A Study of Ant Colony Optimization Algorithms for a Biobjective Permutation Flowshop Problem

Trung Truc Huynh, Thomas Stützle,
Mauro Birattari and Yves De Smet
Université Libre de Bruxelles, Brussels, Belgium

Abstract

In this article, we present a study that compares variants of two ACO algorithms designed to tackle a biobjective permutation flowshop scheduling problem where the makespan and the total tardiness are the objectives considered. These two algorithms use respectively one and two pheromone matrices. The analysis of the results gives indications on the choices to adopt when designing an ACO approach for biobjective flowshop scheduling.

1 Introduction

Ant Colony Optimization (ACO) is a population-based SLS method inspired by the foraging behaviour of some ants species [1]. The main idea in ACO algorithms is to mimic the pheromone trails used by real ants. In ACO algorithms, artificial pheromone trails serve as a distributed, numerical information that the ants use to probabilistically construct solutions to the problem being tackled and to adapt these pheromone at run-time to reflect their search experience.

The permutation flowshop scheduling problem (PFSP) requires scheduling n jobs with given processing times on each of m machines such that the job sequence on all machines is identical. Typical further assumptions are that each job can be processed on only one machine at a time, operations are not preemptable, jobs are available for processing at time zero and setup times are independent. Given its industrial relevance, various variants of the PFSP have been considered, including variants that consider various objective functions to be optimized simultaneously. In multiobjective optimization, the goal is to identify all efficient alternatives, that is, the set of Pareto optimal solutions that comprises all solutions that are non-dominated solutions w.r.t. Pareto dominance, which is defined as follows: if all objectives are minimization, an objective vector $V(x)$ is said to dominate a vector $V(x')$ if and only if $\forall i : v_i(x) \leq v_i(x') \wedge \exists i. v_i(x) < v_i(x')$. In this work, we consider a biobjective version of the PFSP and the two objectives, f_1 and f_2, considered are to minimize the makespan ($f_1 = C_{\max} = \max\{C_1, C_2, \ldots, C_n\}$) and the total tardiness ($f_2 = T = \sum_{i=1}^{n} T_i$, where $T_i = \max\{0, C_i - d_i\}$, where d_i is the due date and C_i the completion time of of the job i).

2 ACO algorithms for biobjective PFSP

The two proposed approaches uses multiple runs of an ACO algorithm and the idea is to force each run to search in different regions of the space. The first method (1phero) consists in aggregating the two objective functions into one single objective function $F = \lambda_1 f_1 + \lambda_2 f_2$, $\lambda_1 + \lambda_2 = 1$. For approximating different areas of the Pareto front, dynamically the search directions are modified by modifying the weights λ_1 and λ_2: if k weight vectors are used, λ_1 and λ_2 change by $\pm 1/k$ when moving from one weight vector to the next one, that is, the weights are uniformly spaced and the minimum amount of change is done when moving from one weight vector to the next one. The second approach (2phero) associates to each objective one pheromone matrix and ants will construct their solutions based on an aggregation of the two pheromone matrices, where a pheromone matrix entry at position i, j is defined by $\tau_{ij} = \lambda_1 \tau_{ij}^1 + \lambda_2 \tau_{ij}^2$; again the values of λ_1 and λ_2 are modified to attain different areas of the Pareto front. The use of two pheromone matrices has already been proposed in [2].

Each resulting ACO algorithm is combined with an iterative improvement algorithm in the insert neighbourhood and with a *component-wise* local search that looks for non dominated solutions in the insert neighbourhood and adds these to the archive. For each of the 1phero and 2phero algorithms, two further variants have been studied. In the scratch approach, the colonies for each weight vector λ work independently of each other. In the 2phase approach, the best solution found for the previous weight, is used for the initialisation for the current weight vector. Hence, for each λ vector, a colony starts with a solution that was good for the previous weight. Hence, the solutions in the 2phase approach are treated like a chain. For the two pheromone matrices approach (2phero), two further sub-variants have been studied. These two configurations, 2pheroG and 2pheroL differ in the update of the pheromones. In 2pheroG, only the best solutions found for each objective function across all already considered weight vectors are allowed to update the pheromone matrices τ_{ij}^1 and τ_{ij}^2, respectively. For 2pheroL, the update is done by the best solutions found for the current aggregation weight. In addition to these variants, we also have studied the influence of the number of weights used and the influence of the direction in which the weight vector changes (initial weight one for either f_1 or f_2).

