http://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Articles in Advance, pp. 1–14
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

A New Scatter Search Design for Multiobjective Combinatorial Optimization with an Application to Facility Location

A. D. López-Sánchez,[a] J. Sánchez-Oro,[b] M. Laguna[c]

[a] Pablo de Olavide University, 41013 Sevilla, Spain; [b] Rey Juan Carlos University, 28933 Madrid, Spain; [c] Leeds School of Business, University of Colorado, Boulder, Colorado 80309

Contact: adlopsan@upo.es (ADL-S); jesus.sanchezoro@urjc.es (JS-O); laguna@colorado.edu, https://orcid.org/0000-0002-8759-5523 (ML)

Received: March 22, 2019
Revised: October 16, 2019; January 13, 2020; January 21, 2020
Accepted: January 25, 2020
Published Online in Articles in Advance: September 18, 2020

https://doi.org/10.1287/ijoc.2020.0966

Abstract. Scatter search (SS) is a well-established metaheuristic solution methodology that has seen most of its success in single-objective optimization. The literature includes a few examples of the SS methodology adapted to multiobjective optimization, almost all dealing with continuous, nonlinear problems. We describe an SS design that we believe has general applicability in the area of multiobjective combinatorial optimization and show its effectiveness by applying it to a facility location problem. Facility location consists of identifying the best locations for a set of facilities. The set of best locations may vary substantially according to the objective function employed to solve the optimization problem. We employ a facility location problem with multiple objectives (mo-FLP) to test our design ideas for a multiobjective optimization scatter search. We focus on the objective functions associated with three well-known location problems in the literature: the *p*-Median Problem (pMP), the Maximal Coverage Location Problem (MCLP), and the *p*-Center Problem (pCP). Our computational experiments are configured to show that the proposed SS design is capable of producing high-quality Pareto-front approximations.

Summary of Contribution: Metaheuristic optimization is at the heart of the intersection between computer science and operations research. The *INFORMS Journal on Computing* has been fundamental in advancing the ideas behind metaheuristic methodologies. Fred Glover's *Tabu Search–Part I* was published more than 30 years ago in the first volume of the then *ORSA Journal on Computing*. This article, one of the most cited in the area of heuristic optimization, paved the way for many contributions to the methodology and practice of operations research. As a continuation of this stream of research, we describe a new scatter search design for multiobjective optimization. The design includes a short-term memory tabu search and a path relinking combination method. We show how the strategies and mechanisms within scatter search and tabu search can be combined to produce a highly effective approach to multiobjective optimization.

History: Accepted by Alice Smith.
Funding: J. Sánchez-Oro was partially supported by the Spanish Ministry of Science, Innovation, and Universities (MCIU/AEI/FEDER, UE) [Grant PGC2018-095322-B-C22] and the Community of Madrid and Structural Funds of the European Union [Grant P2018/TCS-4566]. A. D. López-Sánchez was supported by the Spanish Ministry of Economy, Industry, and Competitiveness [Project ECO2016-76567-C4-1-R]. This work was also supported by the Spanish Ministry of Economy and Competitiveness [Grant TIN2015-65460-C2-2-P].
Supplemental Material: The online supplement is available at https://doi.org/10.1287/ijoc.2020.0966.

Keywords: multiobjective optimization • *p*-median • *p*-center • maximal coverage location • scatter search • path relinking • tabu search

1. Introduction

Scatter search (SS), originally proposed by Glover (1977), is a population-based metaheuristic that has been successfully applied to many optimization problems. SS relies on the idea that systematic designs and strategies for generating solutions usually lead to better results than those found with procedures that rely heavily on randomization. Although SS starts with a large population of solutions, contrary to other evolutionary algorithms, it iterates on a small set of solutions that are deliberately selected to maintain a balance between search diversification and intensification. SS has established itself as a bona fide optimization methodology in the operations research literature (Sánchez-Oro et al. 2015, 2016).

Our goal is to design an SS for multiobjective combinatorial optimization that adheres to the principles of the original methodology and that can be shown to be effective. We chose a facility location problem with multiple objectives as the test case for our ideas. The interest in solving locations problems is endorsed by the significant number of studies and applications in several areas of operations research. The literature includes descriptions of a variety of optimization problems for

locating hospitals, fire stations, police stations, military compounds, refugee camps, rescue centers, banks, ports, airports, warehouses, supermarkets, schools, pharmacies, and electric vehicle charging stations, among many others. The definition of "best" in the facility location literature varies, as it depends on the idiosyncrasies of each particular problem and its solution requirements.

We have organized the description of our work and experiments as follows. Section 2 reviews the literature related to multiobjective scatter search approaches. Section 3 describes the multiobjective location problem that we use as a test case for our scatter search design. Section 4 details our SS implementation and the strategies that we have developed for multiobjective optimization. Section 5 describes our experiments and the computational results. Our conclusions are in Section 6.

2. Multiobjective Scatter Search in the Literature

Although SS was originally conceived as a solution approach for single objective optimization problems, it has been adapted to tackle multiobjective optimization problems. We provide a brief review of twelve applications of scatter search to multiobjective optimization problems (see Table 1). Our descriptions focus on the distinguishing features of each SS implementation.

SSMO (Nebro et al. 2005) is an adaptation of scatter search to the optimization of constrained and unconstrained nonlinear problems with multiple objective functions. The main contributions of SSMO relate to the use of Pareto dominance, ranking, and crowding in order to select reference solutions. The authors also test the idea of applying clustering to identify centroids of highly ranked solutions. These centroids become candidates for the reference set. A simple dominance test is used to update the reference set. AbYSS (Nebro et al. 2008) is a marginal extension of SSMO (Nebro et al. 2005), which (1) adds an external archive to store all nondominated solutions found during the search, (2) uses genetic operators (such as crossover and mutation), and (3) incorporates two different density estimators. Unfortunately, Nebro et al. (2008) do not compare the performance of AbYSS and SSMO, and therefore it is not known whether the proposed extensions resulted in a measurable performance difference. Bong et al. (2012) adapt AbYSS for multiobjective clustering in in medical image segmentation.

Table 1. List of Multiobjective Scatter Search in our Literature Review

Reference	Application
Nebro et al. (2005)	Nonlinear optimization
Vasconcelos et al. (2005)	Nonlinear optimization
Beausoleil (2006)	Nonlinear optimization
Beausoleil (2007)	Nonlinear optimization
Molina et al. (2007)	Nonlinear optimization
Nebro et al. (2008)	Nonlinear optimization
Baños et al. (2009)	Looped water distribution networks
Rao and Lakshmi (2009)	Hybrid laminate composite structures
Caballero et al. (2011)	Clustering
Lwin et al. (2013)	Portfolio optimization
Guo and Liu (2014)	Selective disassembly sequencing
Guo et al. (2019)	Selective disassembly sequencing

The M-scatter search (Vasconcelos et al. 2005) is another SS adaptation to nonlinear multiobjective optimization. The design of M-scatter search mimics the standard procedure in (Laguna and Martí 2005, Guo and Liu 2014) but uses ranking and a niched penalty in order to update the reference set, and an external archive to store nondominated solutions.

