On the Integration of a TSP Heuristic into an EA for the Bi-objective Ring Star Problem

Arnaud Liefooghe, Laetitia Jourdan, Nicolas Jozefowiez, El-Ghazali Talbi

On the Integration of a TSP Heuristic into an EA for the Bi-objective Ring Star Problem

Arnaud Liefooghe[1], Laetitia Jourdan[1], Nicolas Jozefowiez[2,3], and
El-Ghazali Talbi[1]

[1] LIFL – CNRS – INRIA Lille-Nord Europe,
Université des Sciences et Technologies de Lille,
Parc Scientifique de la Haute Borne, 40 av. Halley, 59650 Villeneuve d'Ascq, France
{Arnaud.Liefooghe,Laetitia.Jourdan,El-Ghazali.Talbi}@lifl.fr
[2] LAAS – CNRS, Université de Toulouse,
7 av. du Colonel Roche, F-31077 Toulouse, France
Nicolas.Jozefowiez@laas.fr
[3] Université de Toulouse, INSA, France

Abstract. This paper discusses a new hybrid solution method for a bi-objective routing problem, namely the bi-objective ring star problem. The bi-objective ring star problem is a generalization of the ring star problem in which the assignment cost has been dissociated from the cost of visiting a subset of nodes. Here, we investigate the possible contribution of incorporating specialized TSP heuristics into a multi-objective evolutionary algorithm. Experiments show that the use of this hybridization scheme allows a strict improvement of the generated sets of non-dominated solutions.

1 Introduction

The purpose of the Bi-objective Ring Star Problem (B-RSP) is to locate an elementary cycle, the so-called *ring*, on a subset of nodes of a graph while optimizing two conflicting costs. First is the minimization of a *ring cost*, proportional to the length of the cycle. Then, nodes that do not belong to the ring are all assigned to visited ones so that the associated cost is minimal. The resulting *assignment cost* is the second objective to be minimized. In spite of its natural bi-objective formulation, this problem is generally investigated in a single-objective way, either where both costs are combined [12] or where the assignment cost is treated as a constraint [13]. Note that both versions of the problem have also been heuristically solved in [16, 18]. As pointed out in [10], a large number of routing problems are formulated as multi-objective optimization problems, and according to the same paper, the B-RSP is a generalization of a mono-objective problem. In [15], different multi-objective evolutionary algorithms have been proposed for the B-RSP. Although the approaches were already encouraging, even compared to state-of-the-art mono-objective methods, a few improvement points can be identified. First, a lack of efficiency has been detected on the ring cost throughout the output solutions. Second, the population initialization strategy used within all the search methods was a bit rudimentary, each initial solution having approximately half of its nodes in the cycle. The challenge is then to overcome the identified problems in order to improve the efficiency of those search methods.

An interesting property of the problem under consideration is that, given a fixed set of visited nodes, the related assignment cost is always optimal. It is not the case for the ring cost, for which a classical Traveling Salesman Problem (TSP) is still to be solved among the set of nodes that belong to the ring. Then, once is decided which nodes are visited or not, an objective function is much more difficult to optimize. However, a large number of efficient heuristic methods has been proposed for the TSP. In this paper, our aim is to present a hybrid metaheuristic combining a multi-objective evolutionary algorithm and a problem-specific heuristic, initially designed for the TSP. Approaches where a TSP heuristic is successfully integrated into a multi-objective evolutionary algorithm can, for instance, be found in [8, 9].

The reminder of the paper is organized as follows. In Section 2, we give the necessary background for multi-objective optimization, we introduce the B-RSP and we present a heuristic devoted to the TSP. The hybrid metaheuristic proposed to solve the B-RSP is detailed in Section 3. In Section 4, computational experiments are conducted. At last, conclusions and perspectives are drawn in the last section.

2 Background

In this section, we first discuss multi-objective optimization and define some related concepts. Then, we present the bi-objective ring star problem in details and we introduce a heuristic devoted to the traveling salesman problem.

2.1 Multi-objective Optimization

A general *Multi-objective Optimization Problem* (MOP) can be defined by a set of $n \geq 2$ objective functions $f_1, f_2, \ldots, f_n$; a set of feasible solutions in the *decision space*, denoted by X; and a set of feasible points in the *objective space*, denoted by Z. Each function can be either minimized or maximized, but we here assume that all n objective functions are to be minimized. To each decision vector $x \in X$ is assigned exactly one objective vector $z \in Z$ on the basis of a vector function $f : X \rightarrow Z$ with $z = f(x) = (f_1(x), f_2(x), \ldots, f_n(x))$.

