Scatter Search for the Bandpass Problem

JESÚS SÁNCHEZ-ORO
Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Spain.
jesus.sanchezoro@urjc.es

MANUEL LAGUNA
Leeds School of Business, University of Colorado at Boulder, USA
laguna@colorado.edu

RAFAEL MARTÍ
Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain
Rafael.Marti@uv.es

ABRAHAM DUARTE
Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Spain.
Abraham.Duarte@urjc.es

ABSTRACT

We tackle a combinatorial problem that consists of finding the optimal configuration of a binary matrix. The configuration is determined by the ordering of the rows in the matrix and the objective function value is associated with a value B, the so-called bandpass number. In the basic version of the problem, the objective is to maximize the number of non-overlapping blocks containing B consecutive cells with a value of one in each column of the matrix. We explore variants of this basic problem and use them to test heuristic strategies within the scatter search framework. An existing library of problem instances is used to perform scientific testing of the proposed search procedures to gain insights that may be valuable in other combinational optimization settings. We also conduct competitive testing to compare outcomes with methods published in the literature and to improve upon previous results.

Keywords: bandpass problem, scatter search, path relinking, telecommunications, metaheuristics.

Version: May 18, 2016

1. Introduction

The bandpass problem is associated with early-generation DWDM (Dense Wavelength Division Multiplexing) systems in fiber optic networks [16] [18] [27]. This technology used the *band* concept to allow the transmission of several wavelengths in a single fiber-optic cable. Wavelengths enter or leave the fiber optic cables via a set of devices known as OADMs (Optical Add/Drop Multiplexers), which are installed along the fiber optic cable at origin and destination points [16] [18]. OADMs contain special cards that control the wavelengths and identify which wavelengths should pass through the device and which should be dropped. Typically, several wavelengths pass through the same OADM. The wavelengths may be handled as a block when they are consecutive. A block of wavelengths is known as a bandpass. Without this assembling of wavelengths, a card is needed for each wavelength. However, when a bandpass is created, a single card can handle the entire block of wavelengths. The number of wavelengths in the block is referred to as the bandpass number (B). Therefore, the process of creating a wavelength block results in a reduction of cards from B to 1 and hence a cost reduction in the design and maintenance of the fiber optic network. The optimization opportunity is to assign wavelengths in order to maximize the number of bandpasses with a bandpass number of B.

This bandpass problem was first described by Bell and Babayev in the Annual INFORMS Meeting held in Denver Colorado in October 2004 and the first publication appeared in 2009 [1]. Since then, newer generations of DWDM systems have been developed to add more flexibility. This newer equipment employs reconfigurable optical ADMs (ROADMs) for which the band concept does not apply. The ROADM is a dynamic wavelength arrangement scheme enabled by wavelength selective switching. These new systems, however, are much more expensive and therefore the older systems still have a market in small DWDM deployments.

The bandpass problem consists of creating blocks of wavelengths that need to be sent from a set of origins to a set of destinations. The goal of the blocks is to reduce the number of devices needed for the transmission of the wavelengths, resulting in a decrease of both installation and maintenance costs. The telecommunication network is modeled as a $m \times n$ binary matrix A, where a_{ij} is equal to 1 if wavelength i must be delivered to destination j. In the basic version of the problem, the value of B (i.e., the bandpass number) is given. A bandpass is formed by B consecutive 1s in the same column. Each value of 1 can only be used for one bandpass and therefore two distinct bandpasses in the same column cannot have a common element. Figure 1.a shows a network with 6 wavelengths and 5 destinations. The network shows that, for instance, wavelength 1 must be delivered to all but the third destination. Assuming that $B = 3$, the matrix in Figure 1.a contains 3 bandpasses, as indicated by the gray blocks, where O refers to the ordering of the wavelengths, that is, $O(i)$ is the index of the wavelength assigned to row i.

	1	2	3	4	5
1	1	1	0	1	1
2	1	0	1	0	1
3	1	1	1	0	1
4	1	1	0	1	0
5	1	1	1	0	1
6	0	0	1	1	1

a) O =(1,2,3,4,5,6)

	1	2	3	4	5
1	1	1	0	1	1
2	1	0	1	0	1
3	1	1	1	0	1
5	1	1	1	0	1
4	1	1	0	1	0
6	0	0	1	1	1

b) O =(1,2,3,5,4,6)

Figure 1 Bandpasses for two possible assignments of wavelengths to rows.