MOSS (Beausoleil 2006) and MOSS-II (Beausoleil 2007) combine scatter and tabu search in the context of multiobjective nonlinear optimization. The multistart tabu search is used as the diversification generator every time the reference set needs to be rebuilt. This results in a design with two distinct phases: the tabu search phase and the scatter search phase. MOSS-II incorporates a constraint-handling mechanism and, in the scatter search phase, creates combinations of only those solutions that are both nondominated and feasible. Like MOSS and MOSS-II, SSPMO (Molina et al. 2007) tackles multiobjective nonlinear optimization problems with a combination of tabu and scatter search. Computational experiments in Molina et al. (2007) show that SSPMO is capable of finding better approximations of the Pareto front than various approaches in the literature, including MOSS.

In terms of SS adaptations to combinatorial optimization problems with multiple objectives, we are aware of an extension of SSPMO to clustering (Caballero et al. 2011). Tabu search is used in the initial phase in order to generate a diverse set of solutions from which the first reference set is built. It is also used as the improvement method within the scatter search iterations. In addition, the procedure uses a tabu memory to keep a record of past reference solutions in order to forbid these solutions from belonging to the reference set in future iterations.

Baños et al. (2009) approach the design of a water distribution network as a multiobjective optimization problem. The single-objective problem consists of selecting pipe diameters for a predetermined pipe layout in order to meet demand requirements at a minimum cost. The authors propose to find Pareto optimal solutions that minimize cost and maximize reliability. Their SS implementation uses a single reference set, which is initialized with the best (according to Pareto dominance) solutions in a randomly generated population and the most diverse solutions according to Euclidean distances. Although not explicitly mentioned,

it can be conjectured that the reference set is updated by applying Pareto dominance rules. The improvement method is based on a simulated annealing search. An external archive is used to record nondominated solutions, which in turn become the approximation of the Pareto front.

Rao and Lakshmi (2009) adapt scatter search to the problem of finding an optimal layup sequence of laminate composite structures. The authors formulate several multiobjective optimization models of this problem and find approximations of the optimal Pareto fronts with their SS adaptation. The population of solutions is randomly generated, and the initial reference set consists of two subsets of equal size, one consisting of nondominated solutions from the population and the other one consisting of diverse solutions. Pairwise combinations are limited to solutions within each subset. That is, the method does not combine solutions across the two subsets. The combination of solutions is done with a two-point crossover operator that is common in genetic algorithms. Combined solutions are improved with a search that is based on exchanges that seek "local optimality." However, it is not clear whether all search directions are explored (i.e., one direction for each objective function) or a compromise function is used to determine a single search direction. The reference set is updated with the selection of the best, nondominated solutions from the union of the reference set solutions and the new trial solutions. It is not specified how many of these solutions are retained. The set is completed with nondominated solutions that are diverse with respect to the solutions already selected.

Multiobjective SS has also been applied to portfolio optimization. The portfolio selection problem is concerned with the optimal allocation of a limited capital among available assets that involve various level of risk. Markowitz's mean-variance model for the portfolio selection problem deals with two objectives: maximizing mean returns while minimizing the variance of the returns. In this model, variance is used as a measure of risk. Lwin et al. (2013) apply scatter search to the problem of finding portfolios that are Pareto efficient with respect to these two objectives. The main feature of this SS adaptation consists of a reference set update method that employs an external archive. The external archive of nondominated solutions is updated every time a new solution is produced—e.g., when generating the initial population (randomly) and when combining and improving solutions. The reference set is divided into two subsets of equal size. One subset contains the best solutions (according to Pareto dominance) from the external archive and the other contains the solutions with the smallest crowding measure. After each SS iteration, the reference set is updated with the solutions that are currently in the external archive. The procedure overall departs from the fundamental tenets of the SS methodology and relies heavily on randomization. For instance, the subset generation method is random and the improvement method mimics the genetic algorithm notion of random mutations.

Reuse and remanufacturing of used products in order to protect the environment requires a sequence of disassembly operations. Disassembly modeling and planning have been approached as optimization problems in graphs. Guo and Liu (2014) formulate the selective disassembly sequence problem as a biobjective model that seeks to maximize profit and minimize disassembly time. The authors develop a scatter search that initiates with a randomly generated population of solutions. The improvement method is based on exchanges of positions in the sequence, and all solutions are tested for inclusion in an external archive of nondominated solutions. The initial reference set contains three subsets, one with the best solutions for the first objective function, another one with the best solutions for the second objective function, and a third one with diverse solutions. No details are given regarding the dynamic updating of the reference set. The subset generation consists of all solution pairs, considering all subsets. Combination is done using PPX (precedence preserving crossover), a well-known operator in the genetic algorithm literature (Bierwirth and Mattfeld 1999).

Guo et al. (2019) propose an extension of the disassembly model in (Guo and Liu 2014). The main addition consists of a third objective function; therefore, the model seeks to minimize energy consumption, maximize profit, and minimize disassembly time. The design is essentially the same as in (Guo and Liu 2014) with the difference that the objective functions are treated lexicographically. The authors state that decision makers and mangers in this context prioritize the objective functions, given preference to minimizing energy consumption, followed by maximizing profit, and lastly minimizing disassembly time. Therefore, the reference set update method is designed to assess the merit of new trial solutions following the stated order of preferences. That is, a new solution is tested for inclusion in the reference set for energy consumption first. If rejected, then it is tested for its profit and finally for its disassembly time. If rejected by all three objective functions, then it is tested for diversity.

One of the main shortcomings of the SS adaptations reviewed above is that their use of a single reference set (excect for Guo et al. 2019 lexicographical reference sets) forces an interpretation of the notion of "best" in order to select reference solutions. If a dominance test is implemented, a trial solution is added to the reference set if the solution dominates at least one of the existing reference solutions. The trial

solution is also added if the current reference solutions do not dominate it and the reference set is not full. However, if the reference set is full, then a newly created nondominated solution that does not dominate any of the reference solutions is sent directly to the external archive. This means that the new solution is not given the opportunity to serve as a reference point and therefore never combined with other reference (nondominated) solutions in future iterations.

Another major shortcoming relates to the improvement method. MOSS and MOSS-II do not include an improvement method. Several of the improvement methods are based on random perturbations (mutation) to move from one solution to another as long as the mutated solution dominates the current solution. These random perturbations do not represent a deliberate process for finding solutions that either fill gaps in the Pareto front or stretch the front toward the extreme points (i.e., where the optimal solutions to the single-objective problems are). In addition, the randomness of this process represents a significant departure from the SS methodology.

We seek a new approach to adapting scatter search to multiobjective optimization that focuses on finding the best approximation of the Pareto front. The methods are designed for the specific purpose of identifying nondominated solutions and improving the set of these solutions during the search. As described in Section 4, the diversification generation method addresses each objective in the problem instead of simply producing random solutions. A new solution is improved in the direction of the objective functions that have not been used to create the solution. One reference set per objective function is used and pairwise combinations are limited to solutions proceeding from different sets. To test our design, we apply it to a multiobjective facility location problem.

3. Multiobjective Facility Location Problem

Formally, a location problem can be stated as follows. Let $G = (V, E)$ be a graph such that $V = V_f \cup V_d$, with $V_f = \{1, \ldots, m\}$ denoting the set of candidate facility locations ($|V_f| = m$) and $V_d = \{1, \ldots, n\}$ the set of demand points ($|V_d| = n$). There is a weight w_i associated with each demand point $i \in V_d$, which typically represents the volume of demand. The set of edges $E = \{(i, j) : i \in V_d, j \in V_f\}$ represents paths between demand points and candidate facility locations in such a way that edge (i, j) is the path between the demand point $i \in V_d$ and the candidate facility location $j \in V_f$. The length of path $(i, j) \in E$ is $d_{ij} \geq 0$. The goal is to choose p locations out of m candidate facilities in order to optimize one or more objective functions. Consequently, a different location problem emerges depending on the objective function under consideration. We point out that this definition excludes other types of location problems for which possible locations are not a priori restricted to a finite set of points, such as in competitive location models (Kress and Pesch 2012).

