

Metaheuristics and Their Hybridization to Solve the Bi-objective Ring Star Problem: a Comparative Study

Arnaud Liefoghe, Laetitia Jourdan, El-Ghazali Talbi

► **To cite this version:**

Arnaud Liefoghe, Laetitia Jourdan, El-Ghazali Talbi. Metaheuristics and Their Hybridization to Solve the Bi-objective Ring Star Problem: a Comparative Study. [Research Report] RR-6515, INRIA. 2008. inria-00275659v2

HAL Id: inria-00275659

<https://hal.inria.fr/inria-00275659v2>

Submitted on 28 Apr 2008

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.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Metaheuristics and Their Hybridization to
Solve the Bi-objective Ring Star Problem:
a Comparative Study*

Arnaud Liefooghe — Laetitia Jourdan — El-Ghazali Talbi

N° 6515

April 2008

Thème NUM

 *rapport
de recherche*

Metaheuristics and Their Hybridization to Solve the Bi-objective Ring Star Problem: a Comparative Study

Arnaud Liefoghe , Laetitia Jourdan , El-Ghazali Talbi

Thème NUM — Systèmes numériques
Équipe-Projet Dolphin

Rapport de recherche n° 6515 — April 2008 — 45 pages

Abstract: This paper presents and experiments approaches to solve a new bi-objective routing problem called the ring star problem. It consists of locating a simple cycle through a subset of nodes of a graph while optimizing two kinds of cost. The first objective is the minimization of a ring cost that is related to the length of the cycle. The second one is the minimization of an assignment cost from non-visited nodes to visited ones. In spite of its obvious bi-objective formulation, this problem has always been investigated in a single-objective way. To tackle the bi-objective ring star problem, we first investigate different stand-alone search methods. Then, we propose two cooperative strategies that combines two multiple objective metaheuristics: an elitist evolutionary algorithm and a population-based local search. We apply this new hybrid approaches to well-known benchmark test instances and demonstrate their effectiveness in comparison to non-hybrid algorithms and to state-of-the-art methods.

Key-words: Multi-objective combinatorial optimization, evolutionary algorithms, local exploration, hybrid approaches, routing.

Arnaud.Liefoghe@lifl.fr, Laetitia.Jourdan@lifl.fr, El-Ghazali.Talbi@lifl.fr

Metaheuristiques et leur hybridation pour résoudre le Ring Star Problem bi-objectif: une étude comparative

Résumé : Ce document présente et expérimente différentes approches pour la résolution d'un nouveau problème de routage bi-objectif appelé le ring star problem. Il consiste à localiser un cycle simple dans un sous-ensemble de noeuds d'un graphe en optimisant deux types de coût. Le premier objectif est la minimisation d'un coût d'anneau qui est lié à la longueur du cycle. Le deuxième est la minimisation d'un coût d'assignement depuis les noeuds non visités vers les noeuds visités. En dépit de sa formulation bi-objectif évidente, ce problème a toujours été étudié de façon mono-objectif. Pour attaquer le ring star problem bi-objectif, nous étudions d'abord différentes méthodes de recherche. Ensuite, nous proposons deux stratégies coopératives qui combinent deux métaheuristiques multi-objectif : un algorithme évolutionnaire élitiste et une recherche locale à base de population. Nous appliquons ces approches hybrides des instances de test réputées et démontrons leur efficacité par rapport aux algorithmes non-hybrides et à des méthodes classiques de la littérature.

Mots-clés : Optimisation combinatoire multi-objectif, algorithmes évolutionnaires, exploration locale, approches hybrides, routage.

1 Introduction

First introduced by Labbé et al. [33], the *Ring Star Problem* (RSP), also known as *Median Cycle Problem* (MCP), is a recent routing problem. It aims to locate a simple cycle (the ring) containing a subset of nodes of a graph (including the depot), while nodes that do not belong to the ring must be connected to a visited one. Two costs are considered here, what naturally leads to a bi-criteria formulation. The first one is the ring cost associated to the ring itself and the second one is the assignment cost associated to the arcs directed from non-visited nodes to visited ones. Even so, the RSP has always been investigated in a single-objective way; either where both criterion are combined, or where a criteria is treated as a constraint. Thus, despite its numerous industrial interests, this is the first time that such a problem is regarded as a multi-objective one, probably due to its complexity. Indeed, it is highly combinatorial as, once is decided which nodes have to be visited or not, a traveling salesman problem is still to be solved.

Recently, we studied the effectiveness of existing metaheuristics for solving the bi-objective RSP [40]. In this paper, we provide a deeper analysis of this work, and propose new stand-alone and hybrid approaches. As an initial step, we investigate four metaheuristics to approximate the set of efficient solutions for the problem under consideration. First, IBMOLS, a population-based local search recently proposed in [3], is designed with a variable neighborhood. Next, we present steady-state variations of two well-known multi-objective search methods, namely IBEA [56] and NSGA-II [14]. Last, we propose a new simple elitist evolutionary algorithm, SEEA, able to solve any kind of multi-objective combinatorial optimization problems. We compare all these metaheuristics to each other on state-of-the-art benchmarks test instances. As a second step, we propose two new cooperative strategies between the local search method and SEEA, based on a periodic and on an auto-adaptive scheme, that try to benefit of the advantages of each method it is compound of. The performance of these hybrid metaheuristics is shown in comparison to non-hybrid approaches.

The paper is organized as follows. Section 2 is devoted to the bi-objective RSP. Then, four metaheuristics for multi-objective optimization are introduced in Section 3. A general presentation of these methods and their application to the RSP is followed by a parameter analysis and a comparative case study. In Section 4, we propose new cooperation schemes to solve multi-objective optimization problems. We experiment the resulting search methods on the bi-objective RSP and we compare the obtained computational results to the previous ones. At last, conclusions and perspectives are drawn in the last section.

2 The Bi-objective Ring Star Problem

In this section, we first present some basic concepts, notation and definitions related to multi-objective optimization. Next, we provide a formulation of the Ring Star Problem (RSP) as a bi-objective problem. Finally, we survey the literature relating to the RSP and discuss its industrial concerns.

2.1 Multi-objective Optimization

A *Multi-objective Optimization Problem* (MOP) aims to optimize a set of $n \geq 2$ objective functions f_1, f_2, \dots, f_n simultaneously. Each objective function can be either minimized or maximized. Let X denote the set of feasible solutions in the *decision space*, and Z the set of feasible points in the *objective space*. A decision vector $x = (x_1, x_2, \dots, x_k)$ is represented by a vector of k decision variables. In the case of a *Multi-objective Combinatorial Optimization Problem* (MCOP), a decision vector $x \in X$ has a finite set of possible values. 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 $z = f(x) = (f_1(x), f_2(x), \dots, f_n(x))$. We will assume, throughout the paper, that objective values are normalized. To achieve this, the minimum and the maximum value of each objective function are used in order to adaptively replace each objective function by its corresponding normalized function, so that its values lie in the interval $[0, 1]$. Without loss of generality, we assume that $Z \subseteq \mathbb{R}^n$ and that all n objective functions have to be minimized. Therefore, a MOP can be formulated as follows:

$$(MOP) = \begin{cases} \text{'min'} & f(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{subject to} & x \in X \end{cases} \quad (1)$$

Definition 1 An objective vector $z \in Z$ weakly dominates another objective vector $z' \in Z$ if and only if $\forall i \in \{1, 2, \dots, n\}$, $z_i \leq z'_i$.

Definition 2 An objective vector $z \in Z$ dominates¹ another objective vector $z' \in Z$ if and only if $\forall i \in \{1, 2, \dots, n\}$, $z_i \leq z'_i$ and $\exists j \in \{1, 2, \dots, n\}$ 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 *Pareto optimal*, *non-dominated*) if its mapping in the objective space results in a non-dominated point. The set of all efficient solutions is the *efficient* (or *Pareto optimal*) *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 common approach in solving MOPs is to find or to approximate the minimal set of efficient solutions, *i.e.* a solution $x \in X_E$ for each non-dominated vector $z \in Z_N$ such as $f(x) = z$ (in case there exists multiple solutions mapping to the same non-dominated point). But, generating the entire set of Pareto optimal solutions is usually infeasible due to the complexity of the underlying problem or to the large number of optima. Therefore, the overall goal is often to identify a good approximation of it. Population-based metaheuristics are commonly used to this end as they naturally find multiple and well-spread non-dominated solutions in a single simulation run. A reasonable basic introduction to multi-objective optimization can be found in [13]. But, the interested reader could refer to [9, 12] for more details about evolutionary multi-objective optimization and to [17] for more details about multi-objective combinatorial optimization.

¹We will also say that a decision vector $x \in X$ *dominates* a decision vector $x' \in X$ if $f(x)$ dominates $f(x')$.

2.2 Problem Definition

The *Ring Star Problem* (RSP) can be described as follows. Let $G = (V, E, A)$ be a complete mixed graph where $V = \{v_1, v_2, \dots, v_n\}$ 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 $[v_i, v_j]$ we assign a non-negative *ring cost* c_{ij} , and to each arc (v_i, v_j) we assign a non-negative *assignment cost* d_{ij} . The 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.

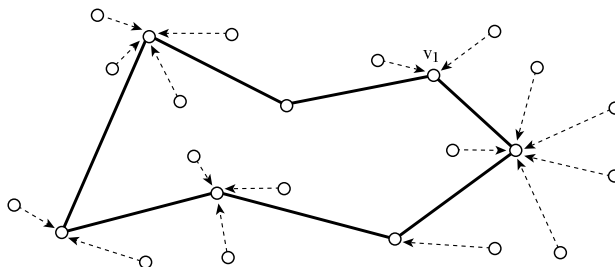


Figure 1: An example of solution for the ring star problem.

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

$$\sum_{[v_i, v_j] \in E} c_{ij} x_{ij} \quad (2)$$

where x_{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:

$$\sum_{v_i \in V \setminus V'} \min_{v_j \in V'} d_{ij} \quad (3)$$

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 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.

2.3 Related Works

The RSP belongs to the class of location-allocation problems aiming at locating structures in a graph (see [32] for a review). It has initially been formulated by Labbé et al. [33] in two different ways. In the first formulation (denoted MCP1 in [33] and more often called ‘ring star problem’), the sum of both objectives is to be optimized. In the second formulation (MCP2 [33], usually

called ‘median cycle problem’), the ring cost is to be minimized, while the assignment cost is bounded by a prefixed value. Even if it was not explicitly noticed by the original authors, these two kinds of formulation are commonly employed to convert a multi-objective problem into a single-objective one by using two well-known scalar approaches; respectively an *aggregation method* and an *epsilon-constraint method*². The first formulation of the problem has been more widely studied in [34]. The authors used a branch-and-bound method and successfully solved TSPLIB and randomly generated instances involving up to 200 nodes in less than two hours. In [35], the same authors investigated the second formulation of the problem using a very close method. Finally, one or both versions of the problem have been heuristically solved by a variable neighborhood tabu search [44], an evolutionary algorithm [47], a multi-start greedy add heuristic [47], and a variable neighborhood tabu search hybridized with a greedy randomized adaptive search procedure [15].

A very similar problem, namely the *capacitated m-ring star problem*, has been introduced in [2]. It extends the first formulation by bounding the number of nodes that can be assigned to a ring using a prefixed parameter (the ring capacity). This problem has been exactly solved by a branch-and-bound method and experiments have been performed on random and real-world instances. Note that the same problem has also been tackled in [41] by a hybrid metaheuristic. Another variant of the RSP is the *Steiner Ring Star Problem*, where only optional nodes can be visited [38]. In addition, Beasley and Nascimento [6] introduced a *Single Vehicle Routing-Allocation Problem* (SVRAP) in which a non-visited node can be either assigned to the cycle or isolated. In this case, a corresponding isolated cost has to be minimized as well. Vogt et al. [53] proposed a tabu search to solve a single-objective problem resulting of this formulation.

As reflected in the survey of Jozefowicz et al. [27], an increasing number of multi-objective routing problems appeared in the literature in recent years. However, in spite of its numerous industrial applications (see subsection 2.4), the RSP has never been explicitly investigated in a multi-objective fashion. Nevertheless, Current and Schilling [11] defined two multiple criteria variants of a very similar problem: the *median tour problem* and the *maximal covering tour problem*. In both, one criterion is the minimization of the total length of the tour, while another one is the maximization of the access to the tour for non-visited nodes. To tackle these problems, the authors used a kind of lexicographic method. This approach consists of assigning a priority value to each objective function, and in solving the problem one objective after another. Also, Dorner et al. [16] recently formulated a tri-objective optimization problem of tour planning for mobile health care facilities in Senegal, closely related to the SVRAP [6] introduced earlier. A mobile facility has to visit a subset of nodes. Non-visited nodes are then assigned to their closest tour stop or are regarded as unable to reach a tour stop (within a predefined maximum distance). The objectives are (i) to minimize the ratio between medical working time and total working time, (ii) to minimize the average distance to the nearest tour stops and (iii) to maximize a coverage criterion. To do so, a Pareto ant colony optimization algorithm as well as two genetic algorithms (namely VEGA [48] and MOGA [20]) were designed to solve real-world instances.

²The reader is referred to [43] for more details about scalar methods for solving MOPs.

2.4 Industrial Interests

The RSP has a wide range of industrial interests, especially in telecommunications. Of course, practical applications may contain additional constraints. For instance, as noticed by Labbé et al. [34, 35], a ring-based network is designed to interconnect a set of hubs in the case of digital data services [54]. Some concentrators are installed on a subset of locations and are interconnected on a ring network (the Internet) while the remaining locations are assigned to this concentrators (the Intranet). Very close problems arise in rapid transit systems planning [34] or while designing optical networks [2]. Also, other kinds of applications appears in the postal collection or delivery routes design, where the distance between a customer and a collection point have to be reasonable. For instance, post-box location while taking both the collection cost and the user inconvenience into account has been studied in [31]. Besides, other applications closely related to the RSP are the location of circular shaped transportation infrastructure (such as metro lines or motorways), the location of recyclable garbage collection bins, and school bus routing. Finally, the routing of essential health care services, already investigated in [1, 16] among other authors, consists of a mobile clinic servicing an area without being able to visit every population nodes. Then, unvisited ones have to reach the nearest tour stop by their own to be medically treated.