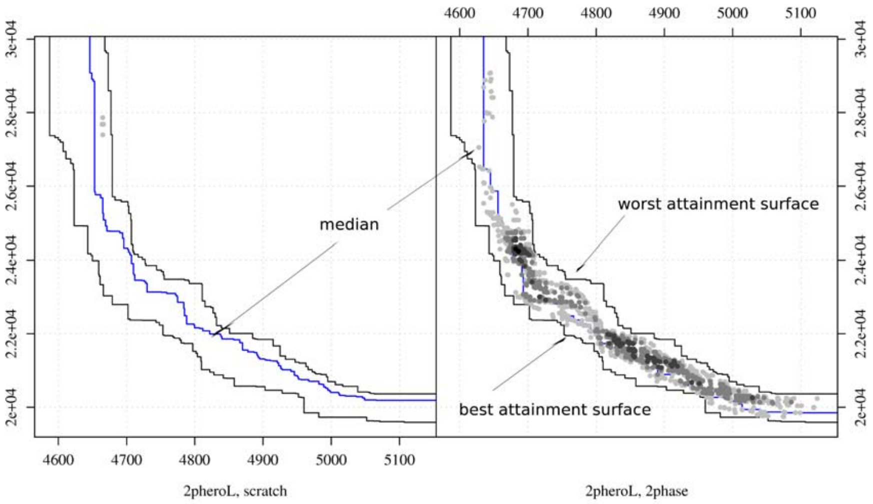

Figure 1: Difference of EAFs, comparison scratch-2phase approach

3 Experimental results

Each of the variants defined above was experimentally tested in a number of benchmark instances using ten independent trials. The computation times were chosen the same for all algorithmic variants. The analysis of the results is based on pairwise comparisons between the outcomes of algorithms. To avoid the known short-comings of performance measures for multi-objective optimizers [4], we first examine whether for one algorithm the outperformance relation holds and, if this is not the case, we compute attainment functions, apply statistical tests on the equality of two attainment functions and, if the test is rejected, we use the visualization of the difference of two attainment surfaces to detect these [3]. In the plots of the attainment surfaces (see Figures 1 and 2), we draw three lines where the rightmost connects the set of points attained by any run of the two configurations (worst case performance) and the leftmost connects the best set of point attained (best case performance). The line between the two gives the median. The different shades of gray indicate how large are the differences at specific points of the attainment function (only differences above 0.2 are given); the darker the point, the larger is the difference. On the left is given the advantage of algorithm A over B, while on the right side, the advantage of B over A is given, where A and B are the variants indicated below the x-axis of each plot.

The two examples of the visualization of the differences between two algorithms indicate the following facts. Figure 1 indicates that 2pheroL,2phase performs better than 2pheroL,scratch in most regions of the objective space.

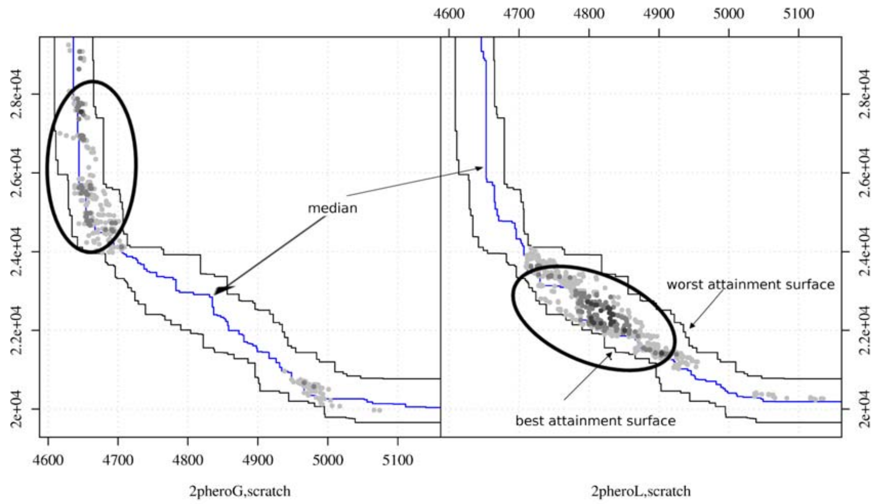

Figure 2: Difference of EAFs, comparison global-local strategy

This means that for the tackled instance, it is advantageous to keep information from previous colonies. In Figure 2, we can observe that the two strategies, 2pheroL,scratch and 2pheroG,scratch have advantages in two distinct regions: 2pheroL,scratch is clearly better in the center, while 2pheroG,scratch is better in the upper left corner.

The main observations of these and further comparisons can be summarized as follows.

– The use of the component-wise local search often improves significantly the quality of the approximation obtained.

– The 2phase approach in most cases leads to an improved performance over the scratch approach.

– A minimum number of weight vectors seems to be necessary to obtain good approximation sets.

– When comparing 2pheroG and 2pheroL, typically the former is less performing in the middle of the front but it can give advantages towards the extremes, where one objective receives a very high weight.

– The direction of the changes of the weight vector can have an influence on the performance in the 2phase approach.

– In general, the performance of the different variants depends strongly on the instances tackled and, apparently, strongly on the number of machines.

Future work can focus on different directions. A first would certainly be to extend further the empirical basis of our findings by enlarging the experimental study by more instances or a more systematic modification of instance characteristics. Other directions are to explore further some of the observations made here. One is to study with our or similar algorithms the influence of different directions taken in the changes to the weight vectors. Another would be to provide combinations between the 2pheroG and 2pheroL search strategies, given that each has its own advantages in different areas of the approximation to the Pareto front. Finally, comparisons with other multiobjective optimizers will also be necessary.

References

[1] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.

[2] S. Iredi, D. Merkle, and M. Middendorf. Bi-criterion optimization with multi colony ant algorithms. In E. Zitzler et al., editors, *Proceedings of EMO'01*, volume 1993 of *LNCS*, pages 359–372. Springer Verlag, 2001.

[3] M. López-Ibáñez, L. Paquete, and T. Stützle. Hybrid population-based algorithms for the bi-objective quadratic assignment problem. *Journal of Mathematical Modelling and Algorithms*, 5(1):111–137, 2006.

[4] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.