In the p-Median Problem (pMP), for instance, the objective function seeks to minimize the average weighted distance between demand points and their nearest facility. We will refer to this objective function as f_1. The pMP was first studied in the mid-1960s by Hakimi (1964, 1965). Since then, many authors have tackled this problem whether using exact methods or heuristics. For instance, García-López et al. (2002) proposed, for the pMP, several parallel-computing designs within a variable neighborhood search framework to reduce the computational time and increase the exploration of the search space. An interesting application of the pMP deals with the location of emergency vehicles (ambulances) in Perth City (Australia) using a simple algorithm that consists of a reduction heuristic and an exchange procedure (Dzator and Dzator 2013). Recently, Herda and Haviar (2017) proposed two hybrid genetic algorithms for finding high-quality solutions for pMP instances that are intractable with exact algorithms.

The Maximal Coverage Location Problem (MCLP) is another variant of the facility location problem. The MCLP consists of maximizing the total weighted number of demand points covered by the facilities. In this context, a demand point is covered if it is located within a specific distance, r, also known as the cover radius of a chosen facility. This problem is applicable to the planning of service and emergency facilities. We will refer to this objective function as f_2. The MCLP was first introduced by Church and ReVelle (1974). A recent study developed and tested a branch-and-bound algorithm that exploits the structure of the problem (Blanquero et al. 2016). Specifically, the authors introduced, within a geometric branch and bound, specialized data structures that successfully coped with the combinatorial nature of the problem.

Lastly, the p-Center Problem (pCP) is a variant that seeks to minimize the maximum distance between demand points and their nearest facilities. We will refer to this objective function as f_3. The first solution method for the pCP was based on a set-covering approach (Minieka 1970). The procedure systematically updates a set covering matrix that eventually converges to an optimal solution in a finite number of iterations. Some years later, Kariv and Hakimi (1979) showed that the pCP is NP-hard and proposed an enumeration algorithm. The first metaheuristic approaches for the pCP were based on variable neighborhood search and tabu search (Mladenović et al. 2003). Since then, various authors have focused on the

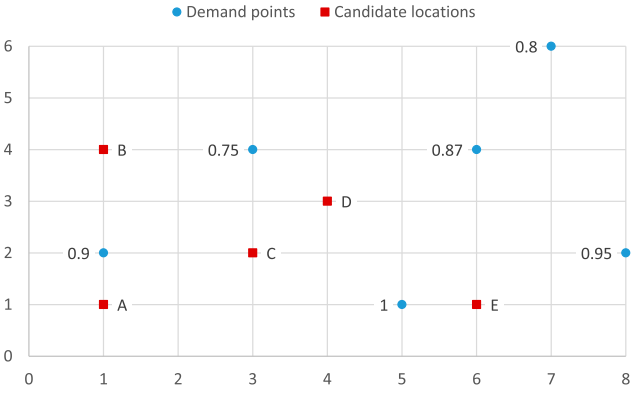
pCP from both theoretical and practical points of view. For a comprehensive review of the pCP, see Calik et al. (2015).

Our interest is in addressing simultaneously the three locations problems described above (pMP, MCLP, and pCP). We focus on the multiobjective facility location problem (mo-FLP) that results from considering the objective functions associated with these three location problems. The interest in tackling the multiobjective problem is to provide decision makers with the possibility of identifying solutions that consider the trade-off among various objective functions. In most situations, the three objectives that we are considering are not aligned, creating conflicts and trade-offs. Typically, no single solution is optimal with respect to the three objective functions. The goal is to find solutions for which it is not possible to improve the value of one objective function without deteriorating the value of at least another objective function. Those solutions are known as efficient solutions, nondominated solutions, or Pareto optimal solutions.

As discussed by Karatas and Yakici (2018), this particular mo-FLP is important in the location of emergency services. Consider, for example, the location of ambulance services. On the one hand, it is desired to minimize the average time of responding to emergencies. However, minimizing average response time tends to favor locations in the proximity of population centers, which neglects patients who are not close to the center. This can be addressed by minimizing the maximum distance between emergency stations and potential patients. Finally, many ambulance service providers operate with a predetermined critical response time threshold in order to mitigate the risk of fatalities. Then, they would like to maximize the number of patients who can be reached within the time threshold.

Figure 1 shows a simple facility location example that helps us illustrate the differences among the three objective functions. The example consists of $m = 5$ candidate facilities (red squares), $n = 6$ demand points (blue circles) with their corresponding weight, and $p = 2$ facilities to locate. Table 2 enumerates the 10 possible solutions, where the Facilities column shows the coordinates of the chosen locations in each solution. The f_1, f_2, and f_3 columns show the values of the three objective functions, where the best objective function values are highlighted in bold. Solution 10, with facilities at D and E, is best for f_1 and has an objective function value of 12.37. Solutions 7 and 9 achieve the best value of 2.85 for f_2 and correspond to locating facilities at B and E, and C and E, respectively. There are four solutions, number 3, 6, 8, and 10, that optimize f_3 with a value of 4.24. None of these solutions coincides with the optimal solution for f_2. Solution 10 is able to optimize two of the objective

Figure 1. (Color online) Candidate Facility Locations and Demand Points



functions, f_1 and f_3. However, in general, there is no guarantee that a single solution will be optimal for more than one of the three objective functions that we are considering. The results in Table 2 indicate that there is a strong preference for selecting the E location, as it appears in at least one of the optimal solutions for all of the objective functions. The strong candidates for the second location are C and D.

A recent study addresses the mo-FLP (Karatas and Yakici 2018). The authors considered the aforementioned objectives functions, developing an algorithm named ITER-FLOC, which consists of a hybrid approach that combines branch and bound techniques and iterative goal programming. Computational results show that ITER-FLOC performs well on small- to medium-sized problems. However, Karatas and Yakici (2018, p. 284) mention the following drawbacks:

> We must also note that, there are a number of disadvantages of the proposed methodology. In the experiments conducted, we apply an exact solution method to solve FLPs. This may not be a desirable option especially when the problem size is big.
>
> It should also be noted that, the algorithm requires that the decision maker must determine a set of input parameters such as the objective function value improvement thresholds, upper and lower bound improvement

Table 2. Enumeration of All Possible Solutions for the Example Depicted in Figure 1

Solution	Facilities	f_1	f_2	f_3
1	A and B	22.53	1.65	7.28
2	A and C	17.05	2.55	5.66
3	A and D	13.45	2.75	**4.24**
4	A and E	13.42	2.10	5.10
5	B and C	17.95	2.55	5.66
6	B and D	14.35	2.75	**4.24**
7	B and E	13.11	**2.85**	5.10
8	C and D	14.35	2.75	**4.24**
9	C and E	13.11	**2.85**	5.10
10	D and E	**12.37**	2.15	**4.24**

thresholds and resolution coefficients before starting iterative solution procedure. Choosing the values of these parameters is another challenge and disadvantage of the ITER-FLOC.

