Hyperheuristic as Component of a Multi-Objective MetaHeuristic

Nadarajen Veerapen[1,2], Dario Landa-Silva[2], Xavier Gandibleux[1]

[1]Département informatique, Faculté des Sciences et Techniques, Université de Nantes, Nantes, France

[2]ASAP Group, School of Computer Science, University of Nottingham, Nottingham, United Kingdom

nadarajen.veerapen@etu.univ-nantes.fr, jds@cs.nott.ac.uk, xavier.gandibleux@univ-nantes.fr

August 14, 2009

Abstract

The numerical peculiarities which inhabit the numerical instance of a MOCO problem may seriously decrease the effectiveness of an approximation method. To deal with this problem we propose a flexible two-phase method for MOCO. Phase 1 produces a good approximation of the efficient frontier. However it may not be of good enough quality in terms of density. The aim of phase 2 is to tackle this problem in a flexible way so as to deal with the potential numerical peculiarities. We test this proposition on the multi-objective Traveling Salesman Problem for which there exists a number of low-level heuristics.

1 Introduction

Multi-objective combinatorial optimization (MOCO) considers $p \geq 2$ (conflicting) objectives to find a set of efficient solutions within a set of discrete feasible solutions, see [3] for more details. It is assumed that a solution which optimizes all objectives simultaneously does not exist. A solution $\hat{x} \in X$ is said to be *efficient* if there is no $x \in X$ such that x *dominates* $\hat{x}$ ($x_k \leq \hat{x}_k, \forall k \in \{1, \ldots, p\}$ with at least one strict inequality).

Finding the whole set of efficient solutions is often not necessary. However, having a good distribution of an approximation of this set is useful for decision makers to perform informed choices.

As resolution methods try to get better results, they often become more complicated, requiring expert knowledge to use them effectively. The hyperheuristic approach [1] provides a high-level view of the problem. Its aim is to allow the user to provide a number of heuristics, usually low-level ones, to solve a problem and then the hyperheuristic tries to find on its own which heuristics are the best online, i.e. as the search progresses. As such, a hyperheuristic does not operate in the *solution space* (this is done by the selected heuristics) but in the *heuristic space*. This makes it different from variable neighborhood search (VNS) [7] which looks for solutions based on the systematic change of neighborhood during the search. There exists a domain barrier between the hyperheuristic and the low-level heuristics and, ideally, the hyperheuristic requires no or minimal knowledge of how the heuristics work but relies instead on the analysis of one or more objective functions (knowing whether maximization or minimization are required) and heuristic performance indicators such as running time.

This work proposes the use of a hyperheuristic to make the efficient frontier more dense by starting from a sparse approximation of the efficient frontier. The method is applied to the multi-objective Traveling Salesman Problem (moTSP). To the best of our knowledge, hyperheuristics have yet to be applied to this problem.

The objective of the TSP is to find the shortest tour passing only once through every location which has to be visited (a Hamiltonian cycle). The multi-objective version of the TSP is of interest because it represents many practical situations, for example having to make a compromise between travel time and cost of transportation.

This paper presents some initial results and discusses possible improvements and future research.

2 Overview of the Method

Our algorithm consists of two phases which are described below. We note that the key contribution here is the second phase of the proposed approach.

2.1 First Phase

The first phase computes a very good subset of efficient solutions. For the bi-ojective TSP we use the same method employed by Lust and Teghem [6]. When considering problems with more objectives, in our case three objectives, the first phase uses the algorithm proposed by Przybylski et al. [8] where the exact solver is replaced by the Lin-Kernighan heuristic [5].

2.2 Second Phase

The second phase iteratively drives a subset of the population across the potential efficient frontier with the goal of maximizing the hypervolume indicator [9]. The non-dominated solutions found in this manner are added to the population.

Components and Layout of the Method. The algorithm requires a *set of one or more heuristics* whose only requirement is to implement a simple interface so that the search mechanism can manipulate them.

Next, the algorithm needs an initial *population of solutions* P_0, $|P_0| \geq 1$. It maintains an *archive* A of all non-dominated points found and, at each iteration, only uses a running population P of maximum size S. The *hypervolume indicator* allows us to consider the movement of each solution with respect to the current running population and not as a point on its own.

At each iteration, each solution $p \in P$ is considered, a heuristic is selected and the neighborhood of p is explored to find a new point which increases the hypervolume. Any non-dominated solution found during this process is added to the archive. Should a point with a strictly improving hypervolume be found, the same heuristic is applied to this new point in a descent fashion.

