# A Study of Local Optima in the Biobjective Travelling Salesman Problem

Luis Paquete, Marco Chiarandini and Thomas Stützle

*FG Intellektik, Technische Universität Darmstadt, Alexanderstr. 10, Darmstadt, GERMANY*
{*lpaquete, machud, tom*}*@intellektik.informatik.tu-darmstadt.de*

Abstract: In this article, we study local optima, in the Pareto sense, of the biobjective Travelling Salesman Problem by means of simple extensions of local improvement algorithms. We propose this approach as a first step for tackling a multiobjective problems that should be used as a reference for future comparisons with metaheuristics in multiobjective problems. We present experimental results on well known benchmark instances and evaluate the performance of our approach by means of attainment functions, the coverage measure and the expected value of the weighted Tchebycheff scalarizing function. Using these performance measures, we also compare our approach to methods from the literature. Additionally, we give details on the run-time behaviour of our local optimizer using run-time distributions.

Keywords: Stochastic local search, Local optima, Performance assessment, TSP

## 1 Introduction

Local search algorithms are at the core of the most successful metaheuristics to solve a wide variety of single objective combinatorial problems such as the Traveling Salesman Problem (TSP). Local search algorithms for single-objective problems can also be adapted in rather straightforward ways to multiobjective problems, opening their use inside a metaheuristic for multiobjective problems. A first idea for extending local search algorithms to multiobjective problems is to maintain the same search strategy of the single-objective local search. The main difference between the single- and the multiobjective case then lies in the acceptance criterion of new solutions in the local search. Here, we adopt an acceptance criterion in the Pareto sense: a solution in the neighborhood of the current one is accepted if it is *not worse, i.e.*, better in at least one objective than the current one and the ones kept on an archive of non-dominated solutions. As a side effect, this extension of local search allows to define the notion of Pareto local optima and has the potential of serving as a reference for comparisons of multiobjective metaheuristics in the very same way as simple local search algorithms are used as a benchmark of the quality single-objective metaheuristics need to reach at least to be of some use.
In this article, we tackle the biobjective TSP by local improvement algorithms based on simple tour modifications and using the notion of Pareto optimality. The experimental analysis is based on well known benchmark instances and the performance assessment is done by means of attainment functions [3], the coverage measure ($C$ measure) [12] and the expected value of the weighted Tchebycheff scalarizing function ($R$ measure) [9]. An insight into the run-time behavior is obtained by measuring run-time distributions [8]. Some comparisons with the literature are also presented.

## 2 The Multiobjective TSP

Given a complete, weighted graph $G = (N, E, d)$ with $N$ being the set of nodes, $E$ being the set of edges fully connecting the nodes, and $d$ being a function that assigns to each edge $(i, j) \in E$ a vector $d_{ij}$, where each element corresponds to a certain measure (e.g., distance, cost) between $i$ and $j$, then the multiobjective TSP is the problem of finding a "minimal" Hamiltonian circuit of the graph, *i.e.*, a closed tour visiting each of the $n = |N|$ nodes of $G$ exactly once. In this study, we consider symmetric problems, *i.e* $d_{ij} = d_{ji}$ for all pairs of nodes $i, j$, and "minimal" refers to the notion of Pareto optimality.

Usually there is not only one Pareto optimal solution, but several, which are elements of a *Pareto global optima set*. This set contains all solutions that are not dominated by any other solution. In analogy, we define the *Pareto local optima set* as the set containing all solutions that are *not worse* than any solution of a pre-defined neighborhood of any solution in the Pareto local optima set, and there is no solution in the set which dominates any of the solutions in this set.
The problem of finding the Pareto global optima set is $\mathcal{NP}$-hard [4]. Since for many problems exact solutions become quickly infeasible with increasing instance size, the main goal shifts from obtaining Pareto global optimal solutions to obtaining a good approximation of the Pareto optimal set and local search algorithms and metaheuristics seem to be a suitable approach for this task [6, 9].