Definition 1. *An objective vector $z \in Z$* weakly dominates *another objective vector $z' \in Z$ if and only if $\forall i \in [1..n]$, $z_i \leq z'_i$.*

Definition 2. *An objective vector $z \in Z$* dominates *another objective vector $z' \in Z$ if and only if $\forall i \in [1..n]$, $z_i \leq z'_i$ and $\exists j \in [1..n]$ such as $z_j < z'_j$.*

Definition 3. *An objective vector $z \in Z$ is* non-dominated *if and only if there does not exist another objective vector $z' \in Z$ such that z' dominates z.*

A solution $x \in X$ is said to be *efficient* (or *non-dominated*) if $f(x)$ is non-dominated. The set of all efficient solutions is the *efficient set*, denoted by X_E. The set of all non-dominated vectors is the *non-dominated front* (or the *trade-off surface*), denoted by Z_N. A possible approach to solve a MOP consists of finding or approximating a minimal set of efficient solutions, *i.e.* one solution $x \in X_E$ for each non-dominated point $z \in Z_N$ such as $f(x) = z$ (in case multiple solutions map to the same non-dominated vector). Evolutionary algorithms are commonly used to this end as they naturally find multiple and well-spread non-dominated solutions in a single simulation run. The reader could refer to [3, 4] for more details about evolutionary multi-objective optimization.

2.2 The Bi-objective Ring Star Problem

The *Bi-objective Ring Star Problem* (B-RSP) can be described as follows. Let $G = (V, E, A)$ be a complete mixed graph where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertexes, $E = \{[v_i, v_j] | v_i, v_j \in V, i < j\}$ is a set of edges, and $A = \{(v_i, v_j) | v_i, v_j \in V\}$ is a set of arcs. Vertex v_1 is the depot. To each edge $[v_i, v_j] \in E$ we assign a non-negative *ring cost* c_{ij}, and to each arc $(v_i, v_j) \in A$ we assign a non-negative *assignment cost* d_{ij}. The B-RSP consists of locating a simple cycle through a subset of nodes $V' \subset V$ (with $v_1 \in V'$) while (*i*) minimizing the sum of the ring costs related to all edges that belong to the cycle, and (*ii*) minimizing the sum of the assignment costs of arcs directed from every non-visited node to a visited one so that the associated cost is minimum. An example of solution is given in Figure 1, where solid lines represent edges that belong to the ring and dashed lines represent arcs of the assignments.

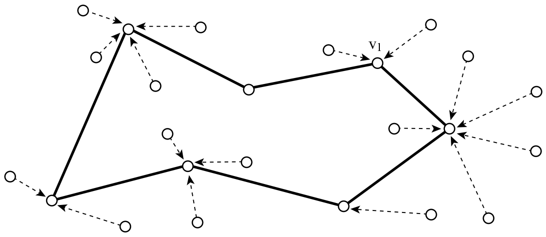

Fig. 1. An example of solution for the ring star problem

The first objective is called the *ring cost* and is defined as:

$$\sum_{[v_i, v_j] \in E} c_{ij} b_{ij} \ , \tag{1}$$

where b_{ij} is a binary variable equal to 1 if and only if the edge $[v_i, v_j]$ belongs to the cycle. The second objective, the *assignment cost*, can be computed as follows:

$$\sum_{v_i \in V \setminus V'} \min_{v_j \in V'} d_{ij} \ . \tag{2}$$

Let us remark that these two objectives are comparable only if we assume that the ring cost and the assignment cost are commensurate one to another, what is rarely the case in practice. Furthermore, the fact of privileging a cost compared to the other is closely related to the decision-maker preferences. However, the B-RSP is an NP-hard combinatorial problem since the particular case of visiting the whole set of nodes is equivalent to a traditional Traveling Salesman Problem (TSP).

2.3 GENIUS, a TSP Heuristic

A specificity of the B-RSP is that many TSP generally need to be solved. An effective TSP heuristic method is thus rather appreciated in order to improve the ring cost of a solution. There exists a large range of heuristics that are devoted to the TSP. One of them is GENIUS, proposed by Gendreau et al. [6]. Briefly, GENIUS contains a tour construction phase, called GENI, and a postoptimization phase, called US. Starting with three arbitrary nodes, GENI inserts, at each iteration, an unrouted node between two of its p closest neighbors on the partially constructed tour, where p is a user-controlled parameter. When inserting the vertex, GENI also performs a local reoptimization of the tour. Once a complete tour has been built, the US postoptimization procedure is repeatedly applied to the tour until no further improvement is possible. During this procedure, nodes are successively removed from the tour, and then reinserted, according to the same rules used in the tour construction phase. The use of GENIUS can be seen as a black-box mechanism integrated into the hybrid metaheuristic presented in the next section, and could practically be replaced by another TSP heuristic.

