploration. From the current set, a number of candidate solutions must be generated by means of a neighborhood structure. Such a set is obtained by a repeated local transformation of every solution contained in the current set. For a given current solution, two classes can be clearly distinguished. Firstly, an *exhaustive neighborhood exploration*, where the neighborhood is evaluated in a full and deterministic way. Every possible move of the current solution is applied and the neighboring solutions are all added to the candidate set. The solutions of the current set can then all be marked as visited. Second, a *partial neighborhood exploration*, where only a subset of moves are applied. The number of moves to be applied is generally defined by a user-given parameter.

Archiving. The archive allows to store either all or a subset of nondominated solutions found during the search process. Its main aim is to prevent the loss of interesting solutions. But archive members are also integrated into the search process by providing solutions to exploit in the DMLS model presented in this paper. Different archiving techniques can be distinguished depending on the problem properties, the designed algorithm and the number of desired solutions: (*i*) an *unbounded archive* or (*ii*) a *bounded archive*. Firstly, when an archive is maintained, it usually comprises the current nondominated set approximation, as dominated solutions are discarded. Then, an unbounded archive can be used in order to save the whole set of nondominated solutions. However, as some MCOPs may contain an exponential number of nondominated solutions, additional operations must be used to bound the archive size.

Stopping Condition. Since an iterative method computes successive approximations, a practical test is generally required to determine when the process must stop. Popular examples are a given number of iterations or a given runtime. However, when it is possible to mark archive members as visited, a natural stopping criterion arises when all archive solutions are marked as visited.

3 Computational Experiments

The goal of this section is to experiment the efficiency of some state-of-the-art strategies for both current set selection and neighborhood exploration. For each component, a set of 2 different schemes are investigated. This gives rise to a combination of 4 DMLS algorithms. Hence, with regards to the current set selection, either (*i*) each or (*ii*) a random single unvisited solution is selected from the archive. Next, with regards to the neighborhood exploration, either (*i*) all or (*ii*) a single random neighbor per solution is proposed as a candidate for integrating the archive. The corresponding algorithms are denoted by DMLS$^{(1\cdot 1)}$, DMLS$^{(1\cdot \star)}$, DMLS$^{(\star\cdot 1)}$ and DMLS$^{(\star\cdot \star)}$. Note that the algorithm denoted by and DMLS$^{(1\cdot 1)}$ is closely related to PAES [4], DMLS$^{(1\cdot \star)}$ to PLS [5], and DMLS$^{(\star\cdot \star)}$ to BLS [6]. For each problem instance to be solved, different maximum runtime values, from 2 to 20 minutes, have been investigated in order to study the evolution of the search efficiency over time. However, as some algorithms stop in a natural way, a simple random restart has been performed to continue the search process until the maximum runtime is reached. For all the experiments, the initial population size is set to 1, and an unbounded archive is maintained.

3.1 A Permutation Flowshop Scheduling Problem

The Flowshop Scheduling Problem (FSP) consists of scheduling N jobs on M machines. We here focus on a permutation FSP, where the operating sequences of the jobs are identical and unidirectional on every machine. We will consider a two-objective FSP (denoted by FSP-2), where both the makespan and the total tardiness are to be minimized. Additionally, we will also consider

a three-objective variant (denoted by FSP-3), where the maximum tardiness is the additional objective to be minimized. The reader is referred to [7] for more information on multiobjective scheduling.

The problem-related components used for the specific case of the FSP presented above are the following ones. Firstly, the representation is based on a permutation of size N. Next, the initialization strategy consists of generating solutions randomly. At last, the neighborhood is based on the *insertion* operator, *i.e.* a job at position i is inserted at position $j \neq i$, and the jobs located between positions i and j are shifted.

3.2 Results and Discussion

To evaluate the performance of the algorithms experimented in this paper, we consider various benchmark test instances[1]. Six problem instances involving from 20 jobs and 5 machines to 50 jobs and 20 machines are experimented. A set of 10 runs per instance has been performed for each search method. For a given instance, let Z^{all} denote the union of the outputs we obtained during all our experiments. We first compute a reference set $Z_N^\star$ containing all the nondominated points of Z^{all}. Now, to measure the quality of an output set A in comparison to $Z_N^\star$, we compute the difference between these two sets by using the unary hypervolume metric and the additive ϵ-indicator [8].