3 Metaheuristics to Solve the Bi-objective Ring Star Problem

Four metaheuristics are proposed to tackle the bi-objective RSP: a variable neighborhood iterative population-based Local Search (LS) and three Evolutionary Algorithms (EAs). These algorithms are respectively steady-state variations of IBMOLS [3], IBEA [56] and NSGA-II [14], as well as a new search method, SEEA, introduced in this paper for the first time. IBMOLS and IBEA are both recent indicator-based metaheuristics, whereas NSGA-II is one of the most often used Pareto-based resolution approach. In this section, RSP-specific components are described after we presented the main characteristics of the LS and of the EAs.

3.1 A Multi-objective Local Search

Since they are easily adaptable to the multi-objective context, many of the search algorithms proposed to tackle MOPs are EAs. However, LS algorithms are known to be effective metaheuristics for solving real-world applications [8, 22]. Several neighborhood search methods to solve MOPs have been proposed in the literature. Most of them are based on a set of aggregations or scalarizations of the objective functions, what is the case, for instance, in [21, 23, 26, 51]. Pareto-based LS algorithms, like the one proposed by Paquete et al. [45], are more rare. An Indicator-Based Multi-Objective Local Search (IBMOLS for short) has recently been presented in [3]. IBMOLS is a generic population-based multi-objective LS dealing with a fixed population size. This allows to obtain a set of efficient solutions in a single run without specifying any mechanism to control the number of solutions during the search process. Moreover, IBMOLS

presents an alternative to aggregation- and Pareto-based multi-objective meta-heuristics. Indeed, as proposed in [56], it is assumed that the optimization goal is given in terms of a binary quality indicator I [58] which can be regarded as an extension of the Pareto dominance relation. A value $I(A, B)$ quantifies the difference in quality between two approximated efficient sets A and B . So, if R denotes a reference set (which can be the Pareto-optimal set X_E or any other set), the overall optimization goal can be formulated as:

$$\arg \min_{A \in \Omega} I(A, R) , \quad (4)$$

where Ω denotes the space of all efficient set approximations. As noted in [56], R does not have to be known, it is just required in the formalization of the optimization goal. Since R is fixed, I actually represents a unary function that assigns a real number reflecting the quality of each approximation set according to the optimization goal. If I is dominance preserving [56], $I(A, R)$ is minimum for $A = R$. One of the main advantages of indicator-based optimization is that no specific diversity preservation mechanism is generally required, according to the indicator being used.

The IBMOLS algorithm maintains a population P of size N . Then, it generates the neighborhood of a solution contained in P until a good solution is found (*i.e.* one that is better than at least one solution of P in terms of the indicator being used). By iterating this simple principle to every solution of P , we obtain a local search step. The whole local search procedure stops when the archive of potentially efficient solutions has not received any new item during a complete local search step. Moreover, as local search methods are usually performed in an iterative way, a population re-initialization scheme has to be designed after each local search. As noticed in [3], several strategies can be used within an Iterative IBMOLS (I-IBMOLS). Solutions can be re-initialized randomly, and crossover or random noise can be applied to solutions belonging to the efficient set approximation. The interested reader could refer to [3] for more details about IBMOLS and I-IBMOLS.

A beneficial feature of this LS is the low number of parameters that are required. In addition to the population size, the binary quality indicator to be used and the population re-initialization strategy (between each local search) are the only two other problem-independent parameters. Indeed, several quality indicators can be used within IBMOLS. Some examples can be found in [24, 28, 58]. One of them is the binary additive ϵ -indicator ($I_{\epsilon+}$) [56, 58], inspired by the concept of ϵ -dominance introduced by Laumanns et al. [36]. This indicator is particularly well-adapted to indicator-based search and seems to be efficient on different kinds of problems (see, for instance, [3, 56]). It is capable of obtaining both a well-converged and a well-diversified Pareto front approximation. $I_{\epsilon+}$ computes the minimum value by which a solution $x \in X$ can be translated in the objective space to weakly dominate another solution $x' \in X$. For a minimization problem, it is defined as follows:

$$I_{\epsilon+}(x, x') = \max_{i \in \{1, \dots, n\}} (f_i(x) - f_i(x')) . \quad (5)$$

Furthermore, to evaluate the quality of a solution $x \in X$ according to a whole population P and a binary quality indicator I , and then to compute the fitness value of x , different approaches exist. As proposed in [56], we will here consider

an additive technique that amplifies the influence of solutions mapping to dominating points over solutions mapping to dominated ones, what can be outlined as follows:

$$I(x, P \setminus \{x\}) = \sum_{x^* \in P \setminus \{x\}} -e^{-I(x^*, x)/\kappa}, \quad (6)$$

where $\kappa > 0$ is a scaling factor. However, the initial experiments were not satisfactory because the algorithm was not able to find the extreme points of the trade-off surface. As pointed out in [25], this is known to be one of the drawbacks of the ϵ -dominance relation, apparently due to the high convexity of the front. Indeed, the authors illustrate that one of the limitation of the ϵ -dominance is that the extreme points of the Pareto front are usually lost. To tackle this problem, we add a condition preventing the deletion of solutions corresponding to the extreme non-dominated vectors during the replacement step of IBMOLS. Besides, the population re-initialization scheme used between each local search procedure is based on random noise, such as in the basic simulated annealing algorithm [8]. This noise consists of multiple mutations applied to N different randomly chosen solutions contained in the archive of potentially efficient solutions. If the size of the archive is less than N , the population is filled with random solutions.

3.2 Multi-objective Evolutionary Algorithms

The first two multi-objective EAs designed to tackle the bi-objective RSP are variations of two state-of-the-art search methods, namely IBEA [56] and NSGA-II [14]. Some minor modifications have been carried out to improve the algorithms for the particular case of the addressed problem, for which a preliminary investigation revealed that the set of non-dominated points was, in general, very large. Finally, a generic approach to solve MOPs, called SEEA, is proposed in this paper for the first time and is presented in details.

3.2.1 IBEA

Introduced by Zitzler and Künzli [56], the *Indicator-Based Evolutionary Algorithm* (IBEA) is, like IBMOLS, an indicator-based metaheuristic. The fitness assignment scheme of this EA is based on a pairwise comparison of solutions contained in a population by using a binary quality indicator. As within IBMOLS, no diversity preservation technique is required, according to the indicator being used. The selection scheme for reproduction is a binary tournament between randomly chosen individuals. The replacement strategy is an environmental one that 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. Moreover, an archive stores solutions mapping to potentially non-dominated points in order to prevent their loss during the stochastic search process. However, in our case, and in contrast to the IBEA defined in [56], this archive is updated at each generation since the beginning of the EA, so that the output size is not necessarily less than or equal to the population size. Just like for the IBMOLS algorithm, the indicator used within IBEA is the additive ϵ -indicator; and the same mechanism has been used to prevent the loss of the extreme points on the trade-off surface. More information about this algorithm can be found in [56].

3.2.2 NSGA-II

At each generation of NSGA-II (the *Non-dominated Sorting Genetic Algorithm II* introduced by Deb et al. in [14]), the solutions contained in the population are ranked into several classes. Individuals mapping to vectors from the first front all belong to the best efficient set; individuals mapping to vectors from the second front all belong to the second best efficient set; and so on. Two values are computed for every solution of the population. The first one corresponds to the *rank* the corresponding solution belongs to, and represents the quality of the solution in terms of convergence. The second one, the *crowding distance*, consists of estimating the density of solutions surrounding a particular point of the objective space, and represents the quality of the solution in terms of diversity. A solution is said to be better than another if it has a best rank value, or in case of equality, if it has the best crowding distance. The selection strategy is a deterministic tournament between two random solutions. At the replacement step, only the best individuals survive, with respect to the population size. Likewise, an external population is added to the steady-state NSGA-II in order to store every potentially efficient solution found during the search. The reader is referred to [14] for more details about NSGA-II.

3.2.3 SEEA

If evaluating a solution in the objective space is not too much time consuming (what is the case for our problem), computing fitness values and diversity information are generally the most computationally expensive steps of a multi-objective EA. Based on this observation, we propose here a simple search method for which none of these phases is required. The resulting EA, called *Simple Elitist Evolutionary Algorithm* (SEEA for short), is detailed in Algorithm 1. An archive of potentially efficient solutions is updated at each generation, and the individuals contained in the main population are generated by applying variation operators to randomly chosen archive members. The replacement step is a generational one, *i.e.* the parent population is replaced by the offspring one. Note that the initial population can, for instance, be filled with random solutions. Then, as proposed in [57] among other authors, the archive is not only used as an external storage, but it is integrated into the optimization process during the selection phase of the EA, what is called *elitism*. The use of elitism is an important issue within evolutionary multi-objective optimization [37] and SEEA is in somehow related to other elitist multi-objective EAs such as PAES [30], PESA [10] or SEAMO [52]. But, contrary to other approaches, no strategy to preserve diversity or to manage the size of the archive is involved here, as solutions are selected randomly and the archive is unbounded. The biggest advantage of this EA is that the population (or the population size if solutions are randomly initialized) is the only problem-independent parameter. If non-dominated solutions are relatively close to each other in the decision space and if the archive is not too small compared to the main population, we believe that SEEA may converge to a good approximation of the efficient set in a very short run-time. However, in some case, this method may prematurely converge or may appear inefficient if promising solutions are far from each other.

Algorithm 1 Simple Elitist Evolutionary Algorithm (SEEA)

Input: P Initial population
Output: A Efficient set approximation

Step 1: Initialization. $A \leftarrow$ non-dominated individuals of P ; $N \leftarrow |P|$; $P' \leftarrow \emptyset$.

Step 2: Selection. Repeat until $|P'| = N$: randomly select an individual from A and add it to the offspring population P' .

Step 3: Variation. Apply crossover and mutation operators to individuals of the offspring population P' .

Step 4: Replacement. $P \leftarrow P'$; $P' \leftarrow \emptyset$.

Step 5: Elitism. $A \leftarrow$ non-dominated individuals of $A \cup P$.

Step 6: Termination. If a stopping criteria is satisfied return A , else go to Step 2.

3.3 Application to the Ring Star Problem

This section provides the problem-specific steps of the metaheuristics introduced earlier. Components designed for the particular case of the bi-objective RSP, such as the encoding mechanism, the population initialization as well as the neighborhood, mutation and crossover operators, are described below.

3.3.1 Solution Encoding

The encoding mechanism used to represent a RSP solution, for both the LS and the EAs, is based on the random keys concept proposed by Bean [5]. This implementation has already been successfully applied for a single-objective version of the RSP in [47]. To each node v_i belonging to the ring we assign exactly one *random key* $x_i \in [0, 1[$, where $x_1 = 0$. A special value is assigned to unvisited nodes. Thus, the ring route associated to a solution corresponds to the nodes ordered according to their random keys in an increasing way; *i.e.* if $x_i < x_j$, then v_j comes after v_i . As an example, a possible representation for the cycle $(v_1, v_7, v_4, v_9, v_2, v_6)$ is given in Figure 2. vertices 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	-

Figure 2: A RSP solution represented by random keys.

3.3.2 Population Initialization

For each optimization method, the initial population has been generated randomly. Each node has a probability $p = 0.5$ that it will be visited or not, and to each visited vertex we associate a key value randomly generated between 0 and 1.

3.3.3 Neighborhood and Mutation Operators

As the RSP is both a routing problem and an assignment problem, different neighborhood and mutation operators have to be designed. Here, we consider the following:

- *insert operator*: adds an unvisited node v_i in the cycle, the position where to insert v_i is chosen in order to minimize the ring cost
- *remove operator*: removes a vertex v_j of the ring
- *2-opt operator*: applies a 2-opt operator between two nodes of the cycle v_i and v_j , *i.e.* the sequence of visited nodes located between v_i and v_j is reversed.

For the LS, the neighbors of a solution are randomly explored, without considering any order between these three operators; and each neighbor is at most visited once. Moreover, note that it is not necessary to completely re-evaluate a solution each time a neighborhood operator is applied. Thus, after an *insert* neighborhood operator, we just have to re-assign unvisited nodes in order to minimize the assignment cost. After a *remove* neighborhood operator, we just have to re-assign the nodes that were previously assigned to the one that has been removed. And, after a *2-opt* neighborhood operator, we just have to recompute the ring cost, the assignment cost being unchanged. In the case of mutations, the operators are applied to randomly chosen vertexes.

3.3.4 Crossover Operator

The crossover operator is a quadratic crossover closely related to the one proposed in [47]. Two randomly selected solutions s_1 and s_2 are divided in a particular position. Then, the first part of s_1 is combined with the second part of s_2 to build a first offspring, and the first part of s_2 is combined with the second part of s_1 to build a second offspring. Every node retains its random key so that it enables an easy reconstruction of the new individuals. Figure 3 illustrates a recombination between two solutions $(v_1, v_7, v_4, v_9, v_2, v_6)$ and $(v_1, v_8, v_4, v_3, v_5)$ after vertex v_6 , what gives rise to a couple of new solutions $(v_1, v_8, v_4, v_2, v_6)$ and $(v_1, v_7, v_9, v_4, v_3, v_5)$. Thanks to the random keys encoding mechanism, solutions having a different ring size can easily be recombined, even if the initial ring structures are generally broken in the offspring solutions.

Parent 1	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	-
Parent 2	Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
	Random key	0	-	0.8	0.7	0.9	-	-	0.2	-	-
Offspring 1	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	-	-
Offspring 2	Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
	Random key	0	-	0.8	0.7	0.9	-	0.2	-	0.5	-

Figure 3: Crossover operator.

Another crossover operator preserving a bigger part of the initial ring structures, has been experimented, but it was tending to reduce the number of nodes belonging to the cycle and then was causing a premature convergence with solutions having a small number of visited nodes.

3.4 Experiments

The metaheuristics described in the previous section have all been implemented using the ParadisEO-MOEO library³ [39]. ParadisEO-MOEO is a C++ white-box object-oriented framework dedicated to the reusable design of metaheuristics for multi-objective optimization. All the algorithms share the same base components for a fair comparison between them. Computational runs were performed on an Intel Core 2 Duo 6600 (2×2.40 GHz) machine, with 2 GB RAM.