3 A Hybrid Metaheuristic for the Bi-objective Ring Star Problem

The main process of the Hybrid Metaheuristic (HM) proposed in the paper to solve the B-RSP consists of an elitist multi-objective Evolutionary Algorithm (EA). A first hybridization mechanism arises at the very beginning of the HM, as the initial population is built thanks to a problem-specific heuristic. This initial population is used as a starting point of the EA, so that both methods cooperate in pipeline way. Second, an additional hybridization scheme conditionally appears at every generation of the EA, where the ring cost of each population member is attempted to be improved thanks to a TSP heuristic. Finally, the EA is itself hybrid, as it is divided into two different phases. Those ones differ the one from the other at the selection and the replacement steps of the EA. During both phases, a secondary population, the so-called archive, is used to store every potentially efficient solutions found so far. The first phase is compound of an elitist selection step where parent individuals are all selected from the archive only. The replacement step is a generational one, *i.e.* the parent population is replaced by the offspring one. This phase corresponds to the *Simple*

Elitist Evolutionary Algorithm (SEEA) introduced in [15]. The main particularity of SEEA is that no fitness assignment scheme is required, the population being the only problem-independent parameter. The second phase is founded on the *Indicator-Based Evolutionary Algorithm* proposed by Zitzler and Künzli [21]. The fitness assignment scheme of IBEA is based on a pairwise comparison of population items by using a binary quality indicator I. Several indicators can be used for such a purpose [21], and we here choose to use the binary additive ϵ-indicator ($I_{\epsilon+}$) proposed in [23]. $\mathrm{I}_{\epsilon+}$ gives the minimum factor by which a non-dominated set A has to be translated in the objective space to weakly dominate a non-dominated set B. The selection scheme for reproduction is a binary tournament between randomly chosen individuals. The replacement strategy consists of deleting, one-by-one, the worst individuals, and in updating the fitness values of the remaining solutions each time there is a deletion; this is continued until the required population size is reached. The first phase of the EA will allow to find a rough approximation of the efficient set in a very short amount of time whereas the second phase will rather be devoted to improve this set in a more intensive way. The transition from Phase 1 to Phase 2 will occur as soon as the archive of non-dominated solutions does not improve enough with regards to the search scenario. The main steps of our HM are the following ones:

1. **Initialization.** Generate an initial population P of size N (see Section 3.2); generate an efficient set approximation A with the non-dominated individuals contained in P; create an empty offspring population P'.
2. **Selection.** Repeat until $|P'| = N$:
   (*Phase* 1*)* Randomly select an individual from A and add it to the offspring population P'.
   (*Phase* 2*)* Select an individual thanks to a binary tournament selection on P and add it to the offspring population P'.
3. **Recombination.** Apply a recombination operator to pairs of individuals contained in P' with a given probability p_r (see Section 3.3).
4. **Mutation.** Apply a mutation operator to individuals contained in P' with a given probability p_m (see Section 3.4).
5. **Fitness assignment.**
   (*Phase* 1*)* $\emptyset$.
   (*Phase* 2*)* Calculate fitness values of any individual x contained in $P \cup P'$; *i.e.* $F(x) \leftarrow \sum_{x' \in (P \cup P') \setminus \{x\}} -e^{-I(\{x'\},\{x\})/\kappa}$, where $\kappa > 0$ is a scaling factor.
6. **Replacement.**
   (*Phase* 1*)* $P \leftarrow P'$; $P' \leftarrow \emptyset$.
   (*Phase* 2*)* $P \leftarrow P \cup P'$; $P' \leftarrow \emptyset$. Iterate the following steps until the size of the population P does not exceed N:
   - Choose an individual $x^\star \in P$ with the smallest fitness value; *i.e.* $F(x^\star) \leq F(x)$ for all $x \in P$.
   - Remove $x^\star$ from P.
   - Update the fitness values of the individuals remaining in P; *i.e.* $F(x) \leftarrow F(x) + e^{-I(\{x^\star\},\{x\})/\kappa}$ for all $x \in P$.

7. **Elitism.** $A \leftarrow$ non-dominated individuals of $A \cup P$.
8. **Improvement.** If a given condition is satisfied, apply an improvement procedure on any individual contained in P (see Section 3.5).
9. **Termination.** If a stopping criteria is satisfied return A, else go to Step 2.