The motivation for our work is to show how scatter search can be adapted to multiobjective optimization and to test the merit of our ideas on the mo-FLP. Our SS adaptation aims at overcoming the aforementioned ITER-FLOC's limitations. In particular, we produce an approximation of the Pareto front with a single run and without requiring preference information. While our work follows the SS methodology (Laguna and Martí 2003, Martí et al. 2006), it introduces new strategies in the context of multiobjective problems that contrast with the hybrid SS designs described in the literature, which we have briefly reviewed above. Our main contributions are as follows:

- We show how to design a diversification generation method that considers simultaneously more than one objective function with the goal of constructing a population of solutions that balances solution quality and diversity.
- We develop a solution improvement method whose behavior depends on whether the reference set is being built from scratch or updated.
- The subset generation method departs from the traditional pairwise mechanism that operates on single reference set.
- We adapt path relinking as the combination method. This adaptation, which is new to the literature, takes into consideration the multiobjective nature of the problem.

Although our SS design is described in the context of the multiobjective facility location problem introduced above, we believe that the ideas have general applicability and can be extrapolated to other problems.

4. Proposed Multiobjective Scatter Search

The metaheuristic known as scatter search (SS) belongs to the family of the evolutionary programming methods. The procedure starts with the construction of a population of solutions from which a reference set (*RefSet*) is selected and evolved by means of combination and improvement mechanisms. The standard SS consists of five methods:

1. Diversification generation method
2. Reference set update method
3. Subset generation method
4. Solution combination method
5. Improvement method.

As described by Laguna and Martí (2003), there are standard (problem independent) implementations of the reference set update and the subset generation methods. For instance, a static update of the reference set is typical in most SS implementations. This method consists of constructing the *RefSet* by selecting solutions from a population created by the diversification generator. Half of the initial reference solutions are the best (according to their objective function values) solutions in the population and the other half are selected for diversity purposes. Diversity is measured by an appropriate distance metric (e.g., Euclidean or Hamming). After the initial construction, the *RefSet* is updated by selecting the solutions with the highest quality in the set of solutions consisting of the union of the current *RefSet* and the set of solutions that have been subjected to the improvement method after being generated by the solution combination method. Both the improvement and the combination methods are designed taking into consideration the problem context and the solution representation. A standard subset generation method consists of all pairs of reference solutions for which at least one of the two solutions is "new." That is, pairs that have already been examined in previous iterations are not considered.

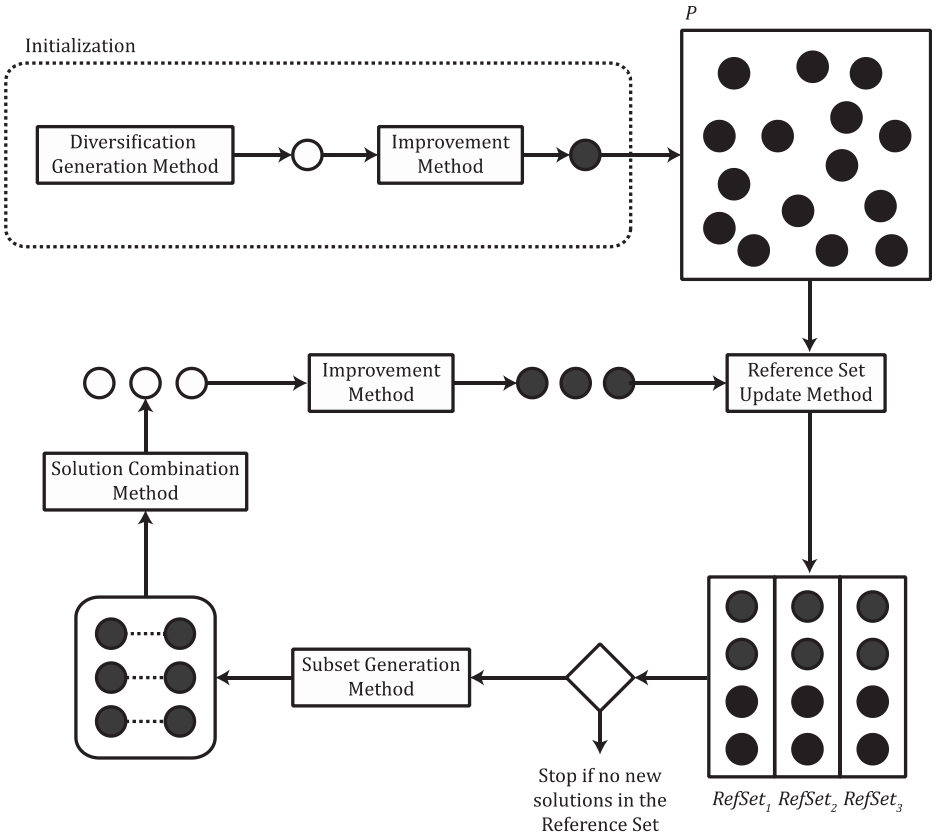
Figure 2 depicts the multiobjective SS framework that we propose. The procedure starts with the sequential application of the diversification generation and improvement methods (described in Sections 4.1 and 4.2), which result in a population P of feasible solutions. Then, we build one reference set (*RefSet*) for each objective function value. Each reference set consists of β solutions. The initial reference solutions are chosen from P. Half of the reference solutions are selected because of their quality (as measured by the corresponding objective function value) and half are selected to add diversity.

The selection of the most diverse subset of solutions is a hard problem on its own right. Instead of solving the maximum diversity problem, as done in Martí et al. (2009), we select solutions heuristically, one at a time. To measure solution diversity, we define the distance between two solutions S_1 and S_2 as the number of facilities selected in S_1 that are not selected in S_2, and vice versa. Then, the distance of a given solution S_i to the *RefSet* is evaluated as the minimum distance between S_i and every solution $S_j \in RefSet$. Formally,

$$d(S_i, RefSet) = \min_{S_j \in RefSet} d(S_i, S_j).$$

The distance $d(S_i, RefSet)$ is calculated for each solution $S_i \in P \backslash RefSet$. The solution with the maximum distance is selected and added to *RefSet*. This is repeated $\beta/2$ to build a *RefSet* with β solutions. As shown in Figure 2, our SS adaptation to multiobjective optimization employs three reference sets instead of one, as SS is traditionally implemented. Solution quality for

Figure 2. Scatter Search Framework



$RefSet_1$, $RefSet_2$, and $RefSet_3$ is measured according to f_1, f_2, and f_3, respectively. This partition of the reference solutions is a departure of the previous SS adaptations to multiobjective optimization problems, all of which operate on a single reference set.

We point out that the proposed search structure has some similarity with what is known in the literature as Vector Evaluated Genetic Algorithms (VEGA) (Schaffer 1985). VEGA divides the population of solutions into subpopulations, each of which evolves toward a single objective. The main difference between VEGA and our proposal is that in VEGA the elements in the subpopulations are shuffled to create a single population before applying the genetic operators. When dealing with a single-objective problem, VEGA behaves like a regular genetic algorithm (GA); therefore, it can be considered a generalization of a scalar GA. As it will become clear later, our SS is very deliberate about how solutions are chosen for combination from the reference sets, while VEGA operates under the typical GA (probabilistic) selection rules.

The subset generation method (Section 4.3) generates pairs of solutions that are the input to the solution combination method described in Section 4.4. The solution pairs (S_i, S_j) are constructed in such a way that S_i has not been combined with S_j in previous iterations and that S_i belongs to a reference set that is different from the reference set of S_j. We have observed that this design induces search diversification and increases the density of the Pareto front by closing gaps that occurred naturally when solutions are generated with only one objective function in mind. These gaps may occur in VEGA implementations because the population to which the genetic operators are applied is not stratified by the objective function.