# 3 The Local Search Algorithms

Our local search algorithm for the multiobjective case uses the same notion of neighborhood as in the single-objective case. However, the acceptance criterion of the single-objective local search algorithms needs to be changed to take into account several objectives: While in iterative improvement local search for the single-objective case a solution is accepted if it is better than the current one, for multiobjective problems an extension of this acceptance criterion should take into account the concept of Pareto optimality. For the sake of simplicity, a first approach for an acceptance criterion may be to accept a neighboring solution, if it is *not worse* than current solution. However, a solution accepted by such a criterion may be dominated by other solutions seen previously in the local search. To avoid this, we maintain an archive of non-dominated solutions. The final acceptance criterion we are using for the local search is the following: each new solution is compared to the current one. If the new solution is *not worse* in the Pareto sense, it is compared in a next step to all solutions in the archive. Only if the new solution is not dominated by any solution of the archive, it is finally accepted and included into the archive. In fact, during this process, some solutions from the archive could become dominated by some of the recently introduced ones. Such solutions are (continuously) eliminated from the archive.
The local search algorithm starts from a randomly generated initial solution that is put into the archive. It then works as follows: First, it picks a solution randomly from the archive and iteratively explores the neighborhood of this solution. If a *not worse* solution is found, it is compared to the solutions in the archive and the local search continues. If at some point all neighboring solutions were explored and none is accepted any more, the solution is flagged as *visited*, *i.e*, this solution is a Pareto local optima solution and it will not be chosen again.
We terminate the local search procedure if all the neighborhoods of all solutions in the archive were explored, *i.e.* every solution in the Pareto local optimum set is flagged as *visited*. In this case, a Pareto local optimum set is found. It should be remarked that this local search is similar to PAES [10], although we stress the importance of neighborhood and use a simpler acceptance criteria for comparing and accepting non-dominated solutions. The algorithm 1 presents a pseudo-code for the local search.
For the TSP case, we considered a 2-opt and a 2H-opt local search algorithm:

**2-opt Algorithm** A 2-opt exchange deletes two edges and replaces them with the only possible couple of new edges that does not break the tour in two disjoint cycles. The 2-opt algorithm, given some starting tour, applies improving 2-opt exchanges and ends when no further improving 2-opt exchange can be found. More precisely, in our case, at each step of the algorithm all edges are investigated according to a randomly chosen order and the first exchange encountered that leads to a non-dominated new solution is applied. If all $n(n-1)/2$ pairs of edges were considered and no solution is accepted, a Pareto local optimum solution is reached.

**2H-opt Algorithm** A 2H-opt exchange, in addition to considering 2-opt exchanges, moves a single city from one position in the tour to another. As suggested in Bentley [1], our implementation is a simple extension of the 2-opt algorithm. The search strategy employed in the 2H-opt algorithm is then analogous to the 2-opt case.

**Algorithm 1** General local search for multiobjective optimization

```
t ← 0
s_t ← GenerateInitialSolution()
UpdateArchive(s_t)
repeat
   for all s'_t ∈ Neighborhood(s_t) do
      Evaluate(s'_t)
      if s'_t not worse then s_t then
         UpdateArchive(s'_t)        % s'_t is accepted if not worse then any solution in the Archive
      end if
   end for
   if all Neighborhood(s_t) visited then
      s_t ← visited
   end if
   t ← t + 1
   s_t ← PickArchive()
until termination condition met
```

## 4 Experimental Results

As benchmark instances we considered all six paired combinations of the benchmark 100-city instances kroA100, kroB100, kroC100 and kroD100. These benchmark instances were defined in [6] and were also used in [2, 9]. Since our local search uses random decisions, it is expected to return different outcomes at each run. Therefore, we ran both algorithms 25 times on each of the 6 instances, until a Pareto local optimum set was found. The code was written in C++ and tested on a Dual Athlon with 1200 MHz and 512 MB of RAM.

Due to the stochastic nature of the runs, statistical inference procedures based on the attainment functions are appropriate for the performance assessment of the algorithms [3]. Considering an arbitrary instance $A$, we formulated a null hypothesis that there is no difference in performance between a local search algorithm using 2-opt and 2H-opt in terms of attainment functions on instance $A$ *versus* the alternative that there is a difference between both. A significance value of 0.05 was defined *a priori* and a permutation test with a Smirnov-like test statistic [11] was conducted at each instance. From the six biobjective instances, the instances (kroA100, kroD100) and (kroB100, kroD100) did not presented any statistical difference for both algorithms. On the instances, for which statistical differences were found, the sign of the differences indicated a better performance of 2H-opt in the "tails" of the Pareto local optima set.

Figure 1 plots the 50$\%$ attainment surfaces [5] from both algorithms on the instance (kroA100, kroB100). This plot describes the typical outcome obtained from several runs of both algorithms in the instance considered. For reference, the outcome obtained by the Genetic Local Search (GLS) in [9] is also plotted.[1] In Figure 1, left, almost no difference among the algorithms are visible. Figure 1, right, stresses the differences in the "tails" by using a loglog scale. The results are similar on the other instances.

For the instances for which we observed statistical differences between the algorithms, we computed the $R$ measure [7]. The metric parameters were defined according to [9].[2] Figure 2, left, plots the boxplot of the metrics for the instance (kroA100, kroB100). The outcome obtained in [9] according to this measure is also plotted for reference. This plot also stresses the better performance of 2H-opt algorithm when compared to 2-opt. The same pattern is also found in all the instances.