If the maximum size of the running population is not reached, improving solutions are added to P as is, otherwise they replace the point they were a neighbor of. If no solutions in P were moved during the last iteration, a new running population is randomly selected from the archive. This prevents the algorithm from getting stuck in a local optimum and contributes to the diversity of the solution set. Since calculating the hypervolume and non-dominated sorting occur constantly throughout the algorithm, it is better to keep S small. We arbitrarily choose $S = 20$.

Heuristic Selection Mechanism. At the start of the search, all heuristics have the same score of 1. The performance of the heuristics is inferred through a system of reward and punishment, whereby improving heuristics obtain a higher score and non-performing heuristics a lower one. The heuristic with the best rank is selected in each iteration of the process described above.

To supplement this strategy, a tabu list is also used to prevent worse heuristics from being used during a certain amount of time (even if it was performing well previously). This strategy is inspired by the one used in Burke et al. [2]. A heuristic is included in the tabu list if it has not been able to improve the distribution by moving a solution in the given amount of time it was allowed to run. However, if an improving solution has been found, the tabu list is cleared (it is also cleared if all heuristics turn out to be tabu). No aspiration criterion is used.

Next, we present some results and name the hyperheuristic Best Rank with Tabu List (BRTL).

3 Initial Experimental Results

A population of 11 low-level classic TSP heuristics is used: subpath insertions and swaps, the 2-exchange move and a dummy heuristic (the first and second cities are inverted). Running time was set to 30 s and the neighborhood of each move was explored for a maximum of 50 ms. Averages of 10 runs are reported. We compare our method with two recent algorithms: evolutionary multi-objective simulated annealing (EMOSA) [4] and two-phase Pareto local search (2PPLS) [6]. Here, BRTL and 2PPLS share the same first phase (but not EMOSA).

BRTL manages to outperform EMOSA for the hypervolume indicator in three out of the four published results for convex instances (Table 1). BRTL performs less well than our implementation of 2PPLS for instances with less objectives (Table 2). However, with more objectives, if the running time of 2PPLS is capped in the same way as for BRTL, initial results indicate that BRTL obtains better results.

Instance	BRTL	EMOSA
kroAB50	**0.3544**	0.2839
kroBC50	**0.4327**	0.2809
kroAB100	**2.1782**	1.9060
kroBC100	1.8630	**1.9392**

Table 1: Hypervolume (10^{10}) comparison of BRTL with EMOSA

Instance	Algo.	$\mathcal{H}(10^8)$	R	Time(s)
kroAB100	BRTL	225.84	0.93516	30
	2PPLS	**226.11**	**0.93526**	**13**
Cluster100	BRTL	233.12	0.94672	30
	2PPLS	**233.35**	**0.94679**	**13**
kroAB200	BRTL	835.37	0.875358	30
	2PPLS	**1076.08**	**0.94507**	**20**
kroABC50	BRTL	*4092608*	*0.98286*	**30**
	2PPLS	3454695	0.97029	**30**
	2PPLS	–	–	3600+

Table 2: Phase 2 of BRTL and 2PPLS

4 Conclusion

Our approach is a new generic resolution method for MOCO. Work is still needed to evaluate the potential advantage of the hyperheuristic with regard to its flexibility when dealing with various numerical instances and its ability to intelligently switch between heuristics. Potential avenues of investigation include a simpler first phase. The second phase could be improved in a number of ways: better indicators of the quality of the distribution of set, intelligent selection of the running population and a smarter selection mechanism. The approach also needs to be tested on more instances and other problems.

Acknowledgements. The authors are grateful for Anthony Przybylski's insights in the implementation of phase 1 and for his helpful remarks.

References

[1] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg. Hyper-heuristics: An emerging direction in modern search technology. In *Handbook of Metaheuristics*, pages 457–474. Kluwer Academic Publishers, 2003.

[2] E. Burke, D. Landa-Silva, and E. Soubeiga. Multi-objective hyper-heuristic approaches for space allocation and timetabling. In *Metaheuristics: Progress as Real Problem Solvers*, pages 129–158. Springer, 2005.

[3] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460, 2000.

[4] H. Li and D. Landa-Silva. Evolutionary Multi-objective Simulated Annealing with Adaptive and Competitive Search Direction. In *Proceedings of IEEE Congress on Evolutionary Computation, 2008. CEC 2008*, pages 3310–3317. Springer, 2008.

[5] S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, pages 498–516, 1973.

[6] T. Lust and J. Teghem. Two-phase pareto local search for the biobjective traveling salesman problem. *Journal of Heuristics*, 2009, DOI: 10.1007/s10732-009-9103-9 (in print).

[7] N. Mladenovic and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100, 1997.

[8] A. Przybylski, X. Gandibleux, and M. Ehrgott. A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *To appear in INFORMS Journal on Computing*, 2009.

[9] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.