The principle of the HM is illustrated in Figure 2. According to the taxonomy of hybrid metaheuristics proposed in [20], the HM proposed in this paper can be classified on the *high-level relay hybrid* class, where self-contained heuristics are executed in sequence. The problem-specific components are explained in details below.

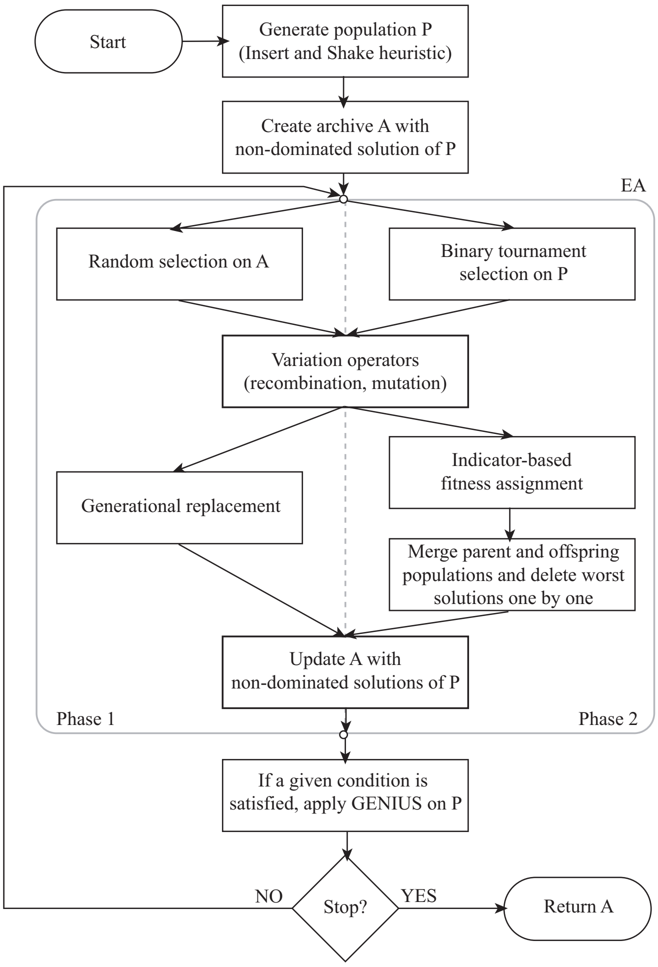
Start
Generate population P (Insert and Shake heuristic)
Create archive A with non-dominated solution of P
EA
Random selection on A
Binary tournament selection on P
Variation operators (recombination, mutation)
Generational replacement
Indicator-based fitness assignment
Merge parent and offspring populations and delete worst solutions one by one
Update A with non-dominated solutions of P
Phase 1
Phase 2
If a given condition is satisfied, apply GENIUS on P
NO
Stop?
YES
Return A

Fig. 2. Flowchart of the Hybrid Metaheuristic (HM)

3.1 Solution Encoding

The representation of a B-RSP solution is based on the random keys mechanism proposed by Bean [1]. A *random key* $k_i \in [0, 1[$ is assigned to every node v_i that belongs to the ring. A special value is assigned to unvisited nodes. Thus, the ring route associated to a solution corresponds to the nodes read according to their random keys in the increasing order; *i.e.* if $k_i < k_j$, then v_j comes after v_i. A possible representation for the cycle (v_1,v_7,v_4,v_9,v_2,v_6) is given in Figure 3. Nodes v_3,v_5,v_8 and v_{10} are assigned to a visited node in such a way that the associated assignment cost is minimum.

Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Random key	0	0.7	-	0.3	-	0.8	0.2	-	0.5	-

Fig. 3. A RSP solution represented by random keys

3.2 Population Initialization

An initial population of N individuals is built by means of repeatedly solving a mono-objective problem closely linked to the B-RSP. This problem, that will be denoted by Ring Cost Constrained RSP (RCC-RSP), consists of minimizing the assignment cost only, while satisfying an upper bound on the ring cost. It is obtained by removing the ring cost from the set of objective functions of the B-RSP, and by adding a new constraint stipulating that the ring cost cannot exceed a given limit $c_{\max}$. A search mechanism is iterated with distinct $c_{\max}$ values such that the set of resulting problems corresponds to different part of the objective space.