New trial solutions are generated as combinations of reference solutions and they are subjected to the improvement method described in Section 4.2. Specifically, each solution S_i is improved in the direction of the three objective functions, generating solutions S_i^1, S_i^2, and S_i^3 for the first, second, and third objective functions, respectively. Every solution during the improvement phase is a candidate for the approximation of the Pareto front, which is kept as an external archive. The best solutions at the end of the improving phase become candidates for the *RefSet*. For instance, S_i^1 is added to $RefSet_1$ if $f_1(S_i^1)$ is better than the f_1 value of at least one of the solutions currently in $RefSet_1$. Let $S_j \in RefSet_1$ and $f_1(S_j) > f_1(S_i^1)$, then S_i^1 replaces the solution S_j for which $d(S_i^1, S_j)$ is minimum. That is, the trial solution to be added to *RefSet* replaces, among those solutions of inferior quality, the most similar reference solution. This replacement strategy adds diversification to each reference set.

4.1. Diversification Generation Method

The first stage of scatter search consists of building a population P of solutions. The literature shows that the best SS performance is achieved when P consists of solutions that balance quality and diversity. This is why the most successful SS implementations do not start with a population of solutions generated completely at random.

Solutions for a location problem consist of selecting a set S of p facilities. We use a single constructive procedure that is adapted to each objective function, creating $|P|/3$ solutions guided by pMP, $|P|/3$ solutions guided by MCLP, and $|P|/3$ solutions guided by pCP.

Solution construction is based on the Greedy Randomized Adaptive Search Procedure (GRASP) methodology. GRASP is an established, stand-alone metaheuristic (Feo and Resende 1989, Feo et al. 1994), but in our case we only use one of its major elements (namely, the solution construction phase) in order to generate well-balanced solutions. We start with a random selection of the first facility, among all possible candidate locations. Then, a Candidate List (CL) is created with all the remaining available locations. Subsequently, a Restricted Candidate List (RCL) is generated with the most promising candidates to host a facility. A facility is considered promising if its greedy function value exceeds a threshold τ, which is calculated as

$$\tau = g_{\max} - \alpha \cdot (g_{\max} - g_{\min}),$$

where $g_{\min}$ and $g_{\max}$ are, respectively, the minimum and maximum value of the greedy function g for all candidate facilities. The greedy function measures the attractiveness of a candidate facility as it relates to the change in the objective function value. Since pMP and pCP are minimization problems, the attractiveness of a facility is measured as the change in $1/f_1$ and $1/f_3$, respectively. The attractiveness of a facility in MCLP is directly related to the change in the value of f_2. Since a facility is selected at random among those facilities in RCL (i.e., those facilities with greedy values that at least τ), then the α parameter controls the level of randomness in the construction. A value of $\alpha = 1$ makes the construction totally random, because RCL = CL, while a value of $\alpha = 0$ makes it totally deterministic, because only the location with a maximum greedy value is considered. Once a facility is selected from the RCL, the CL is updated by removing the selected location. The construction process is repeated p times to create a feasible solution.

4.2. Improvement Method

The improvement method is applied in two different stages of the search, as shown in Figure 2. The first time is during the initialization phase. The $P/3$ solutions that are constructed guided by pMP are improved considering f_1, the $P/3$ solutions that are built guided by MCLP are improved according to f_2, and the $P/3$ solutions that are generated guided by pCP are improved following f_3. During this stage, the search focuses on each objective function separately. The improvement method is also applied after the solution combination method generates trial solutions, as described in Section 4.4. In this stage, solutions are improved in three different directions, each one guided by one of the objective functions. That is, each trial solution is subjected to the improvement method three times in order to find local optima with respect to f_1, f_2, and f_3. When optimizing with respect to one objective function, say f_1, the deterioration of the other two objective functions (in this case, f_2 and f_3) is permitted.

Our improvement method follows the tabu search methodology (Glover 1989, 1990; Glover and Laguna 1997). Tabu search can be viewed as an ordinary local or neighborhood search, proceeding iteratively from one solution to another until a termination criterion is satisfied. Embedding tabu search within a scatter search framework was first explored by Campos et al. (2005). Let S be a solution. We define $N(S)$ as the set of solutions that can be reached by replacing a facility $u \in S$ with a facility $v \in V_f \backslash S$; that is,

$$N(S) \leftarrow \{(u, v) : u \in S, v \in V_f \backslash S\}.$$

Tabu search uses memory structures to escape local optimality and explore the solution space by dynamically modifying the solution neighborhoods. We use a short-term memory structure, which is the most basic form of memory within tabu search. In our implementation, the memory structure consists of a list of facilities that have been included in the solutions in the last *Tenure* iterations, where *Tenure* is a search parameter. These facilities are forbidden from being included in the current solution, unless their inclusion leads to a solution that is better than the best solution visited during the current search. The overwriting of a tabu classification is called *aspiration criteria* within the tabu search methodology. The modified neighborhood $N^*(S)$ is the result of excluding from $N(S)$ all of the facilities that are in the short-term memory T; that is,

$$N^*(S) \leftarrow \{(u, v) : u \in S, v \in V_f \backslash S \cup T\}.$$

Our tabu search implements the first-improving strategy and therefore performs the first non-tabu move (i.e., the exchange of a facility $u \in S$ with a facility $v \in V_f \backslash S \cup T$, where T is the list of tabu facilities) that leads to an improvement of the current solution or a

tabu move that reaches the aspiration criterion. If no such move exists, then the tabu move that deteriorates the objective function the least is selected. After an exchange (u, v), facility u is added to T for *Tenure* iterations. The method ends when the search performs k consecutive non-improving moves.

4.3. Subset Generation Method

The SS methodology contemplates various designs for the generation of subsets of reference solutions. These designs attempt to balance search diversification and intensification. For example, when the best two reference solutions are chosen to form a subset, the intention is to intensify the search around elite solutions. The selection criterion could also be to pair solutions that are as dissimilar as possible in order to induce diversification. The cardinality of the subsets also plays a role in this balancing act.

In our multiobjective design, we have determined that pairing high-quality reference solutions is effective when quality is measured by two different objective functions. Therefore, we generate all pairs (S_i, S_j) of reference solutions for which the reference sets for S_i and S_j are different. Our combination mechanisms are designed to operate on two solutions and therefore we do not generate subsets of higher dimensions.

The subset generation, the combination method, and the improvement method must complement each other to achieve the ultimate goal of building high-quality Pareto fronts. In the previous section, we discussed how the improvement method focuses on each objective function separately. That is, the search for improved solutions does not consider more than one objective function at a time, as done in compromise programming and implemented in SSPMO (Molina et al. 2007, Caballero et al. 2011). A compromise solution seeks a balance among all objective function values, and the goal of compromise programming is finding a compromise set with the guidance of a global criterion that is based on minimizing the maximum distance to an ideal point.

Our subset selection strategy addresses the multiobjective nature of the problem by choosing elite solutions from two different sets. As we will see in the next section, these solutions are combined by a mechanism that marches from one solution to the other. Our computational experiments show that this design is effective in populating the archive of nondominated solutions with compromise points.

4.4. Solution Combination Method

The solution combination method operates on the pairs of references solutions generated by the subset generation method. We employ a form of path relinking (Glover 1997) as our solution combination method. Path relinking generates a sequence of neighborhood searches that link an initiating solution to a guiding solution. We generate paths between the solution pairs that result from the subset generation method. One is the initiating solution (S') and the other is the guiding solution (S''). Then, the sequence of solutions between both is produced by removing locations from S' that are not in S'' (the removal set) and inserting locations in S' that are in S'' (the insertion set).