3.4.1 Experimental Protocol

Benchmark Test Instances. Search methods have all been tested on different benchmark instances taken from the TSPLIB⁴ [46]. These instances involve between 50 and 300 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 . Moreover, note that, for each resolution method proposed to tackle the bi-objective RSP, 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 instance under consideration.

Table 1: Stopping criteria: running time.

Instance	Running time	Instance	Running time
<i>eil51</i>	20'	<i>kroA150</i>	10'
<i>st70</i>	1'	<i>kroA200</i>	20'
<i>kroA100</i>	2'	<i>pr264</i>	30'
<i>bier127</i>	5'	<i>pr299</i>	50'

Performance Assessment. For each one of the search method, a set of 20 runs per instance, with different initial populations, has been performed. In order to evaluate the quality of the non-dominated front approximations obtained for a specific test instance, we follow the protocol given by Knowles et al. [29]. For a specific instance, we first compute a reference set Z_N^* of non-dominated points extracted from the union of all these fronts. Second, we define $z^{max} = (z_1^{max}, z_2^{max})$, 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 measure the quality of an output set A in comparison to Z_N^* , we compute the difference between these two sets by using the

³ParadisEO is available at <http://paradiseo.gforge.inria.fr>.

⁴<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.

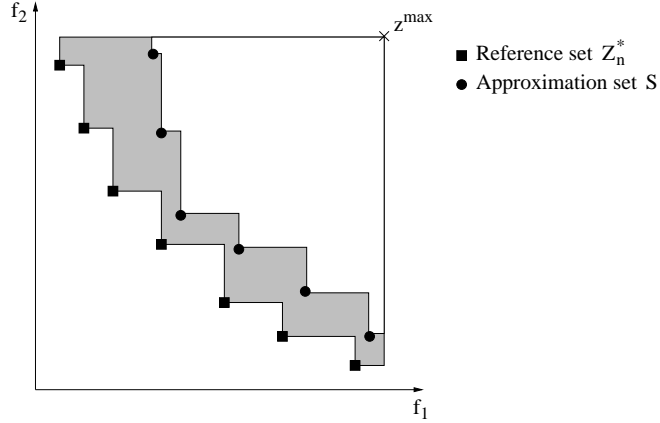


Figure 4: Illustration of the hypervolume difference (I_H^-) between a reference set Z_N^* and a non-dominated front approximation S (shaded area).

unary hypervolume metric [57], $(1.05 \times z_1^{max}, 1.05 \times z_2^{max})$ being the reference point. As illustrated in Figure 4, the hypervolume difference indicator (I_H^-) computes the portion of the objective space that is weakly dominated by Z_N^* and not by A . The more this measure is close to 0, the better is the approximation A . Furthermore, we also consider the additive ϵ -indicator proposed in [58]. Contrary to the one proposed in Equation 5, this indicator is used to compare non-dominated set approximations, and not solutions. The unary additive ϵ -indicator ($I_{\epsilon+}^1$) gives the minimum factor by which an approximation A has to be translated in the criterion space to weakly dominate the reference set Z_N^* . $I_{\epsilon+}^1$ can be defined as follows:

$$I_{\epsilon+}^1(A) = I_{\epsilon+}(A, Z_N^*), \quad (7)$$

where

$$I_{\epsilon+}(A, B) = \min_{\epsilon} \{ \forall z \in B, \exists z' \in A : z'_i - \epsilon \leq z_i, \forall 1 \leq i \leq n \}. \quad (8)$$

As a consequence, for each test instance, we obtain 20 hypervolume differences and 20 $I_{\epsilon+}$ measures, corresponding to the 20 runs, per algorithm. As suggested by Knowles et al. [29], 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 [29], 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 search method M_1 is significantly better than the sample of approximation sets obtained by a search method M_2 , 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 PISA⁵ [7].

⁵The package is available at <http://www.tik.ee.ethz.ch/pisa/assessment.html>.

3.4.2 Local Search Parameter Analysis

Parameter Setting. For the IBMOLS algorithm, we set the population size according to the number of vertexes involved in the instance (see Table 2). For each instance, a small (S), a medium (M) and a large (L) population size have been experimented. Likewise, the noise rate for the population re-initialization in the iterated version of IBMOLS (denoted by I-IBMOLS) is set to a fixed percentage of the instance’s size. We investigate three different values for this noise rate: 5%, 10% and 20%. Then, $0.05 \times n$, $0.1 \times n$ and $0.2 \times n$ random mutations are applied respectively for a problem with n nodes. Finally, the scaling factor κ of the Equation 6 is set to 0.05.

Table 2: Population size parameter for the I-IBMOLS algorithm.

Instance	Population size		
	S	M	L
<i>eil51</i>	10	20	30
<i>st70</i>	10	20	30
<i>kroA100</i>	20	30	50
<i>bier127</i>	20	30	50
<i>kroA150</i>	20	30	50
<i>kroA200</i>	20	30	50
<i>pr264</i>	30	50	70
<i>pr299</i>	30	50	70

Computational Results and Discussion. The results we obtained for the I_H^- and the $I_{\epsilon+}^1$ metrics are respectively summarized in Table 3 and Table 4. These tables give, for every instance, the number of times that a specific parametrization is significantly worse than another one for the I-IBMOLS algorithm. So, a lower score is better, and *zero* means that the parametrization under consideration is never significantly outperformed. According to the I_H^- metric, a medium population size with a random noise of 5% or 10% seems to perform well on every test instance, except for the *pr264* instance where a medium population size and a random noise of 10% is significantly worse than another parametrization. Moreover, as shown in Table 4, a medium population size with a random noise parametrized to 10% generally seems to perform slightly better than with a random noise of 5% according to the $I_{\epsilon+}^1$ metric. The only instances for which such a parametrization is significantly worse than at least another one are the *kroA200* and the *pr264* problems. Then, for all the other instances, we choose to set the number of individuals in the population to the medium size and the noise rate to 10% for comparing I-IBMOLS to the other metaheuristics. For the *kroA200* instance, the same parametrization will be used as it seems to be the best compromise between the two metrics. Finally, for the *pr264* instance, we will use a small population size and a noise rate of 20%. The value of all the parameters used by I-IBMOLS for the remainder of the paper are summarized in Table 8.

Table 3: Performance comparison of different population size and noise rate parameter settings for I-IBMOLS according to the I_H^- metric by using a Mann-Whitney statistical test with a significance level of 5%. Each column gives the number of algorithms with another parameter setting by which it is statistically outperformed for the instance under consideration.

noise rate pop. size	5%			10%			20%		
	S	M	L	S	M	L	S	M	L
<i>eil51</i>	3	0	3	3	0	3	8	1	3
<i>st70</i>	6	0	0	7	0	0	5	0	1
<i>kroA100</i>	1	0	6	1	0	6	2	2	6
<i>bier127</i>	1	0	6	1	0	6	5	1	6
<i>kroA150</i>	1	0	6	4	0	6	5	1	6
<i>kroA200</i>	2	0	1	2	0	2	8	2	2
<i>pr264</i>	5	0	2	2	1	4	0	1	5
<i>pr299</i>	6	0	0	8	0	3	3	0	4

Table 4: Performance comparison of different population size and noise rate parameter settings for I-IBMOLS according to the $I_{\epsilon+}^1$ metric by using a Mann-Whitney statistical test with a significance level of 5%. Each column gives the number of algorithms with another parameter setting by which it is statistically outperformed for the instance under consideration.

noise rate pop. size	5%			10%			20%		
	S	M	L	S	M	L	S	M	L
<i>eil51</i>	7	0	3	6	0	3	5	0	3
<i>st70</i>	7	0	0	7	0	0	6	0	4
<i>kroA100</i>	0	0	6	0	0	6	1	2	6
<i>bier127</i>	0	0	6	1	0	6	1	1	6
<i>kroA150</i>	2	1	6	3	0	6	2	0	6
<i>kroA200</i>	8	1	1	7	1	1	1	0	1
<i>pr264</i>	2	2	6	1	2	6	0	2	6
<i>pr299</i>	8	3	3	6	0	3	0	0	6

3.4.3 Evolutionary Algorithms Parameter Analysis

Parameter Setting. For all the EAs, population sizes of 50, 100 and 200 individuals have been experimented. Note that the instance-dependent population size setting used for I-IBMOLS has also been investigated, but the results were overall less efficient. The scaling factor κ used by IBEA is set to 0.05. At last, the crossover probability is set to 0.25, and the mutation probability to 1.00, with a probability of 0.25, 0.25 and 0.50 for the *remove*, the *insert* and the *2-opt* operator, respectively.

Computational Results and Discussion. As pointed out in Table 5, the results obtained by IBEA are quite heterogeneous from a test instance to another, and no general conclusion can be drawn. Thus, we will use a population size of 50 for the *eil51*, the *st70*, the *pr264* and the *pr299* instances, a population size of 100 for the *kroA100* and the *bier127* instances, and a population size of 200 for the *kroA150* and the *kroA200* instances. For NSGA-II, the best performances are usually obtained with a large population size. Then, a population of 100 individuals for instances with less than 100 nodes and of 200 individuals for instances with more than 100 nodes will be used for the following. These parameters are never significantly outperformed according to the metrics we used. Finally, as shown in Table 7, the population size does not appear to have a big influence on the outcome of SEEA, what is not surprising given the design of the algorithm (see Section 3.2.3). As a consequence, we will use a population of 100 individuals as this is the only parametrization that is strictly never significantly outperformed for every test instance. All the parameters used by the EAs for the next section are recapitulated in Table 8.

Table 5: Performance comparison of different population size parameter settings for IBEA according to the I_H^- and the $I_{\epsilon+}^1$ metrics by using a Mann-Whitney statistical test with a significance level of 5%. Each column gives the number of algorithms with another parameter setting by which it is statistically outperformed for the instance under consideration.

pop. size	I_H^-			$I_{\epsilon+}^1$		
	50	100	200	50	100	200
<i>eil51</i>	0	0	2	0	0	2
<i>st70</i>	0	0	2	0	1	1
<i>kroA100</i>	0	0	1	0	0	0
<i>bier127</i>	0	0	0	0	0	0
<i>kroA150</i>	1	0	0	1	0	0
<i>kroA200</i>	0	2	0	1	2	0
<i>pr264</i>	0	1	1	0	1	1
<i>pr299</i>	0	1	1	0	1	1

3.4.4 Comparison between Search Methods

Previous experiments allowed us to determine satisfying parameters for each one of the proposed algorithms. The parameter setting is summarized in Table 8. In this section, experiments were carried out with the aim of comparing the results obtained by the different metaheuristics to solve the problem under

Table 6: Performance comparison of different population size parameter settings for NSGA-II according to the I_H^- and the $I_{\epsilon+}^1$ metrics by using a Mann-Whitney statistical test with a significance level of 5%. Each column gives the number of algorithms with another parameter setting by which it is statistically outperformed for the instance under consideration.

pop. size	I_H^-			$I_{\epsilon+}^1$		
	50	100	200	50	100	200
<i>eil51</i>	1	0	2	1	0	2
<i>st70</i>	2	0	0	2	0	0
<i>kroA100</i>	2	1	0	2	1	0
<i>bier127</i>	2	1	0	2	1	0
<i>kroA150</i>	2	1	0	2	1	0
<i>kroA200</i>	2	1	0	2	1	0
<i>pr264</i>	2	1	0	2	1	0
<i>pr299</i>	2	1	0	2	1	0

Table 7: Performance comparison of different population size parameter settings for SEEA according to the I_H^- and the $I_{\epsilon+}^1$ metrics by using a Mann-Whitney statistical test with a significance level of 5%. Each column gives the number of algorithms with another parameter setting by which it is statistically outperformed for the instance under consideration.

pop. size	I_H^-			$I_{\epsilon+}^1$		
	50	100	200	50	100	200
<i>eil51</i>	0	0	0	0	0	0
<i>st70</i>	0	0	0	0	0	0
<i>kroA100</i>	0	0	0	0	0	0
<i>bier127</i>	2	0	0	2	0	0
<i>kroA150</i>	0	0	0	0	0	0
<i>kroA200</i>	0	0	0	0	0	0
<i>pr264</i>	0	0	0	0	0	1
<i>pr299</i>	0	0	0	0	0	0

Table 8: Parameter setting used to compare search methods.

Instance	I-IBMOLS		IBEA	NSGA-II	SEEA
	pop. size	noise rate	pop. size	pop. size	pop.size
<i>eil51</i>	M = 20	10%	100	100	100
<i>st70</i>	M = 20	10%	50	100	100
<i>kroA100</i>	M = 30	10%	100	200	100
<i>bier127</i>	M = 30	10%	100	200	100
<i>kroA150</i>	M = 30	10%	200	200	100
<i>kroA200</i>	M = 30	10%	200	200	100
<i>pr264</i>	S = 30	20%	50	200	100
<i>pr299</i>	M = 50	10%	50	200	100

consideration. A further analysis of these results was conducted in comparison to the previous ones. Then, Table 9 and Table 10 present the results we respectively obtained for the I_H^- and the $I_{\epsilon+}^1$ metrics. These tables show, for every test instance, the outcome of the statistical test as well as the significance level given by the p-value. With respect to the I_H^- metric, we can see that I-IBMOLS is never outperformed by any other search method for all the instances we tested. The only instance for which I-IBMOLS does not perform better than the other algorithms is *pr264*, where there is no significant difference between the results it obtained and the ones of SEEA. Furthermore, according to the same metric, SEEA seems overall more efficient than IBEA and NSGA-II. Indeed, it always performs significantly better than NSGA-II and is statistically outperformed by IBEA only on the *bier127* and the *kroA200* instances. Lastly, IBEA performs significantly better than NSGA-II on every test instance. Likewise, as shown in Table 10, the results we obtained for the $I_{\epsilon+}^1$ metric, reveal that I-IBMOLS is globally more efficient than SEEA, IBEA and NSGA-II on most test instances. Nevertheless, it is not the case on the *st70* and *kroA100* instances, where there is no significant difference between I-IBMOLS and SEEA, and on the *pr264* instance, where I-IBMOLS is statistically outperformed by SEEA. Moreover, the results of NSGA-II are always worst than the ones of the other methods for all the instances. At last, SEEA provides better performance than IBEA on every problem, except for the *kroA200* instance, where the difference between both is not significant, and for the *bier127* instance, where IBEA performs better. Then, to summarize the results given in Table 9 and Table 10 for the I_H^- and the $I_{\epsilon+}^1$ metric respectively, I-IBMOLS seems to be the overall most efficient optimization method to solve the bi-objective ring star problem. Then comes SEEA that performs better than any other EA. Finally, IBEA performs better than NSGA-II, the later being always outperformed by the other search methods. All these observations are confirmed by the box-plots given in Figure 5 and Figure 6. We can see that the measures obtained by NSGA-II and, to a lesser extent, IBEA are both generally quite far from the ones obtained by IBMOLS and SEEA, except for some particular instances such as *bier127* and *kroA200*. Moreover, the differences between the measures obtained by I-IBMOLS and SEEA are very thin for most of the cases, even if SEEA is generally outperformed, as pointed out above.