In order to approximately solve a given RCC-RSP, we use the *Insert and Shake Heuristic* (ISH), initially proposed by Gendreau et al. [7] for a single-objective routing problem called the selective TSP. This method combines a TSP tour extension heuristic described in Rosenkrantz et al. [19] and the GENIUS procedure described in Section 2.3. ISH gradually extends a tour T until no other node can be added without violating a given ring cost limit $c_{\max}$. At a given step, the non-visited node to be inserted in T is chosen so that the ratio between its current assignment cost and the increment on the global ring cost after its insertion is minimum. Then, GENIUS is applied in an attempt to obtain a better cycle on the nodes in T. If GENIUS fails, the procedure terminates. Otherwise, more node insertions are attempted and the process is repeated. The steps to build an initial population of solutions are the following ones:

1. GENIUS is applied to find a cycle containing all nodes in V, and this solution is included in the population. Let $c^\star$ be the ring cost of this solution. Set $\alpha \leftarrow \frac{c^\star}{N-1}$, $c_{\max} \leftarrow c^\star - \alpha$ and $i \leftarrow 1$.
2. If $i > N$, stop. Otherwise, generate a RCC-RSP solution by means of ISH and insert this solution into the population.

3. Set $c_{\max} \leftarrow c_{\max} - \alpha$, and $i \leftarrow i + 1$. Go to Step 2.

This initialization strategy is illustrated in Figure 4. Thanks to this heuristic, the starting set of solutions will already be both (*i*) quite efficient, and (*ii*) well-spread on the objective space.

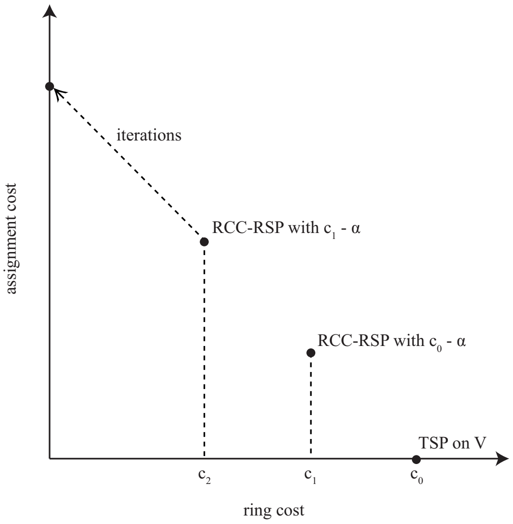

Fig. 4. Illustration of the population initialization heuristic

3.3 Recombination Operator

The recombination operator is a quadratic crossover closely related to the one proposed in [18]. Two randomly selected solutions x_1 and x_2 are first divided according to a particular position. Then, the first part of x_1 is combined with the second part of x_2 to build a first offspring, and the first part of x_2 is combined with the second part of x_1 to build a second offspring. Every node retains its random key so that it enables an easy reconstruction of the new individuals. Thanks to the random keys encoding mechanism, solutions having different ring sizes can easily be recombined, even if the initial ring structures are generally broken in the offspring solutions.

3.4 Mutation Operator

The mutation operator designed for the problem under consideration consists of the following strategy. A node $v^\star \in V \setminus \{v_1\}$ is selected at random. Therefore, two cases may arise. First, if $v^\star$ belongs to the ring, it is removed and then belongs to the set of unvisited nodes. Second, if $v^\star$ does not belong to the cycle, it is added. The position to insert $v^\star$ is chosen so that the increment on the ring cost is minimum.

3.5 Improvement Procedure

The improvement procedure consists of improving the ring cost of the current population by applying GENIUS on any of its members. The main issue is now to determine when this heuristic might start in order to find the good trade-off between the efficiency and the effectiveness of GENIUS. To do so, we decide to launch it only if no more than n solutions per iteration have been included in the archive during the last M consecutive generations. Hence, we hope that the improved population will produce new non-dominated solutions and will help to build more interesting individuals in the future steps of the HM.

4 Computational Experiments

In order to assess the effectiveness of our method, we will measure its performance in comparison to the ACS method proposed in [15]. The latter is an auto-adaptive method based on a simple elitist evolutionary algorithm and a population-based local search. It has been shown to be particularly efficient to solve the RSP as a bi-objective problem. To quantity the impact of GENIUS on our HM, we also implemented a more basic version in which GENIUS is not involved, neither in the improvement step nor in the initialization step. This other hybrid metaheuristic will be denoted by HM2 in the remainder of the paper. All the algorithms have been implemented under the ParadisEO-MOEO library[4] [14] and share the same base components for a fair comparison between them. Computational runs were performed under Linux on an Intel Core 2 Duo 6600 (2 × 2.40 GHz) machine, with 2 GB RAM.