Figure 3 shows an example of how the path relinking works. Let $S' = \{1,2,3,4,5\}$ and $S'' = \{2,3,6,8,9\}$ be the initiating and the guiding solution, respectively. The insertion set is $In = \{6,8,9\}$ and the removal set is $Out = \{1,4,5\}$. The process is a sequence of moves, where one facility from In is added to the solution and one facility from Out is removed. After the move, the chosen facilities are deleted from In and Out. This is done until In and Out are empty. Figure 3 shows the result of the relinking process when the selections are (9,5), (6,1), and (8,4).

Our path relinking design is a departure from the single-objective SS designs in the literature. In those designs, the next move is selected by exploring the neighborhood that results from all the moves in the direction of the guiding solutions. In a single-objective facility location problem, this would mean the selection of the best exchange of a facility from In and a facility from Out. That is, the best exchange according to the objective function value is selected among all those moves that transform the current solution into another that is closer to the guiding solution. The notion of best in single-objective problems is straightforward to implement.

In our path relinking, we evaluate all of the solutions that result from inserting a location from In and removing a location from Out. The resulting solutions are tested for membership in the nondominated set (i.e., the external archive). If at least one of the moves results in a nondominated solution, we arbitrarily choose one of the nondominated solutions to move to continue the relinking process. An arbitrary move is chosen from all available moves if no move results in a nondominated solution.

5. Computational Experiments

We perform experiments on three sets of problem instances, which we generated with the code provided by Karatas and Yakici (2018). Table 3 summarizes the parameter values of the three sets of problem instances.[1]

Figure 3. Path Relinking Example

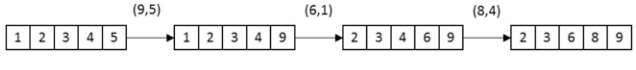


Table 3. Characteristics of Problem Instances

Parameter	Set 1 (small size)	Set 2 (medium size)	Set 3 (large size)
m	20	50	200
n	50	100	400
p	5	10	15
r	20	15	10

As done in previous studies reported in the literature, we randomly generate candidate facilities and demand points in a 100×100 area. The weight for the demand points is randomly chosen within the interval $[0.75, 1]$. We generated 80 random problem instances of each size for a total of 240 instances. Experiments were carried out on an Intel Core i7 920 (2.67 GHz) with 8 GB of RAM, and the algorithms were implemented in Java 8.

The multiobjective optimization literature includes various metrics to compare the quality of the Pareto front approximations constructed with heuristic procedures. The *number of nondominated solutions* found (Solutions). The more solutions in the approximation of the Pareto front, the better. The *hypervolume* (HV) calculates the volume in the objective space covered by the approximation of the Pareto front. The larger the volume, the better the approximation. The *coverage metric*, $C(A, B)$, calculates the proportion of solutions of algorithm B that are weakly dominated by the efficient solutions found by algorithm A. A common way of assessing performance with this metric is to compute $C(R, B)$, where R is the reference Pareto front[2] or the optimal front if known. If $C(R, B) = 1$, all solutions obtained by procedure B are weakly dominated by the reference Pareto front R. $C(R, B) = 0$ means that no solution from B is weakly dominated by the reference Pareto front R. Therefore, smaller C values indicate better performance. The epsilon indicator (ε-indicator) measures the smallest distance needed to transform every point obtained in the approximation of the Pareto front to a point that is nondominated by a reference Pareto front or the optimal front if known. A small ε-indicator value indicates a better approximation. CPU indicates the computing time required to obtain the approximation of the Pareto front. A shorter computational time is preferable. See Audet et al. (2018) for a comprehensive review and detailed description of performance indicators for multiobjective optimization procedures. We use the jMetal framework to evaluate these metrics (Durillo and Nebro 2011).

The remainder of the section is divided into three parts: preliminary experiments, competitive testing, and additional testing. The preliminary experiments focus on the selection of the best parameter settings for our scatter search, the competitive testing compares the performance of our tuned procedure against ITERFLOC (Karatas and Yakici 2018), and the additional testing compares our SS design with one from the literature.

5.1. Preliminary Experiments

In order to fine-tune our SS implementation, we select a representative subset of 45 instances, 15 from each set (see Table 3 above). This training set represents 18.75% of the total number of instances. A full factorial experimental design for the fine-tuning of the search parameters is impractical in terms of computing time. As stated in (Sánchez-Oro et al. 2016), a sequential process to determine parameter values has been shown to be effective. We have four parameters to adjust: α, β, *Tenure*, and k.

The scatter search literature includes numerous experimental studies with the goal of identifying effective values for the size of the reference set. The value of 10 is often used when tackling single-objective optimization problems. Since our design includes three reference sets, the standard value of 10 would result in 30 reference solutions and in too many solution pairs to combine (namely, 300). In order to balance computing effort with early convergence (i.e., the stage of the search when no new solutions are admitted to any of the reference sets), we chose the value of $\beta = 6$. This results in 108 pairs (when all reference solutions are new), which is in line with the amount of effort that SS implementations devote to this element of the solution process.

With the value of β set to 6, we tune the value of α, a parameter used for constructing solutions within the diversification generation method. We test $\alpha = \{0.25, 0.50, 0.75\}$ and α selected at random in each construction. Table 4 shows no significant performance differences among the α values, when considering the various metrics. Since no single α value seems to

Table 4. Summary of Results for the Tested α Values

α	Solutions	HV	$C(R, SS)$	ε-indicator	CPU
Random	25.8235	**0.5464**	**0.2097**	0.0553	57.49
0.25	24.6471	0.5460	0.2355	0.0450	**55.39**
0.50	**25.8824**	0.5450	0.2555	0.0520	56.24
0.75	25.1765	0.5454	0.2372	**0.0451**	60.12

dominate all others in terms of our metrics, we chose the random selection of α in each construction.

With α chosen at random, we calibrate the tabu list size (i.e., the value of *Tenure*). We make this value dependent on the number facilities and test $Tenure = 0.10 \cdot p$, $0.20 \cdot p$, and $0.30 \cdot p$ The results in Table 5 show a similar situation as Table 4, in the sense that no single value of the parameter (*Tenure*) dominates the others in terms of all performance metrics. We choose 0.1 following the principle that the smallest tabu tenure that avoids cycling is preferable in terms of adding search flexibility.

With α, β, and *Tenure* set to their best values, we determine the best value for the number of iterations without improvement, k. The value of this parameter is also made dependent of the number of candidate facility locations—specifically, we test $k = 0.10 \cdot m$, $0.50 \cdot m$, and $0.90 \cdot m$, as shown in Table 6.

As expected, Table 6 shows that the larger the value of k, the better the outcomes. However, the trade-off of an increased solution time must be taken into consideration. We compromise and choose an intermediate value, opting for 0.5 instead of 0.9, which achieves the best values for the metrics associated with solution quality.

5.2. Competitive Testing

This competitive testing consists of comparing the performance of our SS implementation with the performance of ITER-FLOC (Karatas and Yakici 2018). ITER-FLOC hybridizes two methodologies: branch-and-bound and iterative goal programming. Branch-and-bound is used to solve each single-objective problem iteratively, generating lower and upper bounds for each objective function as is customary in this process. Based on a user-controlled parameter, the procedure adds constraints to each single-objective model in order to force the solution method to find new optimal solutions for that model without deteriorating the values of the other two objective functions. ITER-FLOC requires the setting of a number of parameters, as specified in the article. Karatas and Yakici kindly shared their code with us, and our comparison uses the default configuration that they describe in their work and to which the code was set when we received it.