In Figure 1.a, the bandpass for column 1 is shown as rows 1, 2, and 3. However, the bandpass for that column could also be formed by rows 2 to 4 or 3 to 5. Regardless of which one is chosen, there can only be one bandpass in column 1. In order to increase the number of bandpasses, the assignment of wavelengths to rows could be changed. This assignment could also be thought of as a reordering of the rows in the matrix. For instance, Figure 1.b shows the matrix that results by assigning wavelength 4 to row 5 and wavelength 5 to row 4. In other words, wavelengths 4 and 5 have exchanged positions from the original lexicographical order shown in Figure 1.a. The reassignment results in an increase of bandpasses from 3 to 4.

For a given bandpass number B, the bandpass problem consists of finding an ordering O^* of the rows that maximizes the number of bandpasses. The upper bound on the number of bandpasses in column j is given by:

$$\bar{b}_j = \left\lfloor \frac{\sum_i a_{ij}}{B} \right\rfloor$$

Therefore, if $f(O)$ is the number of bandpasses in solution O, then the objective function value of an optimal solution O^* is bounded as follows:

$$f(O^*) \leq \sum_j \bar{b}_j$$

For $B = 3$, the upper bound for the example in Figure 1 is 5 bandpasses. It can be easily verified that this upper bound is achieved by the optimal solution $O = (5,4,1,6,3,2)$.

Babayev, Bell, and Nuriyev [1] refer to the problem described above as BP1. They also suggest an alternative version of the problem, denoted by BP2, which reflects some specific network technologies. In this second version, a total of up to $G + 1$ groups of rows are created, where the first G groups have B

rows. That is, the first group consists of rows 1 to B, the second group consist of rows $B+1$ to $2B$, and so on. If the remainder of the m/B quotient is greater than zero, then an additional group with the last $m-GB$ rows is created. The value of G is determined by:

$$G = \left\lfloor \frac{m}{B} \right\rfloor$$

A bandpass in BP2 is defined as a column within a group for which all its elements are nonzero. If the group is one of the first G, then the bandpass contains B nonzero elements. For group $G+1$, if it exists, the bandpass contains $m-GB$ elements with nonzero values. As mentioned in [1], this model reflects some specific equipment constraints. In contrast with BP1, the group of rows that can form bandpasses are all the same across columns. Figure 2 shows an optimal solution to BP2 for the example matrix shown in Figure 1. This solution corresponds to the row ordering $O = (2,3,6,1,4,5)$. Note that because $m = 6$ and $B = 3$ then $G = 2$ and the remainder of m/B is zero. Therefore, there are only 2 groups, one consisting of the first three rows in the matrix and the second one consisting of rows 4 to 6.

	1	2	3	4	5
2	1	0	1	0	1
3	1	1	1	0	1
6	0	0	1	1	1
1	1	1	0	1	1
4	1	1	0	1	0
5	1	1	1	0	1

Figure 2 Optimal solution to BP2 for matrix in Figure 1.

For BP2, the order of the rows within each group is not relevant and thus, for instance, $O = (2,3,6,1,4,5)$ is the same solution as $O = (6,3,2,5,4,1)$ or as any ordering where the first three rows contain the wavelengths 2, 3, and 6. A permutation vector clearly is not an effective solution representation for BP2 given that it creates a search space with $m!$ solutions while the number of unique solutions for a BP2 instance with G groups of size B is given by:

$$\frac{m!}{(B!)^G G!}$$

In the example of Figure 2, there are 720 permutations of the rows but only 10 unique ways of creating two groups of 3 rows. A more direct representation is given by S for which $S(i)$ is the index of the group to which wavelength i is assigned.

Both BP1 and BP2 assume that the value of B is given. Kurt et al. [19] argue that fixing the value of B may not be optimal and therefore they suggest an optimization model in which B becomes a decision variable. However, since the range of relevant values for B is expected to be somewhat limited, the optimization process to determine the best value for B can be achieved by simply solving BP1 multiple times. We do not address this variant as a separate version of the problem.