4.1 Performance Assessment

Experiments have been conducted on a set of benchmark test instances taken from the TSPLIB[5] [17]. These instances involve between 51 and 264 nodes. The number at the end of an instance's name represents the number of nodes for the instance under consideration. Let l_{ij} denote the distance between two nodes v_i and v_j of a TSPLIB file. Then, the ring cost c_{ij} and the assignment cost d_{ij} have both been set to l_{ij} for every pair of nodes v_i and v_j.

For each search method, a set of 20 runs per instance has been performed. In order to evaluate the quality of the non-dominated front approximations, we follow the protocol given by Knowles et al. [11]. For a given instance, we first compute a reference set $Z_N^\star$ of non-dominated points extracted from the union of all the fronts we obtained during our experiments and the best non-dominated set taken from [15][6]. Second, we define a point $z^{max} = (z_1^{max}, z_2^{max})$, where z_1^{max} (respectively z_2^{max}) denotes the upper bound of the first (respectively second) objective in the whole non-dominated front approximations. Then, to

[4] ParadisEO is available at http://paradiseo.gforge.inria.fr.

[5] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

[6] These results are available at http://www.lifl.fr/~liefooga/rsp/.

measure the quality of an output set A in comparison to $Z^{\star}_{N}$, we compute the difference between these two sets by using the unary hypervolume metric [22], $(1.05 \times z_1^{max}, 1.05 \times z_2^{max})$ being the reference point. The hypervolume difference indicator (I_H^-) computes the portion of the objective space that is weakly dominated by $Z^{\star}_{N}$ and not by A. The more this measure is close to 0, the better is the approximation A. Furthermore, we also consider one of the ϵ-indicators proposed in [23]. The unary additive ϵ-indicator ($\mathrm{I}^1_{\epsilon+}$) gives the minimum factor by which an approximation A has to be translated in the objective space to weakly dominate the reference set $Z^{\star}_{N}$. As a consequence, for each test instance, we obtain 20 I_H^- measures and 20 $\mathrm{I}_{\epsilon+}$ measures, corresponding to the 20 runs, per algorithm. As suggested by Knowles et al. [11], once all these values are computed, we perform a statistical analysis on pairs of optimization methods for a comparison on a specific test instance. To this end, we use the Mann-Whitney statistical test as described in [11], with a p-value lower than 5%. Hence, for a specific test instance, and according to the p-value and to the metric under consideration, this statistical test reveals if the sample of approximation sets obtained by a given search method is significantly better than the one obtained by another search method, or if there is no significant difference between both. Note that all the performance assessment procedures have been achieved using the performance assessment tool suite provided in PISA[7] [2].

4.2 Parameter Setting

For each investigated metaheuristic, the search process stops after a certain amount of run time. As shown in Table 1, this stopping criteria has been arbitrary set according to the size of the problem instance to be solved. Next, the population size N is set to 100; the recombination probability p_r is set to 0.25 and the mutation probability p_m is set to 1.0. Following [21], the scaling factor κ is set to 0.05. The improvement procedure of HM is launched only if the number of elements received by the archive is less than 1.0% of its current size for $|V|$ consecutive iterations, where $|V|$ is the number of nodes for the instance under consideration. Finally, the GENIUS parameter p is set to 7.

Table 1. Stopping criteria: running time

Instance	Running time	Instance	Running time
eil51	20"	*kroA150*	10'
st70	1'	*kroA200*	20'
kroA100	2'	*pr264*	30'
bier127	5'		

[7] The package is available at http://www.tik.ee.ethz.ch/pisa/assessment.html.

4.3 Results and Discussion

First of all, note that we initially experimented some algorithm versions where only the first or the second phase of the EA is involved, with and without GENIUS. But the resulting metaheuristics turned out to be significantly outperformed by HM and HM2. The comparison of results obtained by HM, HM2 and ACS are presented in Table 2. According to both indicators (I_H^- and $I_{\epsilon+}$), HM is statistically better than any other search methods on every instance we investigated. Besides, the difference between HM2 and ACS is often not significant according to the I_H^- metric, whereas ACS generally outperforms HM2 according to the $I_{\epsilon+}$ metric. The only instance for which HM2 performs statistically higher than ACS is the *bier127* instance, where it obtains better values for both metrics. In order to study on which part of the trade-off surface the differences between HM, HM2 and ACS appear, examples of empirical attainment functions [5] are given in Figure 5 and Figure 6 for the *bier127* instance. They represent the limit of the objective space that is attained by at least 90% of the runs for every search method. For the instance under consideration, we can see that HM seems to be more capable of finding solutions having both a large number of visited nodes and a good ring cost. Thus, the superiority of HM relatively to HM2 reveals the benefit of integrating a TSP heuristic, here symbolized by GENIUS, into our EA for the problem to be solved. Moreover, despite its relative simplicity in comparison to ACS, the HM is quite effective to solve the B-RSP, especially to find solutions having a low ring cost. This indicates that the hybridization scheme largely improves the method and reveals that the HM introduced in this paper outperforms the metaheuristics proposed so far to solve the RSP as a bi-objective optimization problem.