In order to study the differences between I-IBMOLS and SEEA more finely, examples of the empirical attainment function obtained by these two algorithms are represented in Figure 8 and Figure 10 for the *bier127* and the *pr264* instances respectively. These figures illustrate the limit of the objective space that is attained by at least 90% of the runs by both methods. The reader is referred to [19] for more details about the concept of empirical attainment function. For the *bier127* instance, where the results obtained by I-IBMOLS are much better than the ones of SEEA with respect to both metrics, we can see that SEEA seems to have more trouble in finding solutions having a low ring cost, whereas it seems slightly better than I-IBMOLS for finding interesting solutions with a small assignment cost. For the *pr264* instance, where SEEA is overall more efficient, it is less obvious to observe any superiority graphically. Such a result mainly seems to be due to the discontinuity of the trade-off surface, for which a good approximation is given in Figure 9. An example of a more often encountered Pareto front is illustrated in Figure 7 for the *bier127* instance.

Table 9: Performance comparison for I-IBMOLS, IBEA, NSGA-II and SEEA according to the I_H^- metric by using a Mann-Whitney statistical test. The “p-value” columns give the p-value of the statistical test. The “T” columns give the outcome of the statistical test for a significance level of 5%: either the results of the search method located at a specific row are significantly better than those of the search method located at a specific column (\succ), either they are worse (\prec), or there is no significant difference between both (\equiv).

		I-IBMOLS		IBEA		NSGA-II	
		p-value	T	p-value	T	p-value	T
<i>eil51</i>	IBEA	$4.578 \cdot 10^{-6}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.151 \cdot 10^{-8}$	\prec		
	SEEA	0.076	\prec	$1.053 \cdot 10^{-4}$	\succ	$3.151 \cdot 10^{-8}$	\succ
<i>st70</i>	IBEA	$1.781 \cdot 10^{-4}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.151 \cdot 10^{-8}$	\prec		
	SEEA	0.011	\prec	0.111	\succ	$3.151 \cdot 10^{-8}$	\succ
<i>kroA100</i>	IBEA	0.042	\prec				
	NSGA-II	$5.608 \cdot 10^{-7}$	\prec	$5.608 \cdot 10^{-7}$	\prec		
	SEEA	0.004	\prec	> 5%	\equiv	$3.151 \cdot 10^{-8}$	\succ
<i>bier127</i>	IBEA	$4.832 \cdot 10^{-4}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.151 \cdot 10^{-8}$	\prec		
	SEEA	$3.151 \cdot 10^{-8}$	\prec	$4.889 \cdot 10^{-7}$	\prec	$9.641 \cdot 10^{-7}$	\succ
<i>kroA150</i>	IBEA	0.003	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.151 \cdot 10^{-8}$	\prec		
	SEEA	0.003	\prec	> 5%	\equiv	$8.914 \cdot 10^{-8}$	\succ
<i>kroA200</i>	IBEA	0.112	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.151 \cdot 10^{-8}$	\prec		
	SEEA	$5.483 \cdot 10^{-5}$	\prec	$3.122 \cdot 10^{-5}$	\prec	$3.151 \cdot 10^{-8}$	\succ
<i>pr264</i>	IBEA	$1.084 \cdot 10^{-5}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.151 \cdot 10^{-8}$	\prec		
	SEEA	> 5%	\equiv	$3.151 \cdot 10^{-8}$	\succ	$3.151 \cdot 10^{-8}$	\succ
<i>pr299</i>	IBEA	$5.877 \cdot 10^{-6}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.151 \cdot 10^{-8}$	\prec		
	SEEA	$2.114 \cdot 10^{-7}$	\prec	> 5%	\equiv	$3.151 \cdot 10^{-8}$	\succ

Table 10: Performance comparison for I-IBMOLS, IBEA, NSGA-II and SEEA according to the $I_{\epsilon+}^1$ metric by using a Mann-Whitney statistical test. The “p-value” columns give the p-value of the statistical test. The “T” columns give the outcome of the statistical test for a significance level of 5%: either the results of the search method located at a specific row are significantly better than those of the search method located at a specific column (\succ), either they are worse (\prec), or there is no significant difference between both (\equiv).

		I-IBMOLS		IBEA		NSGA-II	
		p-value	T	p-value	T	p-value	T
<i>eil51</i>	IBEA	$1.177 \cdot 10^{-6}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$3.663 \cdot 10^{-8}$	\prec		
	SEEA	0.117	\prec	$9.604 \cdot 10^{-6}$	\succ	$3.151 \cdot 10^{-8}$	\succ
<i>st70</i>	IBEA	$1.084 \cdot 10^{-5}$	\prec				
	NSGA-II	$3.658 \cdot 10^{-8}$	\prec	$9.595 \cdot 10^{-6}$	\prec		
	SEEA	> 5%	\equiv	$5.733 \cdot 10^{-8}$	\succ	$3.146 \cdot 10^{-8}$	\succ
<i>kroA100</i>	IBEA	0.005	\prec				
	NSGA-II	$7.363 \cdot 10^{-7}$	\prec	$1.846 \cdot 10^{-6}$	\prec		
	SEEA	> 5%	\equiv	0.027	\succ	$1.192 \cdot 10^{-7}$	\succ
<i>bier127</i>	IBEA	$8.497 \cdot 10^{-5}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	$4.941 \cdot 10^{-8}$	\prec		
	SEEA	$3.151 \cdot 10^{-8}$	\prec	$1.377 \cdot 10^{-7}$	\prec	0.008	\succ
<i>kroA150</i>	IBEA	$2.947 \cdot 10^{-6}$	\prec				
	NSGA-II	$1.192 \cdot 10^{-7}$	\prec	$1.390 \cdot 10^{-4}$	\prec		
	SEEA	0.003	\prec	$4.898 \cdot 10^{-6}$	\succ	$5.608 \cdot 10^{-7}$	\succ
<i>kroA200</i>	IBEA	0.037	\prec				
	NSGA-II	$5.733 \cdot 10^{-8}$	\prec	$3.146 \cdot 10^{-8}$	\prec		
	SEEA	0.037	\prec	> 5%	\equiv	$3.151 \cdot 10^{-8}$	\succ
<i>pr264</i>	IBEA	$8.902 \cdot 10^{-8}$	\prec				
	NSGA-II	$4.256 \cdot 10^{-8}$	\prec	0.084	\succ		
	SEEA	$1.745 \cdot 10^{-5}$	\succ	$3.142 \cdot 10^{-8}$	\succ	$3.146 \cdot 10^{-8}$	\succ
<i>pr299</i>	IBEA	$3.151 \cdot 10^{-8}$	\prec				
	NSGA-II	$3.151 \cdot 10^{-8}$	\prec	0.0495	\prec		
	SEEA	$5.877 \cdot 10^{-6}$	\prec	$3.225 \cdot 10^{-7}$	\succ	$3.151 \cdot 10^{-8}$	\succ

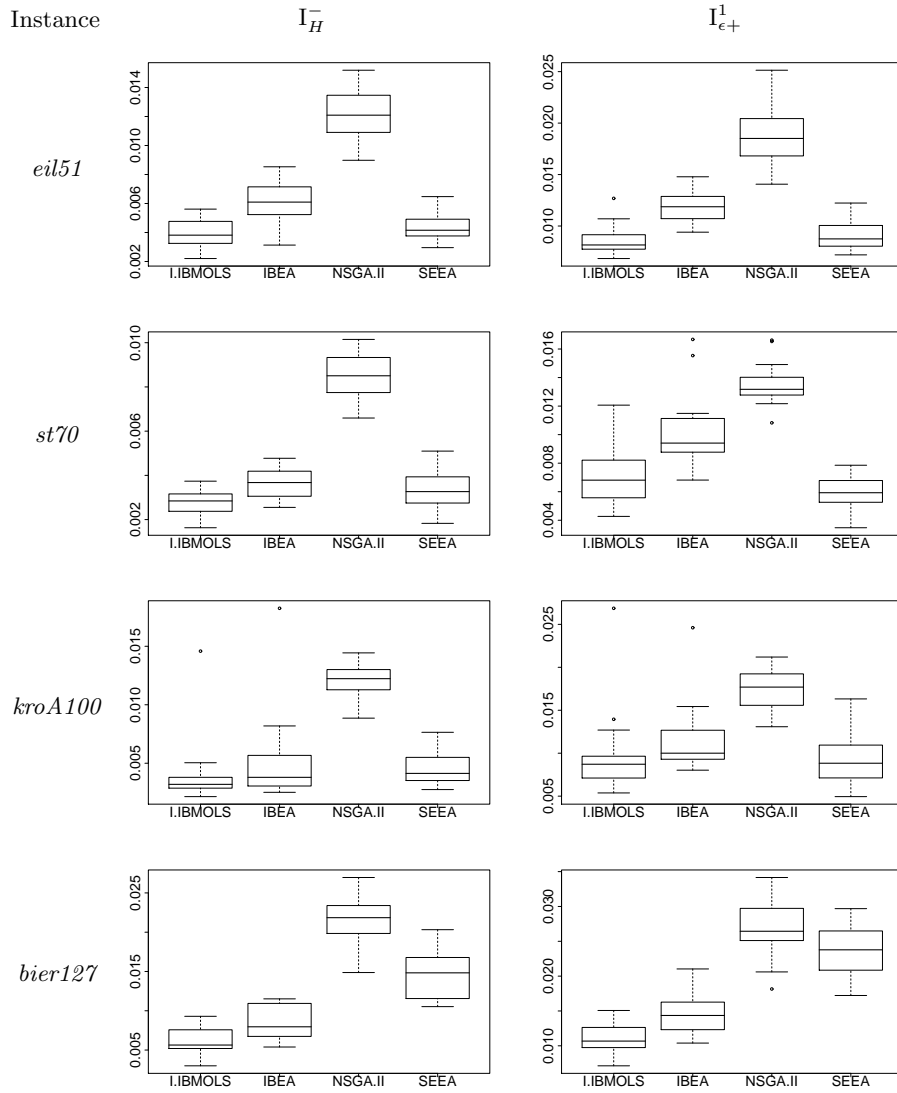


Figure 5: Performance comparison for I-IBMOLS, IBEA, NSGA-II and SEEA according to the I_H^- and the $I_{\epsilon+}^1$ metric (1).

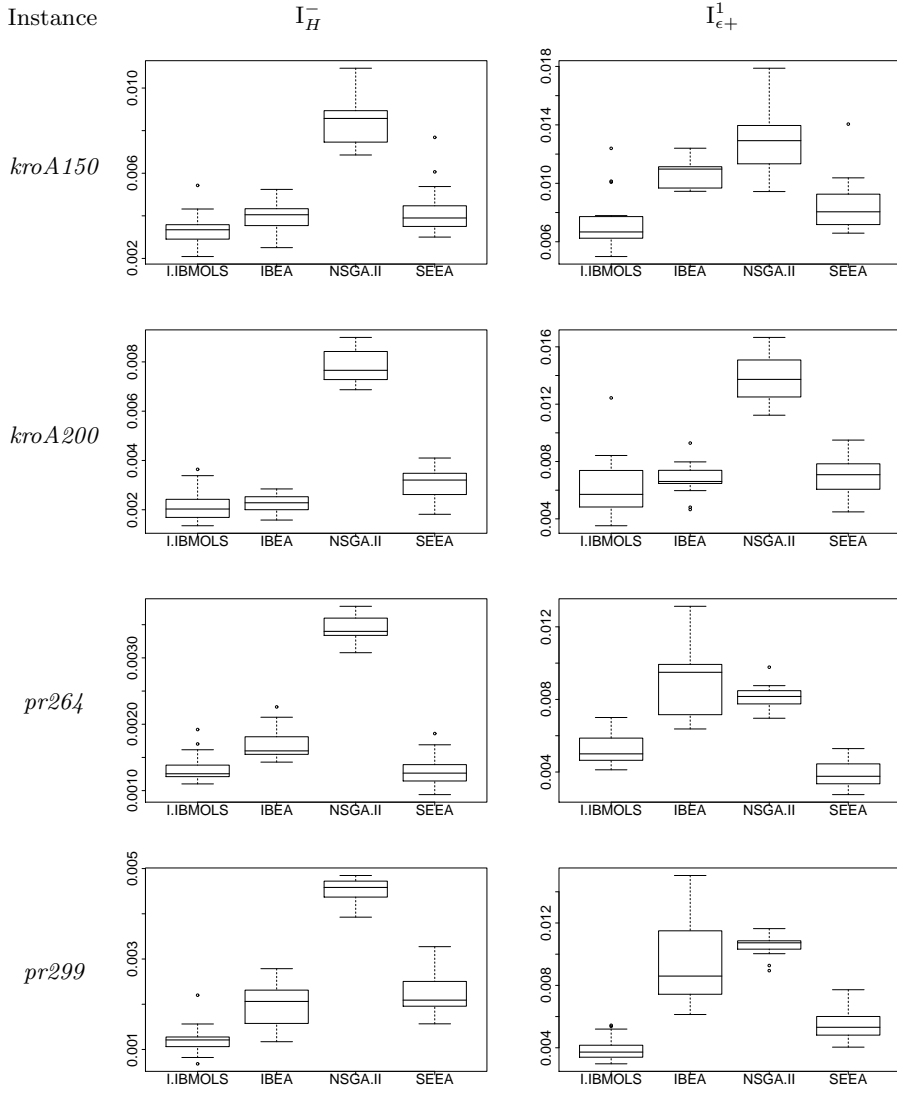


Figure 6: Performance comparison for I-IBMOLS, IBEA, NSGA-II and SEEA according to the I_H^- and the $I_{\epsilon+}^1$ metric (2).