The $C$ measure [12] was also applied to both algorithms. Figure 2, middle, presents the box plot for instance (kroA100, kroB100) for this metric. The values observed can be interpreted as by how much the outcome of each run of the algorithm covers all the runs of the other. For all instances we observed a better performance of 2H-opt when compared to 2-opt according to this metric. Figure 2, right, presents a comparison of 2H-opt with GLS in [9] for reference.

We analyzed the run-time behavior of both algorithms by means of Run Time Distributions (RTD) [8],

[1]The results were taken from file ND_kroab100_4.txt taken from http://www-idss.cs.put.poznan.pl/ jaszkiewicz/motsp.

[2]We downloaded the code from http://www-idss.cs.put.poznan.pl/ jaszkiewicz/mokp, compiled it with gcc 2.95.3 on the Linux Suse 7.3 operating system.

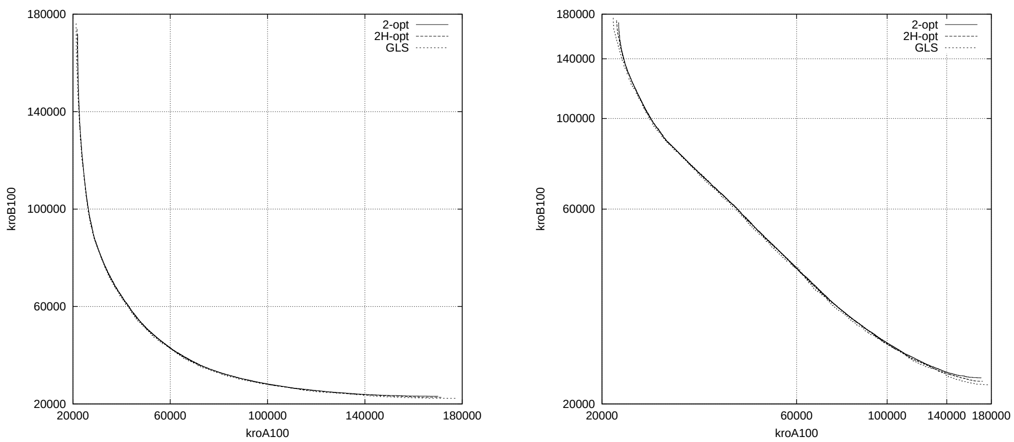


Figure 1: 50% attainment surfaces for the 2-opt and 2H-opt algorithms and the outcome for the GLS on the instance (kroA100, kroB100) with linear scale (left) and loglog scale (right).

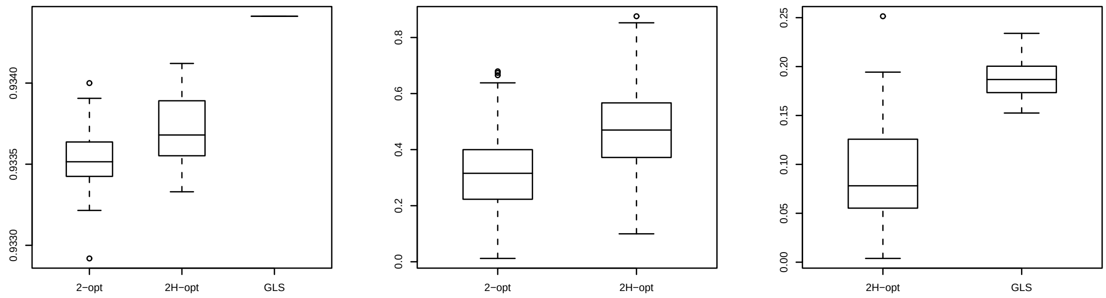


Figure 2: Results of 2-opt, 2H-opt and GLS in the instance (kroA100, kroB100) according to the $R$ measure (left) and to the $C$ measure between 2-opt and 2H-opt (middle) and 2H-opt and GLS (right)

which give the empirical probability of finding a solution as a function of the run-time. For the instances in which no difference was found in terms of attainment functions, we considered the CPU-time of both algorithms until a Pareto local optima set is found. Otherwise, we estimate the empirical distributions to reach a certain measure value bound of the $R$ metric.

Figure 3, left, plots the RTDs of both algorithms on the instance (kroA100, kroD100). The plot indicates a clearly faster computation time of 2-opt. Figure 3, right, plots the RTDs on the instance (kroA100, kroB100) considering two bounds on the $R$ metric values: the minimum value (a) and the median (b) for the 2H-opt. These RTDs indicate a trade-off between the two algorithms: 2-opt is faster than 2H-opt, but gets worse solutions, as stressed previously by the other measures.