Table 7 summarizes the results of this experiment.[3] For each metric and problem size, the table shows two rows for each method. The first row is the average value of the metric. The second row consists of the percentage of instances in which the corresponding method obtains a better result than the competing method. Note that because of the instances when both methods obtain the same value, the sum of the fractions for a particular metric does not have to be equal to one. For example, ITER-FLOC obtains an average HV value of 0.227 for the medium-sized instances while our SS obtains 0.195. Under this metric statistic and for these instances, ITER-FLOC performs better than SS. However, although the average HV favors ITER-FLOC, SS obtains better individual HV values in more instance (47.5%) than ITER-FLOC (35%). This indicates that in 17.5% of the instances, both methods obtained the same HV value.

The metrics show that ITER-FLOC performs generally better than SS in the small instances. A similar conclusion can be drawn regarding the medium-sized problems. However, as we move to the large instances, the advantage of the heuristic search techniques within SS emerge. The results in the last section of Table 7 show that SS clearly dominates ITER-FLOC when tackling the large instances in our test set. SS generates two orders of magnitude more solutions than ITER-FLOC. All average values favor SS, in some cases by an order of magnitude (e.g., HV and the ε-indicator). Individual results also favor SS, with ITER-FLOC only able to produce one instance in which the coverage value is better than SS.

In terms of computational effort, ITER-FLOC employs two termination criteria. The procedure stops

Table 5. Summary of Results for the Tested *Tenure* Values

Tenure	Solutions	HV	$C(R, SS)$	ε-indicator	CPU
0.1	**25.9412**	**0.5567**	**0.1890**	0.0434	56.36
0.2	25.7647	0.5552	0.2010	**0.0274**	55.83
0.3	23.7059	0.5516	0.3013	0.0609	**48.65**

Table 6. Summary of Results for the Tested k Values

k	Solutions	HV	$C(R, SS)$	ε-indicator	CPU
0.1	26.7647	0.5332	0.3119	0.0698	**61.71**
0.5	30.3529	0.5395	0.2070	0.0381	313.58
0.9	**30.6471**	**0.5411**	**0.1648**	**0.0371**	501.08

Table 7. Summary of Results for ITER-FLOC and SS

Method	Solutions	$C(R, B)$	HV	ε-indicator	CPU
Small					
ITER-FLOC	4.040	0.064	0.124	0.658	17.8
	24.0%	54.7%	36.0%	52.0%	0.0%
SS	6.613	0.250	0.147	0.729	0.1
	65.3%	10.7%	52.0%	36.0%	100.0%
Medium					
ITER-FLOC	8.038	0.063	0.227	0.607	22.6
	22.5%	30.0%	35.0%	55.0%	26.3%
SS	14.913	0.094	0.195	0.692	8.5
	70.0%	15.0%	47.5%	30.0%	73.8%
Large					
ITER-FLOC	3.225	0.425	0.233	0.640	3969.3
	0.00%	1.25%	0.00%	0.00%	3.75%
SS	137.025	0.014	0.700	0.046	2169.2
	100.0%	78.8%	96.3%	96.3%	96.3%

when any of the single-objective models become infeasible due to the addition of the constraints that prevent the deterioration of the objective functions that are not part of that model. The procedure also stops when the compromise solution is not able to improve by at least a user-specified epsilon value. This solution is such that "all of the three models have approximately the same performance for each objective." In contrast, SS stops when no solutions are included in the Pareto front after a complete SS iteration. In addition, our SS implementation takes advantage of the multicore computer systems by parallelizing the construction, local search, and combination phase of SS. In particular, each construction, improvement, and combination is executed in an independent thread, which enables the scalability of the proposal.

When taking into consideration both the quality of the Pareto fronts (as measured by the metrics in Table 7) and the required computational effort, it seems clear that the proposed SS becomes the preferred method as the problem size increases. We attribute the difference in performance to the following characteristics of our SS implementation:

- The balance between intensification and diversification in both the construction and the improvement methods results in dense Pareto front approximations.
- The combination complementary designs of the combination and improvement methods are largely responsible for improving the quality of the solutions in the approximation of the Pareto front.

5.3. Additional Testing

The computational results in the previous section show that our approach is highly competitive when compared with a specialized procedure for the multiobjective facility problem described in Section 3. What those experiments are not able to show is whether our proposed SS design improves upon what has already appeared in the scatter search literature. Our literature review revealed that there is no "standard" scatter search for multiobjective optimization. However, from the adaptations described in Section 2, we believe that AbYSS (Nebro et al. 2008) is the one that provides the best contrast to our proposal. AbYSS was originally developed for nonlinear optimization, but we adapted to our context in order to produce a meaningful comparison. We used the same set of test problems and report a summary of the results in Table 8.

Table 8. Summary of Results for AbYSS and SS

Method	Solutions	$C(R,B)$	HV	ε-indicator	CPU
Small					
AbYSS	6.387	0.277	0.135	0.753	2384.868
	10.7%	46.7%	33.3%	5.3%	0.0%
SS	6.613	0.250	0.147	0.729	0.110
	22.7%	28.0%	46.7%	14.7%	100.0%
Medium					
AbYSS	8.938	0.780	0.137	0.771	2405.795
	7.5%	5.0%	0.0%	0.0%	0.0%
SS	14.913	0.094	0.195	0.692	8.485
	77.5%	93.8%	80.0%	50.0%	100.0%
Large					
AbYSS	13.825	0.978	0.095	2.260	2625.632
	2.50%	0.00%	0.00%	0.00%	12.50%
SS	137.025	0.014	0.700	0.046	2169.195
	96.3%	100.0%	96.3%	100.0%	87.5%

The results in Table 8 show that the two SS implementations perform at a similar level when dealing with the small problems. This similar performance quickly ends when moving to the medium-sized problems and then to the large instances. For the large instances, our proposed SS produces average results that are at least one order of magnitude better than those produced by AbYSS. To give AbYSS an opportunity to find improved approximations of the Pareto front, we ran a final test where we executed AbYSS for up to four times longer than our procedure. We observed no significant changes in the results reported in Table 8.

6. Conclusions

Our motivation for this project was twofold: exploring new ideas for the adaptation of scatter search in the context of multiobjective combinatorial optimization and developing a state-of-the-art solution method for the multiobjective facility location problem (mo-FLP). In terms of advancing the state of the art for the mo-FLP, the computational results included here indicate that our scatter search is able to produce approximations of the Pareto front that existing procedures are not able to deliver. As the problem size increases, the quality of these approximations surpasses the approximations found by the current best procedure in the literature (ITER-FLOC). Quality is measured with various indicators, all of which favor our SS adaptation in the test with the largest problem instances.

The main algorithmic design ideas include a unique structure and management of the reference sets, multidirectional searches in the improvement method, and a path relinking mechanism to combine solutions. These ideas and the structure of our SS implementation are a departure from most of what has been tried before in terms of applying scatter search to multiobjective optimization. We firmly believe that the proposed design is generalizable and that the effectiveness that we observed when we applied it to the mo-FLP can be expected when tackling other multiobjective optimization problems.

Acknowledgments

The authors wish to express their gratitude to Professors Karatas and Yakici for sharing their code. This is a refreshing gesture of collegiality and willingness to advance our field.

Endnotes

[1] All data sets can be found here: https://github.com/jesussanchezoro/ProblemInstances.