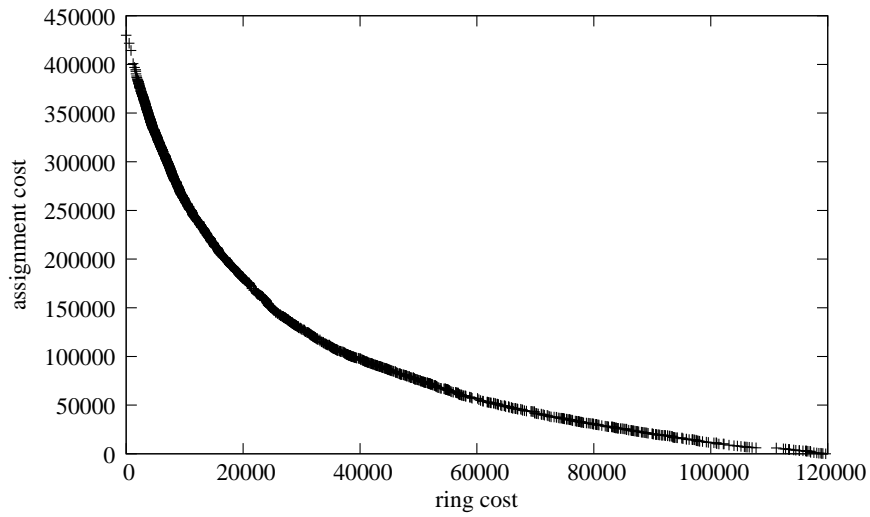


Figure 7: Best set of non-dominated points found for the *bier127* test instance.

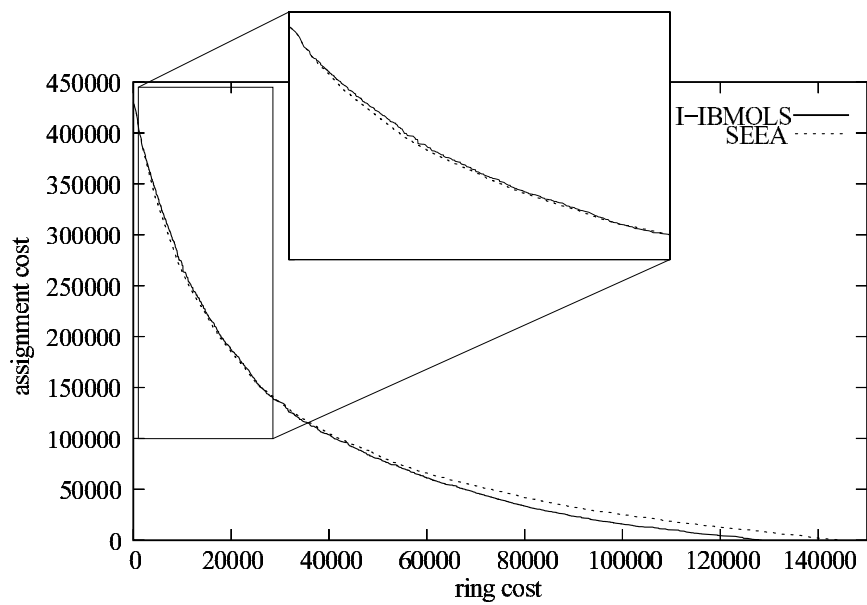


Figure 8: 90%-attainment surface plot obtained by the approximation sets found by I-IBMOLS and SEEA for the *bier127* test instance.

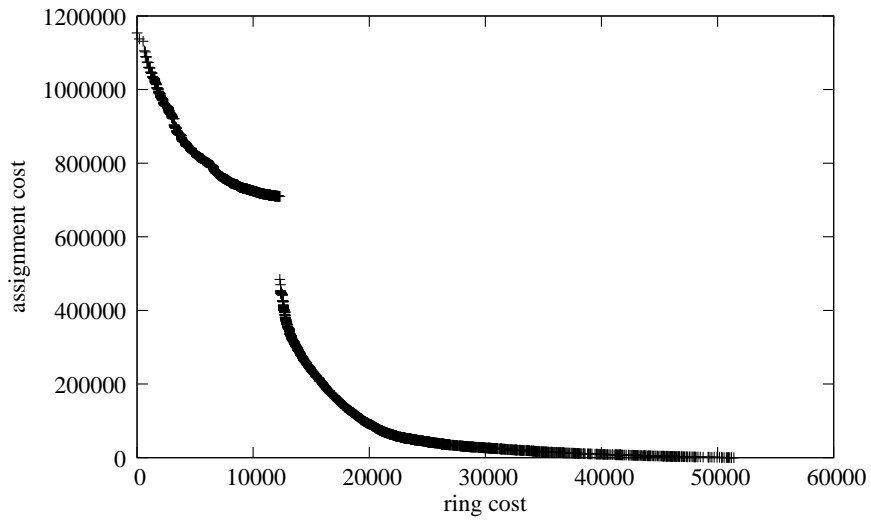


Figure 9: Best set of non-dominated points found for the *pr264* test instance.

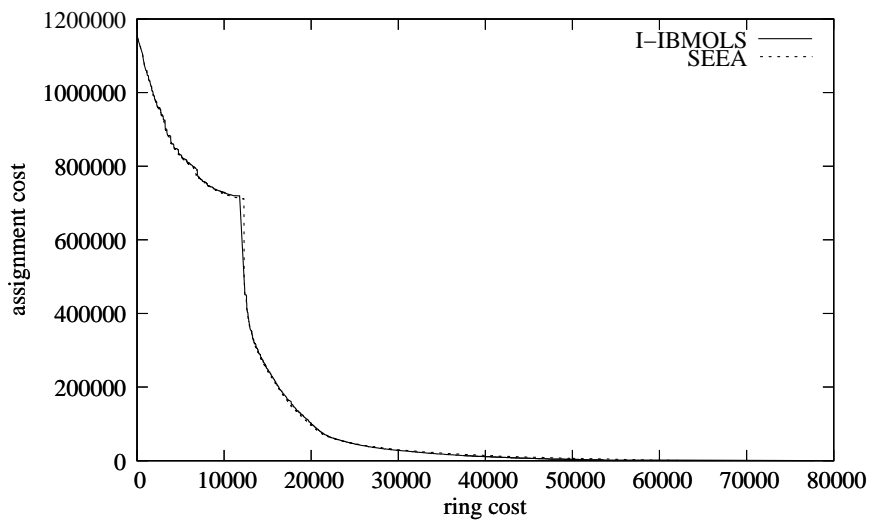


Figure 10: 90%-attainment surface plot obtained by the approximation sets found by I-IBMOLS and SEEA for the *pr264* test instance.

One of the main characteristics of the problem under consideration seems to be the high number of points located in the trade-off surface, as shown in Figure 7 and Figure 9. Then, after a certain number of iterations, a large part of the population involved in NSGA-II, IBEA and I-IBMOLS might map to potentially non-dominated points. This could explain the low efficiency of NSGA-II. Since the same rank is assigned to the major part of the population, only the crowding distance is used to compare solutions. However, it is not the case in SEEA because all non-dominated solutions contained in the archive can potentially take part in the evolution engine. Moreover, the indicator-based fitness assignment scheme is obviously much more suited to discriminate potentially efficient solutions than the single crowding distance. Moreover, the high performance of I-IBMOLS in comparison to IBEA might depend on how close are the solutions which map to non-dominated points in the decision space. If these solutions are close to each other, according to the neighborhood operators, a LS is known to be particularly well-suited to find additional interesting solutions by exploring the neighborhood of a potentially efficient solution. On the contrary, an EA usually explores the decision space in a more random way. Thus, a landscape analysis could be interesting to study the bi-objective RSP in more depth. Besides, the results obtained by SEEA are quite satisfying in contrast to those obtained by the other EAs. Indeed, NSGA-II and IBEA are two state-of-art search methods for multi-objective optimization and are widely used in the community. Nevertheless, despite its simplicity, SEEA seems to be much more efficient for solving the B-RSP in a given amount of time. It could then be interesting to experiment the ability of this method for the resolution of other kinds of multi-objective combinatorial optimization problems.

As a next step, we will try to solve problem instances involving an even bigger number of nodes to verify if our observations are still valid. It could then be interesting to design a cooperation scheme between two different methods (*i.e.* the local search procedure and an evolutionary algorithm) in order to benefit from the respective quality of each one of them. The resulting hybrid metaheuristic could be particularly effective for solving large size problems.

4 A Cooperative Approach

This section presents a cooperative approach combining SEEA and the non-iterative version of IBMOLS in order to solve the bi-objective RSP. Two variants are proposed: a periodic one that operates a systematic cooperation and an auto-adaptive one that decides on-line when the cooperation must occur.

4.1 Motivation

Designing metaheuristics for solving combinatorial optimization problems is generally a matter of intensification and diversification. This is even more pronounced for MOPs where the goal is to find a well-converged and well-diversified efficient set approximation. However, LSs are known to be particularly efficient as intensifying methods whereas EAs are clearly powerful to explore the decision space thanks to their variation operators. Instead of trying to improve one method in term of diversification or the other in term of intensification, a common approach is to hybridize both in order to make them cooperate and then to

benefit of their respective behaviors. Thus, hybrid metaheuristics have shown their efficiency to solve different kinds of optimization problems [49], including MOPs [18]. They generally consist of a search method that cooperates with a second one in order to improve a solution or a set of solutions.

In the previous section, we saw that SEEA and I-IBMOLS were overall more efficient than NSGA-II and IBEA to approximate the efficient set for the problem under consideration. Moreover, these two methods are quite different to each other and do not explore the search space in the same way. SEEA has been conceived in order to find a rough approximation of the Pareto set in a very short amount of time whereas the non-iterative version of IBMOLS is able to improve an approximated set in a very efficient way. It could then be interesting to design a cooperation scheme between these two algorithms. The resulting hybrid metaheuristic could be particularly efficient for solving large size problems. Furthermore, both methods maintain a secondary population (the archive) in parallel of the main population to store non-dominated solutions. This archive is not only used as an external storage (what is the case, for instance, in IBEA and NSGA-II), but also takes part in the evolution engine as it serves to build new solutions to explore. Thus, each method can manage its own population and therefore use the archive as a single shared memory.

4.2 Cooperative Schemes

The general idea of our hybridization scheme is to run SEEA and to launch IBMOLS regularly by using a subset of archive items as an initial population. As the non-iterative version of IBMOLS stops when its own archive does not receive any new efficient solution anymore, we can restart the SEEA process until the next step of the hybrid algorithm. A step of the hybrid metaheuristic can, for instance, be defined by a certain amount of time or by a certain number of generations. Besides, as SEEA uses the non-dominated solutions found by IBMOLS to create new ones and vice versa, the global archive is the only memory shared by the two search agents to exchange information. Resulting from this, we can imagine two versions of the hybrid algorithm: (i) a *periodic version*, in which IBMOLS is launched at each step, and (ii) an *auto-adaptive version*, in which IBMOLS is launched at a specific step only if a condition is verified. These two approaches will be denoted by PCS (for *Periodic Cooperative Search*) and ACS (for *Auto-adaptive Cooperative Search*) in the remainder of the paper, and are respectively illustrated in Figure 11 and Figure 12.

The ACS method decides by itself, and on-line, if it would be interesting to launch IBMOLS at a specific instant of the search process. The condition that we consider here to start IBMOLS during a particular step of the ACS is that the archive of potentially efficient solutions does not improve enough regarding to the optimization scenario. One possibility is to measure the quality of the current archive A^t in comparison to the one of the previous step A^{t-1} . Different metrics exist to evaluate the convergence of an approximated efficient set regarding to another. For instance, we could have computed the hypervolume [57] of A^t and A^{t-1} and measure the difference between both. But, even if it evaluates the quality of an approximated efficient set both in term of convergence and diversity, the hypervolume metric has the drawback of being time consuming. Thus, let us introduce the *contribution* metric C proposed by Meunier et al. [42]. This metric gives an idea of the quality of an approxi-

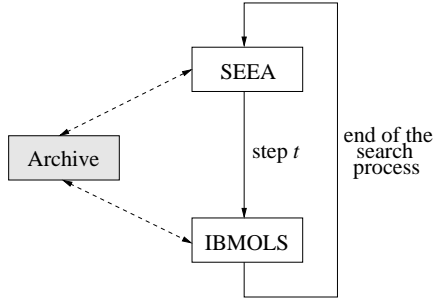


Figure 11: Illustration of the Periodic Cooperative Search (PCS).

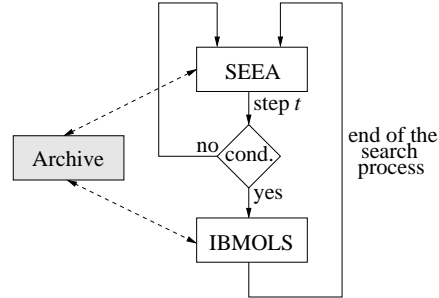


Figure 12: Illustration of the Auto-adaptive Cooperative Search (ACS).

mated efficient set S_1 compared to another one S_2 in term of convergence and can be computed in a reasonable running time. If S^* denotes the set of non-dominated solutions given by $S_1 \cup S_2$, the contribution of S_1 on S_2 evaluates the proportion of non-dominated solutions represented by S_1 in S^* . Then, if $f(S_1) = f(S_2)$, $C(S_1, S_2) = C(S_2, S_1) = 0.5$. If every solution of S_1 is dominated by at least one solution of S_2 , $C(S_1, S_2) = 0$. And, generally speaking, $C(S_1, S_2) + C(S_2, S_1) = 1$. In our case, at each step t of the ACS, we compute the contribution of the current archive A^t on the archive of the previous step A^{t-1} . Thus, as non-dominated solutions are not lost between two steps, A^t is at least as good as A^{t+1} and the set of non-dominated solutions of $A^t \cup A^{t-1}$ is then A^t . As a consequence, we know that $C(A^t, A^{t-1}) \in [0.5, 1]$. Assuming that the archive does not improve enough if the contribution of A^t on A^{t-1} is less than a pre-defined threshold $\delta \in [0.5, 1]$, we choose to launch IBMOLS only if $C(A^t, A^{t-1}) \leq \delta$. Let us remark that an ACS with a $\delta = 0.5$ would result in SEEA, and that an ACS with a $\delta = 1.0$ would result in PCS, the time spent in calculating the different contribution values in less.