# 5 Conclusions and Further Work

The local search algorithms presented in this article are appealing, because they are based on simple principles and they do not require (i) the calculation of ideal points, (ii) the aggregation of criteria, and (iii) no parameters to be determined. Despite the high quality solutions the local search algorithms obtained, some preliminary experimental results suggest that they are not really competitive with the currently best performing metaheuristics. However, the proposed local search algorithm have the potential to serve as a baseline comparison to benchmark the performance of metaheuristics for multiobjective combinatorial optimization problems.

In the biobjective TSP benchmarks studied, the performance of the two algorithms seems to be instance dependent: In some cases, no statistical difference was found between the outcomes of 2-opt and 2H-opt, while in some other cases they differed significantly. Therefore, also some care must be taken when aggregating results on such instances. One obvious observation is that 2-opt seems to find faster a Pareto local optimum set in any case.

The next steps for continuing this research are (i) the study of larger neighborhoods such as 3-opt etc., (ii) a more fine-tuned local search implementation exploiting speed-up techniques such as *don't look bits*

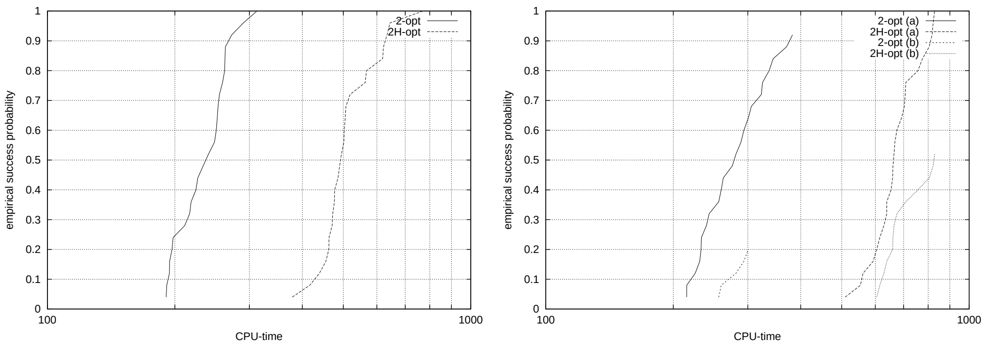


Figure 3: Run-time distribution of 2-opt and 2H-opt algorithms in the instance (kroA100, kroD100) (left) and in the instance (kroA100, kroB100) (right) considering a bound on the minimum value (a) and on the median (b) of $R$ measure in the 2H-opt.

and nearest neighbor lists, and (iii) the introduction of simple techniques to better approximate the tails of the Pareto fronts.

**Acknowledgments** We would like to thank Carlos Fonseca and Viviane Grunert da Fonseca from the University of Algarve for valuable discussions on the topic of this research. This work was supported by the Metaheuristics Network, a Research Training Network funded by the Improving Human Potential programme of the CEC, grant HPRN-CT-1999-00106. The information provided is the sole responsibility of the authors and does not reflect the Community's opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

[1] J.L. Bentley. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on Computing*, 4(4):387–411, 1992.

[2] P. C. Borges and P. H. Hansen. A study of global convexity for a multiple objective travelling salesman problem. In C.C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 129–150. Kluwer, 2000.

[3] V. G. da Fonseca, C. Fonseca, and A. Hall. Inferential performance assessment of stochastic optimisers and the attainment function. In E. Zitzler and et al., editors, *Evolutionary Multi-Criterion Optimization (EMO'2001)*, LNCS 1993, pages 213–225. Springer Verlag, 2001.

[4] M. Ehrgott. Approximation algorithms for combinatorial multicriteria problems. *International Transactions in Operations Research*, 7:5–31, 2000.

[5] C. Fonseca and P. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In H.-M. Voigt and et al., editors, *Proceedings of the 4th Conference on PPSN*, LNCS, pages 584–593. Springer Verlag, 1996.

[6] M.P. Hansen. Use of subsitute scalarizing functions to guide a local search base heuristics: the case of motsp. *Journal of Heuristics*, 6:419–431, 2000.

[7] M.P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby , Denmark, 1998.

[8] H. Hoos and T. Stützle. Evaluating las vegas algorithms – pitfalls and remedies. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 238–245. Morgan Kaufmann, 1998.

[9] A. Jaszkiewicz. Genetic local search for multiple objective combinatorial optimization. *European Journal of Operational Research*, 1(137):50–71, 2002.

[10] J. Knowles and D. Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Proceedings of CEC'99*, pages 98–105, 1999.

[11] K. Shaw, C. Fonseca, A. Nortcliffe, M. Thompson, J. Love, and P. Fleming. Assessing the performance of multiobjetive genetic algorithms for optimization of a batch process scheduling problem. In *Proceeding of CEC'99*, pages 37–45, 1999.

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