DMLS$^{(1\cdot 1)}$ and DMLS$^{(\star\cdot 1)}$ can generally not compete with other algorithms on small size problem instances. This can be explained by the fact that they do not handle any kind of natural stopping condition, so that they are never able to restart. On the contrary, DMLS$^{(1\cdot\star)}$ and DMLS$^{(\star\cdot\star)}$ quickly reach a state where each archive member is marked as visited, and can then restart with a different initial solution. However, DMLS$^{(1\cdot 1)}$ and DMLS$^{(\star\cdot 1)}$ perform better on bigger instances. In particular, these two algorithms appear very efficient in comparison to the others on the 50_10_01 and 50_20_01 instances when a short amount of runtime is available. Moreover, on the 50_20_01 instance of FSP-3, they perform better than all the other algorithms, even when a large runtime is allowed. Now, with regards to DMLS$^{(\star\cdot\star)}$, this approach performs very well on 20-job instances. For larger ones, the convergence is really slow in the biobjective case, but finally reaches competitive results after a long runtime. However, in the three-objective case, this method appears inefficient for problem instances of 50 jobs. Finally, the DMLS$^{(1\cdot\star)}$ algorithm, that embeds similar techniques than PLS [5], seems to reach the best overall performances. Indeed, it appears to be as good as DMLS$^{(\star\cdot\star)}$ on 20-job instances for both FSP-2 and FSP-3. For the 50_05_01 instance, it clearly outperforms the other algorithms all time long. For bigger instances, even if it is slightly dominated at the beginning of the search, it finally reaches the better results in the two-objective case. For FSP-3, same conclusions can be drawn on the 50_10_01 instance. But the last instance is the single one where DMLS$^{(1\cdot\star)}$ is always dominated by DMLS$^{(1\cdot 1)}$ and DMLS$^{(\star\cdot 1)}$.

[1] These instances are available at `http://www.lifl.fr/~liefooga/benchmarks/`.

4 Conclusion

In this paper, a unification of dominance-based local search approaches for multiobjective combinatorial optimization has been attempted. Such methods can be seen as a generalization of the classical single-objective hill climbing, combined with the use of a population of solutions. They are based on the iterative improvement of the set of nondominated solutions by means of a neighborhood operator. A unified model has been proposed and its main issues have been identified. The problem-independent components of current set selection, neighborhood exploration as well as archiving strategies have been especially discussed. This model has been used as a starting point for the design and the implementation of an open-source software framework for dominance-based multiobjective local search. This contribution has been conceived as a plug-in to be integrated into the ParadisEO-MOEO software framework[2]. At last, the issues of current set selection and neighborhood exploration have been experimentally compared on a multiobjective flowshop scheduling problem. We showed the benefit of performing a full neighborhood exploration in order to avoid the revaluation of some neighboring solutions and to reach a natural stopping criterion. Furthermore, we concluded that selecting a single solution from the current population to explore its neighborhood in an exhaustive manner was especially efficient for the problem under consideration. As a next step, we will investigate larger instances for the flowshop scheduling problem as well as other kinds of multiobjective problems.

References

1. Ehrgott, M., Gandibleux, X.: Approximative solution methods for multiobjective combinatorial optimization. TOP **12**(1) (2004) 1–89
2. Talbi, E.G.: Metaheuristics: from design to implementation. Wiley (2009)
3. Paquete, L., Stützle, T.: Stochastic local search algorithms for multiobjective combinatorial optimization: A review. In: Handbook of Approximation Algorithms and Metaheuristics. Chapman & Hall / CRC (2007)
4. Knowles, J.D., Corne, D.: Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation **8**(2) (2000) 149–172
5. Paquete, L., Chiarandini, M., Stützle, T.: Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In: [9]. Springer-Verlag (2004) 177–199
6. Angel, E., Bampis, E., Gourvés., L.: A dynasearch neighbohood for the bicriteria traveling salesman problem. In: [9]. Springer-Verlag (2004) 153–176
7. T'Kindt, V., Billaut, J.C.: Multicriteria Scheduling: Theory, Models and Algorithms. Springer-Verlag, Berlin, Germany (2002)
8. Zitzler, E., Thiele, L., Laumanns, M., Foneseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation **7**(2) (2003) 117–132
9. Gandibleux, X., Sevaux, M., Sörensen, K., T'Kindt, V., eds.: Metaheuristics for Multiobjective Optimisation. Volume 535 of Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin, Germany (2004)

[2] The plug-in is available at `http://paradiseo.gforge.inria.fr/DMLS/`.