4.3 Related Works

Different schemes exist on how two search methods can be combined. According to the taxonomy proposed in [49], two levels (low and high) and two modes (relay and cooperative) of hybridization can be distinguished. In *low-level* hybridization, a given function of a metaheuristic is replaced by another metaheuristic. In *high-level* hybridization, there is no direct relationship between the internal workings of a metaheuristic. A second hierarchical classification deals with the mode of cooperation. The *relay* mode consists of a method applied after another one in a pipeline way, the last one using the output of the previous one as its input. On the contrary, *teamwork* hybridization represents a cooperative optimization model in which search agents exchange information with the others. Here, we consider that the optimization process alternates between two search algorithms: SEEA and the non-iterative version of IBMOLS. Both methods are self-contained and cooperates with each other via a global archive. Therefore, the resulting hybrid metaheuristic can be classified on the *High-level Teamwork Hybrid* (HTH) class of the taxonomy introduced in [49] and can then

be denoted by HTH(SEEA+IBMOLS). In their survey on hybrid metaheuristics to solve multi-objective combinatorial optimization problems [18], Ehrgott and Gandibleux identify three categories of methods hybridizing an EA with a neighborhood search algorithm: (i) an hybridization to make a method more aggressive, (ii) an hybridization to drive a method and (iii) an hybridization for exploiting complementary strengths. The last one consists of alternating between both search methods, what is the case within our hybridization. But, most of existing approaches occur in a pipeline way, like in the relay mode. The few teamwork hybridization techniques of the literature to solve MOPs are most often compound of the same kind of metaheuristics, what is generally the case in the island model. Moreover, this class of hybrid methods often uses a neighborhood search method combining a set of scalarizing functions. The originality of the approach proposed in this paper is that search agents are based on different types of metaheuristics, are hybridized in a teamwork mode, and do not use any scalar approach to convert the multiple objective functions into a single one. Furthermore, one of the variant we propose, the auto-adaptive one, automatically detects when to start the neighborhood search according to the optimization scenario.

4.4 Experiments

In order to experiment the efficiency of our two cooperative approaches, we compare them to SEEA and to the iterative version of IBMOLS by using the same experimental protocol as the one defined in Section 3.4.1. After having set the parameters for each hybrid search method, we give some computational results and we discuss the contribution of the hybridization for the resolution of the B-RSP. Finally, we give an idea of the behavior of our approach compared to a state-of-art single-objective approach proposed in [34].

4.4.1 Parameter Setting

For both cooperative search methods, the population size managed by SEEA is set to 100, and the population size managed by IBMOLS is set according to the instance under consideration. These sizes have been set on the same way that we did for the stand-alone iterative version of IBMOLS on the previous section (see Table 2). But, for large-size instances, initial experiments were not satisfying since the hybrid algorithms were generally not able to launch IBMOLS more than once during the search process as it was too much time consuming. For this reason, we bounded the IBMOLS population size to 30. Then, a IBMOLS population of 20 individuals has been set for instances with less than 100 nodes, and a IBMOLS population of 30 individuals has been set for instances with 100 nodes and more. The step t of the hybrid algorithms has been set to 0.5% of the total run time for the instance under consideration, rounded to the nearest upper integer. Finally, we investigated different δ values for the ACS method: 0.6, 0.7, 0.8 and 0.9. As pointed out above, an ACS with a δ value of 0.5 and 1.0 is similar to PCS and SEEA, respectively.

4.4.2 Computational Results and Discussion

ACS Parameter Analysis. As a first step, we analyze the influence of the δ parameter on the results of the ACS method. Results are summarized in Table 11. A value of 0.6 is generally less efficient than at least one other setting on most test instances according to both metrics. For the other values, the results are roughly comparable, even if a $\delta = 0.8$ is never statistically outperformed by any other value according to I_H^- and $I_{\epsilon+}^1$. Therefore, we will use this parameter setting for the next experiments.

Table 11: Performance comparison of different δ parameter settings for the ACS method according to the I_H^- and the $I_{\epsilon+}^1$ metrics by using a Mann-Whitney statistical test with a significance level of 5%. Each column gives the number of algorithms with another parameter setting by which it is statistically outperformed for the instance under consideration.

δ	I_H^-				$I_{\epsilon+}^1$			
	0.60	0.70	0.80	0.90	0.60	0.70	0.80	0.90
<i>eil51</i>	0	0	0	0	0	0	0	0
<i>st70</i>	0	0	0	0	0	0	0	0
<i>kroA100</i>	1	0	0	0	2	1	0	0
<i>bier127</i>	0	0	0	1	3	0	0	0
<i>kroA150</i>	2	1	0	0	3	2	0	0
<i>kroA200</i>	3	0	0	0	3	2	0	0
<i>pr264</i>	0	0	0	0	0	0	0	0
<i>pr299</i>	3	0	0	0	3	0	0	0

Comparison between Search Methods. In addition to the benchmark test instances we investigated previously, we experimented two bigger ones: *pr439* and *pr1002*. The maximum running time for these new instances has been set to 70 and 100 minutes, respectively. Following the general trend identified in Section 3.4.2, we set the instance-specific parameters for the iterative version of IBMOLS as follows: a population of 70 and 100 individuals respectively, and a noise rate of 10%. The other parameters were set in the same way than for other instances.

Table 12 and Table 13 respectively summarize the results we obtained by comparing I-IBMOLS, SEEA, PCS and ACS according to the I_H^- and to the $I_{\epsilon+}^1$ metrics. For the 10 benchmark test instances we experimented, we can see that both cooperative search methods are never statistically outperformed by SEEA. Moreover, they perform significantly better according to at least one metric on all instances except *eil51*. Similarly, PCS and ACS are at least as good as I-IBMOLS on every test instance. While there is generally no significant difference for small-size instances, the efficient set approximations found by the cooperative approaches are statistically better than I-IBMOLS for problems with 150 nodes and more (except for the *pr299* instance). Thus, contrary to what has been pointed out in the previous section about SEEA, the cooperative methods do not seem to have trouble in finding solution having a low ring cost in comparison to I-IBMOLS, as illustrated in the empirical attainment function given in Figure 15. At last, in most cases, the differences between PCS

and ACS are not statistically significant, except for the *bier127* instance where PCS performs better, and for the *kroA150* and the *pr439* instances where ACS performs better according to the I_H^- metric. Box-plots for I_H^- and $I_{\epsilon+}^1$ are given in Figure 13 and Figure 14. In general, these figures graphically confirm the superiority of the cooperative search methods for test instances having a high number of nodes. Thus, to summarize the results obtained by the hybrid search methods, the benefit of the cooperation scheme is being felt in most large-size test instances, but the addition of an auto-adaptive aspect does not seem to have a major influence on the results. Nevertheless, as pointed out in Table 14, the difference between the average number of times that IBMOLS is launched during the search process of PCS and ACS is relatively thin. This can be explained by the fact that (i) IBMOLS puts more time to find non-dominated solutions by starting with a population of poorer quality, what is the case in PCS comparing to ACS (at least for the first launch), and (ii) a part of the run-time allocated to the algorithm is used to compute a contribution value at every step of ACS, whereas PCS devotes all of its run-time in the search process. These two aspects lead to the fact that the number of times that IBMOLS is launched is, in the end, more or less balanced between both cooperative methods.

Comparison with Exact Single-objective Results. As a last step, we provide a comparison between the results found for the bi-objective version of the RSP proposed in this paper and the ones of the mono-objective RSP investigated in [34], where both costs are summed up. Note that it was not possible to compare our results to the ones of the other formulation of a single-objective RSP investigated in [35, 44, 47], where the assignment cost is subject to a constraint, due to the way the bound has been fixed. In [34], the authors propose an exact method to solve a single-objective RSP aiming at minimizing the sum of the ring cost and of the assignment cost. But, in order to provide optimal solutions visiting approximately 25, 50, 75 and 100% of the total number of nodes, they set the ring cost c_{ij} and the assignment cost d_{ij} between two nodes v_i and v_j in the following way: $c_{ij} = \lceil \alpha l_{ij} \rceil$ and $d_{ij} = \lceil (10 - \alpha) l_{ij} \rceil$ with $\alpha \in \{3, 5, 7, 9\}$, where l_{ij} denotes the distance between v_i and v_j given in the TSPLIB files. In order to make our results comparable to their, we transform the objective vectors of the best found Pareto front Z_N^* we obtained for every instance into a scalar value as detailed above, and we compare the minimal one with the optimum⁶ found in [34]. Additionally, we compare the best solution visiting every nodes we have found with the optimal TSP solution available in the TSPLIB website⁷. Table 15 gives the error ratio between the best known value and the best one we have found for every identified (single) objective and every benchmark test instance. In comparison to [34], this ratio is always under 1.5%, and is mostly below 1% for every instance. The optimum is even found for the *kroA100* and the *kroA150* instances with an $\alpha = 9$, and a better solution is found for the *kroA200* instances with an $\alpha = 3$ and an $\alpha = 9$. As regards to the optimal TSP solutions (corresponding to solutions visiting every node in the case of the RSP), our results are similar for instances with 200 nodes or less. For larger instances, they are slightly less good, especially for the *pr1002* instance,

⁶In fact, the authors imposed a time limit for their experiments. They report the best solution found so far for the instances exceeding this time limit, what is the case for the *kroA200* instance with an $\alpha = 3, 5$ and 9 in Table 15.

⁷<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.

Table 12: Performance comparison for I-IBMOLS, SEEA, PCS and ACS according to the I_H^- metric by using a Mann-Whitney statistical test. The “p-value” columns give the p-value of the statistical test. The “T” columns give the outcome of the statistical test for a significance level of 5%: either the results of the search method located at a specific row are significantly better than those of the search method located at a specific column (\succ), either they are worse (\prec), or there is no significant difference between both (\equiv).

		I-IBMOLS		SEEA		PCS	
		p-value	T	p-value	T	p-value	T
<i>eil51</i>	SEEA	$> 5\%$	\equiv				
	PCS	$> 5\%$	\equiv	$> 5\%$	\equiv		
	ACS	$> 5\%$	\equiv	$> 5\%$	\equiv	$> 5\%$	\equiv
<i>st70</i>	SEEA	0.011	\prec				
	PCS	0.013	\prec	$2.677 \cdot 10^{-4}$	\prec		
	ACS	$> 5\%$	\equiv	0.001	\prec	$> 5\%$	\equiv
<i>kroA100</i>	SEEA	0.004	\prec				
	PCS	$> 5\%$	\equiv	0.006	\prec		
	ACS	0.015	\prec	$1.053 \cdot 10^{-4}$	\prec	$> 5\%$	\equiv
<i>bier127</i>	SEEA	$3.151 \cdot 10^{-8}$	\prec				
	PCS	$> 5\%$	\equiv	$7.700 \cdot 10^{-8}$	\prec		
	ACS	$> 5\%$	\equiv	$1.377 \cdot 10^{-7}$	\prec	0.012	\prec
<i>kroA150</i>	SEEA	0.003	\prec				
	PCS	$1.639 \cdot 10^{-6}$	\prec	$3.663 \cdot 10^{-8}$	\prec		
	ACS	$1.259 \cdot 10^{-6}$	\prec	$2.114 \cdot 10^{-7}$	\prec	0.037	\prec
<i>kroA200</i>	SEEA	$5.483 \cdot 10^{-5}$	\prec				
	PCS	$4.387 \cdot 10^{-4}$	\prec	$1.192 \cdot 10^{-7}$	\prec		
	ACS	$9.330 \cdot 10^{-4}$	\prec	$8.914 \cdot 10^{-8}$	\prec	$> 5\%$	\equiv
<i>pr264</i>	SEEA	$> 5\%$	\equiv				
	PCS	0.003	\prec	0.015	\prec		
	ACS	0.012	\prec	0.058	\prec	$> 5\%$	\equiv
<i>pr299</i>	SEEA	$3.225 \cdot 10^{-7}$	\prec				
	PCS	$> 5\%$	\equiv	$4.941 \cdot 10^{-8}$	\prec		
	ACS	$> 5\%$	\equiv	$4.941 \cdot 10^{-8}$	\prec	$> 5\%$	\equiv
<i>pr439</i>	SEEA	0.009	\prec				
	PCS	$8.503 \cdot 10^{-6}$	\prec	$5.877 \cdot 10^{-6}$	\prec		
	ACS	$1.377 \cdot 10^{-7}$	\prec	$1.031 \cdot 10^{-7}$	\prec	0.033	\prec
<i>pr1002</i>	SEEA	$7.523 \cdot 10^{-6}$	\prec				
	PCS	$1.192 \cdot 10^{-7}$	\prec	$> 5\%$	\equiv		
	ACS	$1.868 \cdot 10^{-6}$	\prec	$> 5\%$	\equiv	$> 5\%$	\equiv

Table 13: Performance comparison for I-IBMOLS, SEEA, PCS and ACS according to the $I_{\epsilon+}^1$ metric by using a Mann-Whitney statistical test. The “p-value” columns give the p-value of the statistical test. The “T” columns give the outcome of the statistical test for a significance level of 5%: either the results of the search method located at a specific row are significantly better than those of the search method located at a specific column (\succ), either they are worse (\prec), or there is no significant difference between both (\equiv).