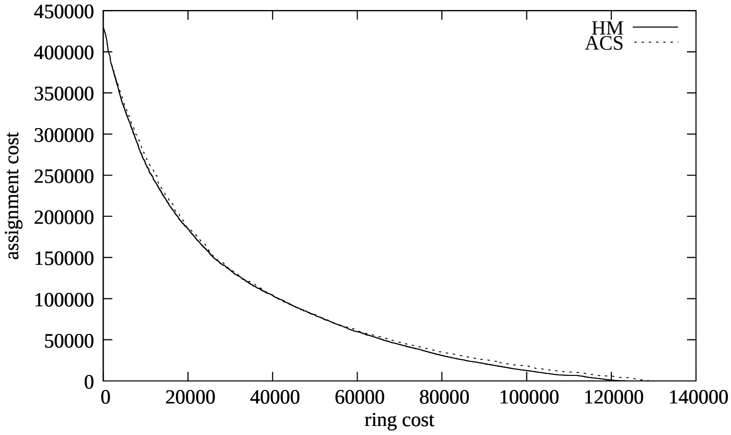

Fig. 5. 90%-attainment surface plot obtained by the approximation sets found by HM and ACS [15] for the *bier127* test instance

Table 2. Comparison of HM, HM2 and ACS [15] according to the I_H^- and the $I_{\epsilon+}$ metrics by using a Mann-Whitney statistical test with a p-value of 5%. For the metric under consideration, either the results of the algorithm located at a specific row are significantly better than those of the algorithm located at a specific column ($\succ$), either they are worse ($\prec$), or there is no significant difference between both ($\equiv$).

			I_H^-			$I_{\epsilon+}$	
		HM	HM2	ACS	HM	HM2	ACS
eil51	HM	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	HM2	$\prec$	-	$\equiv$	$\prec$	-	$\equiv$
	ACS	$\prec$	$\equiv$	-	$\prec$	$\equiv$	-
st70	HM	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	HM2	$\prec$	-	$\equiv$	$\prec$	-	$\equiv$
	ACS	$\prec$	$\equiv$	-	$\prec$	$\equiv$	-
kroA100	HM	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	HM2	$\prec$	-	$\prec$	$\prec$	-	$\prec$
	ACS	$\prec$	$\succ$	-	$\prec$	$\succ$	-
bier127	HM	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	HM2	$\prec$	-	$\succ$	$\prec$	-	$\succ$
	ACS	$\prec$	$\prec$	-	$\prec$	$\prec$	-
kroA150	HM	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	HM2	$\prec$	-	$\equiv$	$\prec$	-	$\prec$
	ACS	$\prec$	$\equiv$	-	$\prec$	$\succ$	-
kroA200	HM	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	HM2	$\prec$	-	$\equiv$	$\prec$	-	$\prec$
	ACS	$\prec$	$\equiv$	-	$\prec$	$\succ$	-
pr264	HM	-	$\succ$	$\succ$	-	$\succ$	$\succ$
	HM2	$\prec$	-	$\equiv$	$\prec$	-	$\prec$
	ACS	$\prec$	$\equiv$	-	$\prec$	$\succ$	-

5 Conclusion and Perspectives

In this paper, a new hybrid metaheuristic has been proposed to approximate the efficient set of a multi-objective routing problem called the bi-objective ring star problem. This problem is commonly investigated in a single-objective way, either where both objectives are aggregated, or where one objective is regarded as a constraint. However, within the frame of the ring star problem, many traveling salesman problems generally need to be solved. The purpose of the hybrid metaheuristic proposed here is then to integrate a heuristic algorithm for the traveling salesman problem, namely GENIUS, into a multi-objective evolutionary algorithm to solve the bi-objective ring star problem as a whole. The hybrid search method starts with a problem-specific heuristic to generate an initial set of solutions, and continues with a two-phase elitist evolutionary algorithm hybridized to the GENIUS heuristic. The latter is launched to intensify the search in an auto-adaptive manner, according to the convergence scenario of the main process. Experiments were conducted on a set of benchmark test instances, and validated the contribution of the traveling salesman problem heuristic into the