[2] The reference Pareto front, denoted by R, is constructed with all of the solutions found during the testing of all procedures. That is, R represents the best-known approximation of the Pareto front.

[3] The values obtained for each metric by each procedure on each instance can be found in the online supplement associated with this article.

References

Audet C, Bigeon J, Cartier D, La Digabel S, Salomon L (2018) Performance indicators in multiobjective optimization. *Optim. Online*. Accessed April 22, 2020, http://www.optimization-online.org/DB_FILE/2018/10/6887.pdf.

Baños R, Gil C, Reca J, Martinez J (2009) Implementation of scatter search for multi-objective: A comparison study. *Comput. Optim. Appl.* 42:421–441.

Beausoleil RP (2006) MOSS multiobjective scatter search applied to non-linear multiple criteria optimization. *Eur. J. Oper. Res.* 169(2): 426–449.

Beausoleil RP (2007) MOSS-II tabu/scatter search for nonlinear multiobjective optimization. Siarry P, Michalewicz Z, eds. *Advances in Metaheuristics for Hard Optimization* (Springer, Berlin), 39–67.

Bierwirth C, Mattfeld DC (1999) Production scheduling and rescheduling with genetic algorithms. *Evolutionary Comput.* 7(1):1–18.

Blanquero R, Carrizosa E, Tóth BG (2016) Maximal covering location problems on networks with regional demand. *Omega* 64:77–85.

Bong CW, Lam HY, Khader AT, Kamarulzaman H (2012) Adaptive multi-objective archive-based hybrid scatter search for segmentation in lung computed tomography imaging. *Engrg. Optim.* 44(3):327–350.

Caballero R, Laguna M, Martí R, Molina J (2011) Scatter tabu search for multiobjective clustering problems. *J. Oper. Res. Soc.* 62(11):2034–2046.

Calik H, Labbé M, Yaman H (2015) p-Center problems. Laporte G, Nickel S, Saldanha da Gama F, eds. *Location Science* (Springer, Cham, Switzerland), 79–92.

Campos V, Laguna M, Martí R (2005) Context-independent scatter and tabu search for permutation problems. *INFORMS J. Comput.* 17(1):111–122.

Church R, ReVelle C (1974) The maximal covering location problem. *Papers Regional Sci. Assoc.* 32(1):101–118.

Durillo JJ, Nebro AJ (2011) jMetal: A Java framework for multi-objective optimization. *Adv. Engrg. Software* 42(10):760–771.

Dzator M, Dzator J (2013) An effective heuristic for the p-median problem with application to ambulance location. *OPSEARCH* 50:60–74.

Feo T, Resende M (1989) A probabilistic heuristic for a computationally difficult set covering problem. *Oper. Res. Lett.* 8(2):67–71.

Feo T, Resende M, Smith S (1994) A greedy randomized adaptive search procedure for maximum independent set. *Oper. Res.* 42(5):860–878.

García-López F, Melián-Batista B, Moreno-Pérez JA, Moreno-Vega JM (2002) The parallel variable neighborhood search for the p-median problem. *J. Heuristics* 8(3):375–388.

Glover F (1977) Heuristics for integer programming using surrogate constraints. *Decision Sci.* 8(1):156–166.

Glover F (1989) Tabu search—Part I. *ORSA J. Comput.* 1(3):190–206.

Glover F (1990) Tabu search—Part II. *ORSA J. Comput.* 2(1):4–32.

Glover F (1997) Tabu search and adaptive memory programming—Advances, applications and challenges. Barr RS, Helgason RV, Kennington JL, eds. *Interfaces in Computer Science and Operations Research—Advances in Metaheuristics, Optimization, and Stochastic Modeling Technologies*, Operations Research/Computer Science Interfaces, vol. 7 (Springer, Boston), 1–75.

Glover F, Laguna M (1997) *Tabu Search* (Springer, New York)

Guo X, Liu S (2014) A scatter search approach for multiobjective selective disassembly sequence problem. *Discrete Dynam. Nature Society* 2014:Article 756891.

Guo X, Zhou M, Liu S, Qi L (2019) Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints. *IEEE Trans. Cybernetics*, ePub ahead of print March 27, https://doi.org/10.1109/TCYB.2019.2901834.

Hakimi SL (1964) Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper. Res.* 12(3): 450–459.

Hakimi SL (1965) Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Oper. Res.* 13(3):462–475.

Herda M, Haviar M (2017) Hybrid genetic algorithms with selective crossover for the capacitated p-median problem. *Central Eur. J. Oper. Res.* 25(3):651–664.

Karatas M, Yakici E (2018) An iterative solution approach to a multiobjective facility location problem. *Appl. Soft Comput.* 62:272–287.

Kariv O, Hakimi S (1979) An algorithmic approach to network location problems. I: The p-centers. *SIAM J. Appl. Math.* 37(3):513–538.

Kress D, Pesch E (2012) Sequential competitive location on networks. *Eur. J. Oper. Res.* 217(3):483–499.

Laguna M, Martí R (2003) *Scatter Search: Methodology and Implementations in C* (Springer, New York).

Laguna M, Martí R (2005) Experimental testing of advanced scatter search designs for global optimization of mutimodal functions. *J. Global Optim.* 33(2):235–255.

Lwin K, Qu R, Zheng J (2013) Multi-objective scatter search with external archive for portfolio optimization. Rosa A, Dourado A, Madani K, Filipe J, Kacprzyk J, eds. *5th Internat. Joint Conf. Comput. Intelligence* (SciTePress, Vilamoura, Algarve, Portugal), 111–119.

Martí R, Duarte A, Laguna M (2009) Advanced scatter search for the max-cut problem. *INFORMS J. Comput.* 21(1):26–38.

Martí R, Laguna M, Glover F (2006) Principles of scatter search. *Eur. J. Oper. Res.* 169(2):359–372.

Minieka E (1970) The m-center problem. *SIAM Rev.* 12(1):138–139.

Mladenović N, Labbé M, Hansen P (2003) Solving the p-center problem with tabu search and variable neighborhood search. *Networks* 42(1):48–64.

Molina J, Laguna M, Martí R, Caballero R (2007) SSPMO: A scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS J. Comput.* 19(1):91–100.

Nebro AJ, Luna F, Alba E (2005) New ideas in applying scatter search to multiobjective optimization. Gaspar-Cunha A, Antunes CH, Coello Coello C, eds. *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, vol. 9018 (Springer, Berlin), 443–458.

Nebro AJ, Luna F, Alba E, Dorronsoro B, Durillo JJ, Beham A (2008) AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Trans. Evolutionary Comput.* 12(4):439–457.

Rao AM, Lakshmi K (2009) Multi-objective optimal design of hybrid laminate composite structures using scatter search. *J. Composite Materials* 43(20):2157–2182.

Sánchez-Oro J, Laguna M, Duarte A, Martí R (2015) Scatter search for the profile minimization problem. *Networks* 65(1):10–21.

Sánchez-Oro J, Laguna M, Martí R, Duarte A (2016) Scatter search for the bandpass problem. *J. Global Optim.* 66(4):769–790.

Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. Grefenstette JJ, ed. *1st Internat. Conf. Genetic Algorithms* (Lawrence Erlbaum Associates, Hillsdale, NJ), 93–100.

Vasconcelos JA, Maciel JH, Parreiras RO (2005) Scatter search techniques applied to electromagnetic problems. *IEEE Trans. Magnetics* 41(5):1804–1807.