		I-IBMOLS		SEEA		PCS	
		p-value	T	p-value	T	p-value	T
<i>eil51</i>	SEEA	> 5%	\equiv				
	PCS	> 5%	\equiv	> 5%	\equiv		
	ACS	> 5%	\equiv	> 5%	\equiv	> 5%	\equiv
<i>st70</i>	SEEA	> 5%	\equiv				
	PCS	> 5%	\equiv	> 5%	\equiv		
	ACS	> 5%	\equiv	> 5%	\equiv	> 5%	\equiv
<i>kroA100</i>	SEEA	> 5%	\equiv				
	PCS	> 5%	\equiv	> 5%	\equiv		
	ACS	0.007	\succ	0.006	\succ	> 5%	\equiv
<i>bier127</i>	SEEA	$3.146 \cdot 10^{-8}$	\prec				
	PCS	> 5%	\equiv	$3.151 \cdot 10^{-8}$	\succ		
	ACS	> 5%	\equiv	$4.935 \cdot 10^{-8}$	\succ	> 5%	\equiv
<i>kroA150</i>	SEEA	0.004	\prec				
	PCS	$1.378 \cdot 10^{-5}$	\succ	$1.192 \cdot 10^{-7}$	\succ		
	ACS	$1.747 \cdot 10^{-5}$	\succ	$3.708 \cdot 10^{-7}$	\succ	> 5%	\equiv
<i>kroA200</i>	SEEA	0.047	\prec				
	PCS	$6.124 \cdot 10^{-5}$	\succ	$6.428 \cdot 10^{-7}$	\succ		
	ACS	$7.067 \cdot 10^{-4}$	\succ	$8.503 \cdot 10^{-6}$	\succ	> 5%	\equiv
<i>pr264</i>	SEEA	$4.386 \cdot 10^{-5}$	\succ				
	PCS	$2.208 \cdot 10^{-5}$	\succ	> 5%	\equiv		
	ACS	$1.963 \cdot 10^{-5}$	\succ	> 5%	\equiv	> 5%	\equiv
<i>pr299</i>	SEEA	$3.556 \cdot 10^{-6}$	\prec				
	PCS	> 5%	\equiv	$1.259 \cdot 10^{-6}$	\succ		
	ACS	> 5%	\equiv	$1.974 \cdot 10^{-4}$	\succ	> 5%	\equiv
<i>pr439</i>	SEEA	0.018	\succ				
	PCS	$7.363 \cdot 10^{-7}$	\succ	$5.733 \cdot 10^{-8}$	\succ		
	ACS	$4.889 \cdot 10^{-7}$	\succ	$5.733 \cdot 10^{-8}$	\succ	> 5%	\equiv
<i>pr1002</i>	SEEA	$9.604 \cdot 10^{-6}$	\succ				
	PCS	$4.256 \cdot 10^{-8}$	\succ	$3.268 \cdot 10^{-4}$	\succ		
	ACS	$6.646 \cdot 10^{-8}$	\succ	0.006	\succ	> 5%	\equiv

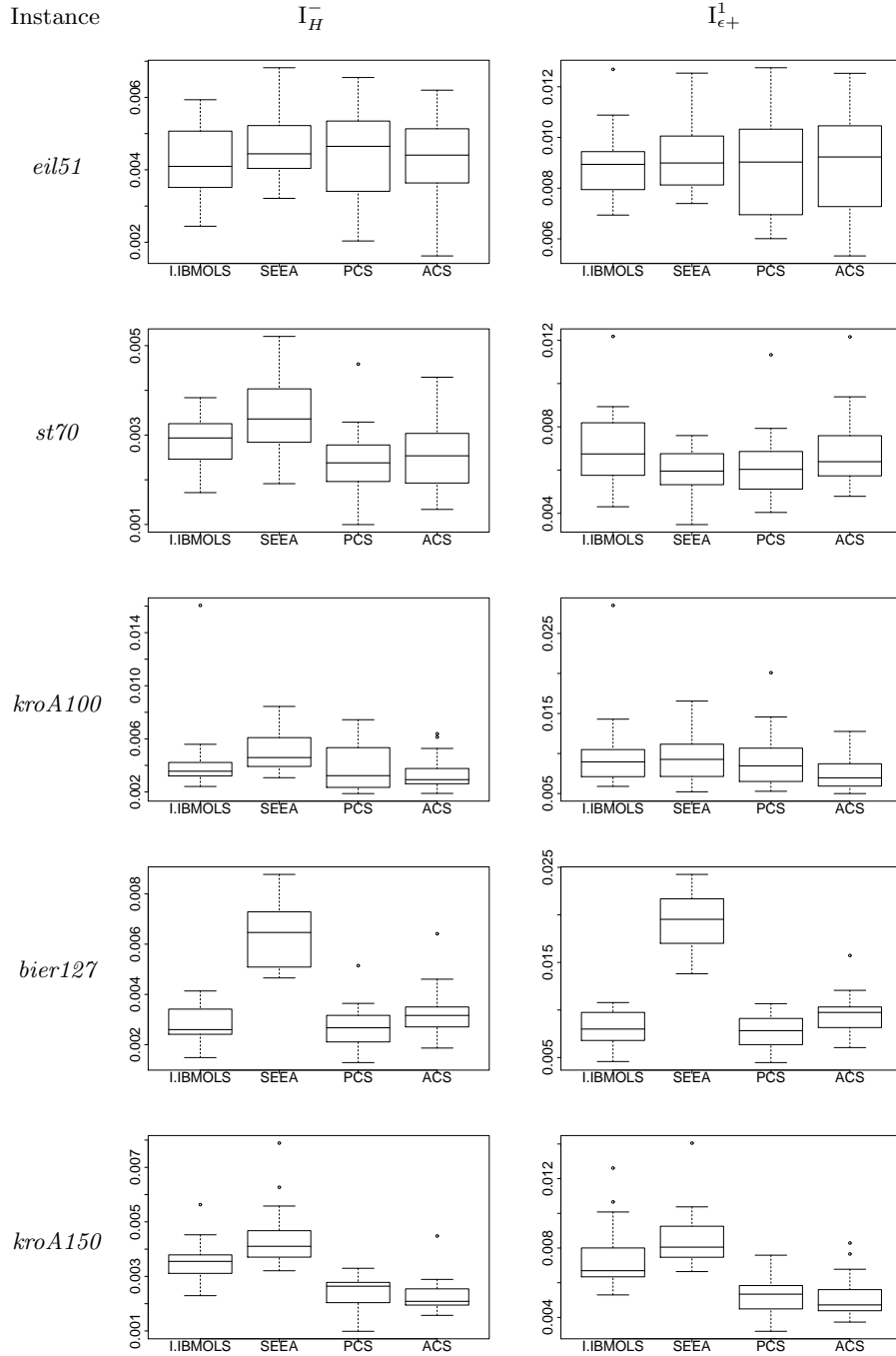


Figure 13: Performance comparison for I-IBMOLS, SEEA, PCS and ACS according to the I_H^- and the $I_{\epsilon+}^1$ metric (1).

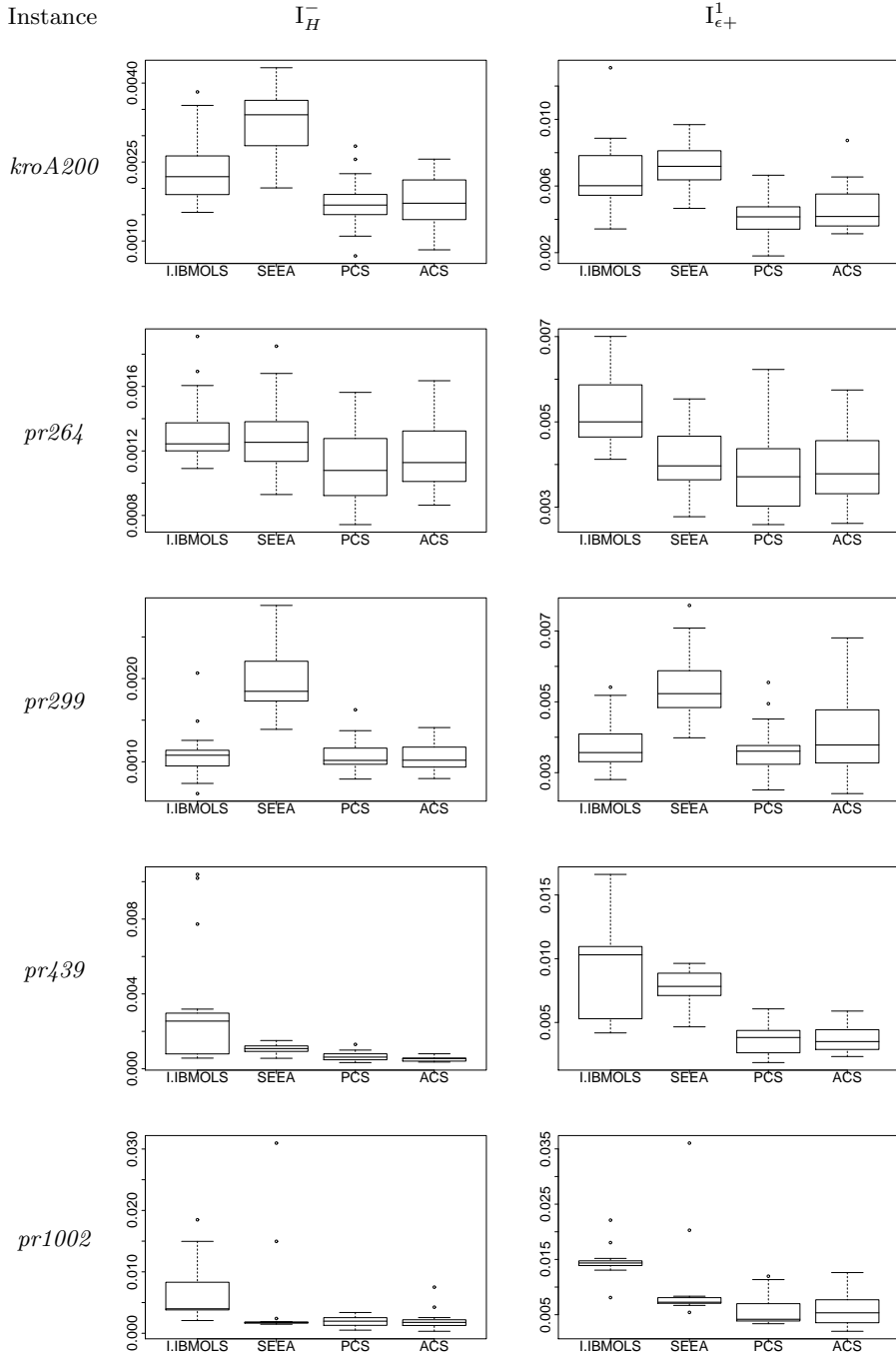


Figure 14: Performance comparison for I-IBMOLS, SEEA, PCS and ACS according to the I_H^- and the $I_{\epsilon+}^1$ metric (2).

Table 14: Average number of times that IBMOLS is launched during the search process of PCS and ACS.

Instance	PCS	ACS
<i>eil51</i>	9.35	8.70
<i>st70</i>	19.80	20.00
<i>kroA100</i>	4.60	6.25
<i>bier127</i>	4.70	3.50
<i>kroA150</i>	16.35	17.45
<i>kroA200</i>	23.15	22.15
<i>pr264</i>	43.00	43.15
<i>pr299</i>	38.90	26.90
<i>pr439</i>	18.05	12.80
<i>pr1002</i>	3.50	2.05

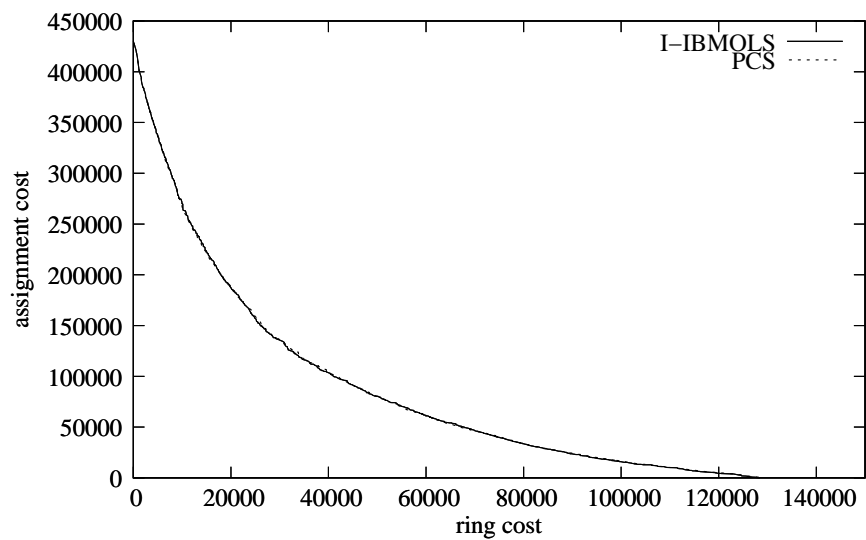


Figure 15: 90%-attainment surface plot obtained by the approximation sets found by I-IBMOLS and PCS for the *bier127* test instance.

where the error ratio is close to 10%. But, proportionally to the number of nodes, the running time available to solve this test instance is lower than for the other instances. To sum up, compared with single-objective optimal or near optimal results, the search methods we proposed in this paper to solve the B-RSP are quite promising with regard to the relatively small computational time available and to the size of the problem instances to be solved. Nevertheless, it is true that the solutions used to compare the results are the best ones we obtained during the set of all experiments we performed, and may not be as good for a single simulation run of a given search method.

Table 15: Error ratio between the ring cost value of the best known TSP solution and the best single objective value found in [34] in comparison to the best value found during our experiments.