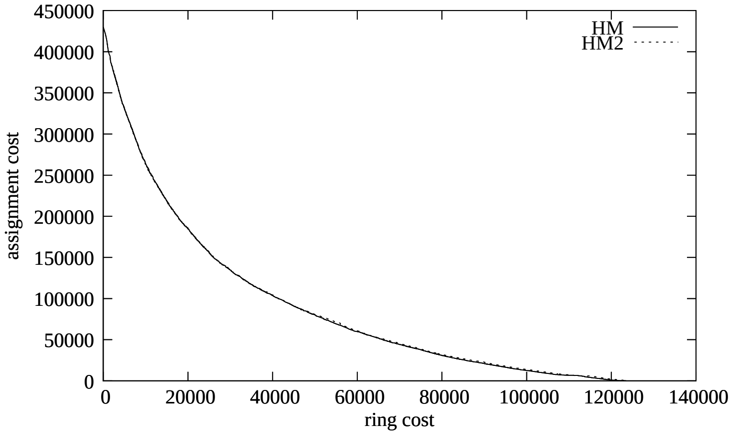

Fig. 6. 90%-attainment surface plot obtained by the approximation sets found by HM and HM2 for the *bier127* test instance

hybrid method. They also reveal that the metaheuristic proposed in the paper largely outperforms our previous investigations for resolving the bi-objective ring star problem. As a next step, we will experiment other strategies to scale the GENIUS application factor in order to study the influence of this parameter on the global performance of our method. We will also try to replace GENIUS by other kinds of traveling salesman problem heuristics, or even exact methods, within our hybrid metaheuristic to assert the genericity of our method. Moreover, it could be interesting to design a more universal model of hybridization to solve multi-objective optimization problems, where both problem-specific and meta methods could be integrated.

References

1. J. Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.
2. S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA — a platform and programming language independent interface for search algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science (LNCS)*, pages 494–508, Faro, Portugal, 2003. Springer-Verlag.
3. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers, Boston, MA, USA, 2002.
4. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons, Chichester, UK, 2001.
5. C. M. Fonseca, V. Grunert da Fonseca, and L. Paquete. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Conference on*

Evolutionary Multi-Criterion Optimization (EMO 2005), volume 3410 of *Lecture Notes in Computer Science (LNCS)*, pages 250–264, Guanajuato, Mexico, 2005. Springer-Verlag.

6. M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6):1086–1094, 1992.
7. M. Gendreau, G. Laporte, and F. Semet. A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, 106:539–545, 1998.
8. N. Jozefowiez, F. Glover, and M. Laguna. Multi-objective meta-heuristics for the traveling salesman problem with profits. *Journal of Mathematical Modelling and Algorithms*, 7(2):177–195, 2008.
9. N. Jozefowiez, F. Semet, and E.-G. Talbi. The bi-objective covering tour problem. *Computers and Operations Research*, 34(7):1929–1942, 2007.
10. N. Jozefowiez, F. Semet, and E.-G. Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309, 2008.
11. J. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2006. (revised version).
12. M. Labbé, G. Laporte, I. Rodríguez Martín, and J. J. Salazar González. The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, 43:177–189, 2004.
13. M. Labbé, G. Laporte, I. Rodríguez Martín, and J. J. Salazar González. Locating median cycles in networks. *European Journal of Operational Research*, 160(2):457–470, 2005.
14. A. Liefooghe, M. Basseur, L. Jourdan, and E.-G. Talbi. ParadisEO-MOEO: A framework for evolutionary multi-objective optimization. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *Lecture Notes in Computer Science (LNCS)*, pages 386–400, Matsushima, Japan, 2007. Springer-Verlag.
15. A. Liefooghe, L. Jourdan, and E.-G. Talbi. Metaheuristics and their hybridization to solve the bi-objective ring star problem: a comparative study. Working paper RR-6515, Institut National de Recherche en Informatique et Automatique (INRIA), 2008.
16. J. A. Moreno Pérez, J. M. Moreno-Vega, and I. Rodríguez Martín. Variable neighborhood tabu search and its application to the median cycle problem. *European Journal of Operations Research*, 151(2):365–378, 2003.
17. G. Reinelt. TSPLIB – A traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991.
18. J. Renaud, F. F. Boctor, and G. Laporte. Efficient heuristics for median cycle problems. *Journal of the Operational Research Society*, 55(2):179–186, 2004.
19. D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.
20. E.-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(2):541–564, 2002.
21. E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *Lecture Notes in Computer Science (LNCS)*, pages 832–842, Birmingham, UK, 2004. Springer-Verlag.

22. 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.
23. E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.