Instance	Optimal TSP solution	Optimal single-objective RSP solution [34]			
		$\alpha = 3$	$\alpha = 5$	$\alpha = 7$	$\alpha = 9$
<i>eil51</i>	0.67%	0.67%	0.75%	0.37%	0.69%
<i>st70</i>	0.31%	0.31%	0.49%	0.56%	0.42%
<i>kroA100</i>	0.08%	0.08%	0.12%	0.56%	0.00%
<i>bier127</i>	0.64%	0.64%	0.34%	0.52%	0.01%
<i>kroA150</i>	1.40%	1.40%	1.11%	0.36%	0.00%
<i>kroA200</i>	1.22%	-4.82%	1.05%	0.81%	-1.44%
<i>pr264</i>	2.26%	-	-	-	-
<i>pr299</i>	1.72%	-	-	-	-
<i>pr439</i>	2.74%	-	-	-	-
<i>pr1002</i>	9.51%	-	-	-	-

5 Conclusion and Perspectives

A new multi-objective routing problem, the bi-objective ring star problem, has been investigated in this paper for the first time. It aims to locate a cycle through a subset of nodes of a graph while minimizing a ring cost, related to the length of the cycle, and an assignment cost, from non-visited nodes to visited ones. Despite it is clearly bi-objective, this problem has always been tackled in a single-objective formulation, either where both costs are combined [34], or where one cost is regarded as a constraint [35, 47]. As a first step, four population-based metaheuristics have been proposed to approximate the minimal set of efficient solutions for the problem under consideration. A population-based local search method, IBMOLS, recently proposed in [3], has first been designed, in an iterative way, with a variable neighborhood. Then, two state-of-the-art multi-objective evolutionary algorithms, namely IBEA [56] and NSGA-II [14], as well as a third one, the simple elitist evolutionary algorithm (SEEA), proposed here for the first time, have all been fitted for the particular case of the bi-objective ring star problem to be solved. Experiments were conducted using a set of multiple benchmark test instances. After having studied the influence of some parameters on the efficiency of the different search methods, we compared them to each other and we concluded that the iterative version of IBMOLS was, in general, significantly better than all the evolutionary algorithms on the problem

instances we tackled. Nevertheless, the behavior of a simple search method like SEEA in comparison to the ones of IBEA and NSGA-II is very encouraging with regard to combinatorial problem solving. As a second step, we designed two cooperative schemes, able to solve any kind of multi-objective optimization problems, between SEEA and the non-iterative version of IBMOLS. Both consist of using SEEA as the main search process and of launching IBMOLS regularly to intensify the search at a given time. The first hybrid metaheuristic, the periodic cooperative search (PCS), operates a systematic launching of IBMOLS at each of its step. The second one, the auto-adaptive cooperative search (ACS), launches IBMOLS at a given step of the algorithm only if a certain condition is verified. Thus, ACS evolves adaptively according to the search scenario and decides by itself, and online, when the cooperation must occur. In comparison to stand-alone metaheuristics, these two hybrid search methods statistically improves the results on a large number of test instances, and particularly on large-size ones. However, the efficiency difference between PCS and ACS is almost negligible. This can be explained by the fact that ACS spends time to compute whether the cooperation should occur at a given step of the algorithm. This is not the case for PCS, so that the latter devotes all of its computational time to the search process, even if a cooperation may start at an inopportune time, and then be less gainful.

Although the approaches proposed in this paper are already promising, even compared to the mono-objective approaches of the literature, a few future research directions are open. First is the possibility to improve the population initialization strategy used within all the proposed search methods. Indeed, the simple one that is used in the paper is a bit rudimentary, as each initial solution has approximately half of its nodes that belong to the cycle. Surely, an initial population having a well-spread number of visited nodes among its members should accelerate the convergence of the search methods toward a well-spread set of non-dominated solutions. Moreover, the use of a traveling salesman problem heuristic would be helpful in order to improve the ring cost of these initial solutions. Second, we pointed out that the crossover operator designed for the problem under consideration has a tendency to break the parent ring structures in the offspring individuals. Then, as proposed in [47], we could employ the same heuristic to improve the ring cost of the newly generated solutions. Third, given that several mutation operators are used within the evolutionary algorithms, it is always difficult to determine the appropriate probability selection of each one of them. Then, a possibility would be to set these rates adaptively, during the search process, according to the efficiency of each operator. Such a strategy has already been proposed in [4] in the frame of multi-objective optimization. Note that the last two points could greatly improve the efficiency of the evolutionary algorithms compared to the local search method, what would also be beneficial for the hybrid methods. From a purely algorithmic point of view, it would be interesting to try other quality indicators than the ϵ -indicator within the fitness assignment scheme of IBMOLS. We know that this binary indicator, although experiencing some difficulties concerning the extreme points of the trade-off surface, performs in general much better than others [3]. But the indicator-based fitness assignment strategy using the unary hypervolume indicator recently proposed in [55] could be an interesting alternative to explore. Next, we saw that the stand-alone metaheuristics taking part in the cooperative approach proposed in this paper, SEEA and IBMOLS, can both be launched

independently, exchanging information via the archive containing the set of non-dominated solutions found so far. Then, analogously to what has been proposed in COSEARCH [50] for the mono-objective case, these two search agents can be launched in parallel while an adaptive memory stores different kind of information, including the archive. But, a similar generic model adapted to the case of multi-objective optimization still has to be established. Finally, we found out that SEEA was a good alternative to state-of-the-art evolutionary algorithms for multi-objective combinatorial optimization when a given, relatively small, amount of computational time is allowed. In the future, we plan to tackle other kinds of combinatorial problems within SEEA to verify if our observations are still valid, especially in the case where more than two objectives are involved. The same remark can be done with regard to the cooperative approaches we proposed.

Acknowledgment

A subpart of this work has already been published in [40]. The authors would like to gratefully acknowledge Matthieu Basseur and Edmund K. Burke for their helpful contribution.

References

- [1] U. Akinc and K. Srikanth. Optimal routing and process scheduling for a mobile service facility. *Networks*, 22(2):163–183, 1992.
- [2] R. Baldacci, M. Dell’Amico, and J. J. Salazar González. The capacitated m-ring star problem. *Operations Research*, 55(6):1147–1162, 2007.
- [3] M. Basseur and E. K. Burke. Indicator-based multi-objective local search. In *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 3100–3107, Singapore, 2007. IEEE Press.
- [4] M. Basseur, F. Seynhaeve, and E.-G. Talbi. Design of multi-objective evolutionary algorithms: Application to the flow shop scheduling problem. In *IEEE Congress on Evolutionary Computation (CEC 2002)*, pages 1151–1156, Piscataway, NJ, USA, 2002. IEEE Press.
- [5] J. Bean. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.
- [6] J. E. Beasley and E. M. Nascimento. The vehicle routing-allocation problem: A unifying framework. *Top*, 4(1):65–86, 1996.
- [7] 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.

-
- [8] E. K. Burke and G. Kendall, editors. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer-Verlag, Berlin, Germany, 2005.
- [9] 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.
- [10] D. W. Corne, J. D. Knowles, and M. J. Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *Conference on Parallel Problem Solving from Nature (PPSN VI)*, volume 1917 of *Lecture Notes in Computer Science (LNCS)*, pages 839–848, Paris, France, 2000. Springer-Verlag.
- [11] J. R. Current and D. A. Schilling. The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, 73:114–126, 1994.
- [12] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.
- [13] K. Deb. Multi-objective optimization. In E. K. Burke and G. Kendall, editors, *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, chapter 10, pages 273–316. Springer-Verlag, Berlin, Germany, 2005.
- [14] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [15] T. C. S. Dias, G. F. de Sousa Filho, E. M. Macambira, L. A. F. Cabral, and M. H. C. Fampa. An efficient heuristic for the ring star problem. In C. Álvarez and M. J. Serna, editors, *Proceedings of the 5th International Workshop on Experimental Algorithms (WEA 2006)*, volume 4007 of *Lecture Notes in Computer Science (LNCS)*, pages 24–35, Menorca, Spain, 2006. Springer-Verlag.
- [16] K. Doerner, A. Focke, and W. J. Gutjahr. Multicriteria tour planning for mobile healthcare facilities in a developing country. *European Journal of Operational Research*, 179(3):1078–1096, 2007.
- [17] M. Ehrgott and X. Gandibleux. Approximative solution methods for multi-objective combinatorial optimization. *Top*, 12(1):1–89, 2004.
- [18] M. Ehrgott and X. Gandibleux. Hybrid metaheuristics for multi-objective combinatorial optimization. In C. Blum, M. J. Blesa, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics - An emergent approach for optimization*, volume 114 of *Studies in Computational Intelligence*, chapter 9. Springer-Verlag, Berlin, Germany, 2008. (to appear).
- [19] 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.
- [20] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms (ICGA 1993)*, pages 416–423, Urbana-Champaign, IL, USA, 1993. Morgan Kaufmann.
- [21] X. Gandibleux, N. Mezdaoui, and A. Fréville. A tabu search procedure to solve multiobjective combinatorial optimization problems. In R. Caballero, F. Ruiz, and R. Steuer, editors, *Advances in Multiple Objective and Goal Programming*, volume 455 of *Lecture Notes in Economics and Mathematical Systems*, pages 291–300, Berlin, Germany, 1997. Springer-Verlag.
- [22] F. W. Glover and G. A. Kochenberger, editors. *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, MA, USA, 2003.
- [23] M. P. Hansen. Tabu search in multiobjective optimisation: MOTS. In *Proceedings of the 13th International Conference on Multiple Criteria Decision Making (MCDM 1997)*, Cape Town, South Africa, 1997.
- [24] M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations of the non-dominated set. Technical Report IMM-REP-1998-7, Institute of Mathematical Modeling, Technical University of Denmark, 1998.
- [25] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello, and J. Molina. Pareto-adaptive epsilon-dominance. *Evolutionary Computation*, 15(4):493–517, 2007.
- [26] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):392–403, 1998.
- [27] N. Jozefowicz, F. Semet, and E.-G. Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309, 2008.
- [28] J. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *IEEE Congress on Evolutionary Computation (CEC 2002)*, pages 711–716, Piscataway, NJ, USA, 2002. IEEE Press.
- [29] 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).
- [30] J. D. Knowles and D. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- [31] M. Labbé and G. Laporte. Maximizing user convenience and postal service efficiency in post box location. *Belgian Journal of Operations Research, Statistics and Computer Science*, 26:21–35, 1986.

-
- [32] M. Labbé, G. Laporte, and I. Rodríguez Martín. Path, tree and cycle location. In T.G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 187–204. Kluwer Academic Publishers, Boston, MA, USA, 1998.
- [33] M. Labbé, G. Laporte, I. Rodríguez Martín, and J. J. Salazar González. The median cycle problem. Working paper, CRT-99-29, Université de Montréal, 1999.
- [34] 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.
- [35] 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.
- [36] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.
- [37] M. Laumanns, E. Zitzler, and L. Thiele. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science (LNCS)*, pages 181–196, Zurich, Switzerland, 2001. Springer-Verlag.
- [38] Y. Lee, S.Y. Chiu, and J. Sanchez. A branch and cut algorithm for the steiner ring star problem. *International Journal of Management Science*, 4:21–34, 1998.
- [39] A. Liefoghe, 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.
- [40] A. Liefoghe, L. Jourdan, M. Basseur, E.-G. Talbi, and E. K. Burke. Metaheuristics for the bi-objective ring star problem. In J. Van Hemert and C. Cotta, editors, *Eighth European Conference on Evolutionary Computation in Combinatorial Optimisation (EvoCOP 2008)*, volume 4472 of *Lecture Notes in Computer Science (LNCS)*, pages 206–217, Napoli, Italy, 2008. Springer-Verlag.
- [41] A. Mauttone, S. Nesmachnow, A. Olivera, and F. Robledo. A hybrid metaheuristic algorithm to solve the capacitated m-ring star problem. In *International Network Optimization Conference (INOC 2007)*, Spa, Belgium, 2007.
- [42] H. Meunier, E.-G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimization. In *IEEE Congress on Evolutionary*

- Computation (CEC 2000)*, pages 317–324, San Diego, USA, 2000. IEEE Press.
- [43] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Kluwer Academic Publishers, Boston, MA, USA, 1999.
- [44] 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.
- [45] L. Paquete, M. Chiarandini, and T. Stützle. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter 7, pages 177–199. Springer-Verlag, Berlin, Germany, 2004.
- [46] G. Reinelt. TSPLIB – A traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991.
- [47] 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.
- [48] J. D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the 1st International Conference on Genetic Algorithms (ICGA 1985)*, pages 93–100, Pittsburgh, PA, USA, 1985. Lawrence Erlbaum Associates.
- [49] E.-G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(2):541–564, 2002.
- [50] E.-G. Talbi and V. Bachelet. COSEARCH: A parallel cooperative metaheuristic. *Journal of Mathematical Modelling and Algorithms*, 5(2):5–22, 2006.
- [51] E. L. Ulungu, J. Teghem, P. Fortemps, and D. Tuyttens. MOSA method: a tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis*, 8(4):221–236, 1999.
- [52] C. L. Valenzuela. A simple evolutionary algorithm for multi-objective optimization (SEAMO). In *IEEE Congress on Evolutionary Computation (CEC 2002)*, pages 717–722, Piscataway, NJ, USA, 2002. IEEE Press.
- [53] L. Vogt, C. Poojari, and J. Beasley. A tabu search algorithm for the single vehicle routing allocation problem. *Journal of the Operational Research Society*, 58(4):467–480, 2007.
- [54] J. Xu, S. Y. Chiu, and F. Glover. Optimizing a ring-based private line telecommunication network using tabu search. *Management Science*, 45(3):330–345, 1999.

-
- [55] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. 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 862–876, Matsushima, Japan, 2007. Springer-Verlag.
- [56] 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.
- [57] 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.
- [58] 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.

Contents

1	Introduction	3
2	The Bi-objective Ring Star Problem	3
2.1	Multi-objective Optimization	4
2.2	Problem Definition	5
2.3	Related Works	5
2.4	Industrial Interests	7
3	Metaheuristics to Solve the Bi-objective Ring Star Problem	7
3.1	A Multi-objective Local Search	7
3.2	Multi-objective Evolutionary Algorithms	9
3.2.1	IBEA	9
3.2.2	NSGA-II	10
3.2.3	SEEA	10
3.3	Application to the Ring Star Problem	11
3.3.1	Solution Encoding	11
3.3.2	Population Initialization	11
3.3.3	Neighborhood and Mutation Operators	12
3.3.4	Crossover Operator	12
3.4	Experiments	13
3.4.1	Experimental Protocol	13
3.4.2	Local Search Parameter Analysis	15
3.4.3	Evolutionary Algorithms Parameter Analysis	17
3.4.4	Comparison between Search Methods	17
4	A Cooperative Approach	26
4.1	Motivation	26
4.2	Cooperative Schemes	27
4.3	Related Works	28
4.4	Experiments	29
4.4.1	Parameter Setting	29
4.4.2	Computational Results and Discussion	30
5	Conclusion and Perspectives	37



Centre de recherche INRIA Lille – Nord Europe
Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex

Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier

Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex

Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex

Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex

Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur

INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)

<http://www.inria.fr>

ISSN 0249-6399