The third version that we include in our study is the one introduced by Nuriyev, Kutucu, and Kurt [25] and for which Gürsoy and Nnuriyev [17] developed a genetic algorithm. This is the so-called multi bandpass problem (MBP). In this version, instead of a single B value across the entire network, each destination j has its own B_j value. The assumption is that ADMs may be programmed independently from each other with different bandpass numbers. In the MBP, the input data include the B_j values for $j = 1, \dots n$. An extension of the MBP, suggested by Kurt et al. [19], turns the B_j values into variables. This extension is referred to as the "Optimization of Lengths in the Multi-Bandpass Problem". The optimization problem involves not only finding the best ordering of the rows in the matrix but also the best bandpass number for each destination. The literature does not include any experimentation on this problem and it is not clear whether the extension corresponds to an actual technology for which the B_j values can be optimized within a given range. We do not address this variant here.

The literature on the bandpass problem is limited to seven articles [1][2][17][19][21][22][25]. Mathematical models for BP1 and BP2 are presented in [1] along with a NP-hard proof of BP1. Babayev, Bell, and Nuriyev [1] developed two solution procedures: an exact polynomial algorithm for BP1 with $n = 2$ and a heuristic for the general case. The authors also report optimal solutions for BP1 problem instances with $m \leq 32$ and $n \leq 8$, found by solving a mathematical programming model. A description of a generator of problem instances is provided and used to produce 90 instances with $m = 64$ and 96 and $n = 8, 12, 16$, and 25. These instances are divided into two groups of 45 instances each, one for which the optimal solutions are known (OS instances) and one for which the optimal solutions are not known (BKS instances). Collectively, these problems are referred to as the Library of Bandpass Problems (BPLIB)[1]. No experiments are performed for BP2 in [1]. Additional mathematical proofs for special cases of BP1 are provided in [21] and [22].

Berberler and Gürsoy [2] developed four variants of a genetic algorithm and applied them to the 45 OS instances in BPLIB. Optimality gaps are reported for all variants in all problem instances. Several GA variants are also developed and tested in [19]. This article introduces a new set of problems for the MBP. The data include both the B_j values and the binary A matrix[2]. Finally, some of the same authors in [1] and [19] are co-authors of [25] where mathematical models for BP1, BP2, and MBP are presented. Some of these models already appeared in [1] and [19] and some are new. No additional experimentation or results are included in [25].

Our current investigation into this problem class includes BP1, BP2, and MBP. We use these problems as a platform to test heuristic strategies within the scatter search (SS) framework. An existing library of

[1] The Library of Bandpass Problems can be found here http://sci.ege.edu.tr/~math/BandpassProblemsLibrary/.
[2] Problem instances introduced in [19] are found here http://fen.ege.edu.tr/arifgursoy/mbpopt/.

problem instances is used to perform scientific testing to identify the contributions of several search strategies. In particular, we develop and test:

- Two diversification generation methods
- Four improvement methods
- Two combination methods
- A scatter search method, SS1, for which solutions are represented by an ordering O of the wavelengths and therefore equipped to tackle BP1 and MBP instances.
- A scatter search method, SS2, for which solutions are represented as an assignment S of wavelengths to groups and that it is configured to solve BP2 instances.

We also conduct competitive testing to compare the solutions obtained with SS1 and SS2 with the best methods published in the literature for these problems (DSR [1], GAs [2] and [17]). In addition, we compare performance against commercial optimizers, one based on metaheuristic technology (LocalSolver) and the other built on exact procedures(Gurobi).

2. Standard Scatter Search Methods

Scatter search (SS) [13][20] is a population-based metaheuristic framework that consists of methods that generate, maintain, and transform a reference set of solutions ($RefSet$). The methods are organized as shown in Algorithm 1. Basic and advanced SS implementations have been developed for a variety of problems (see, e.g. [5][10][23][26][29]). In this section, we describe the implementation of the standard methods within the SS framework that we developed for BP1, BP2, and MBP. Two solution representations are used: 1) O, the ordering of the wavelengths (for BP1 and MBP); and 2) S, the group assignment (for BP2). Since the search strategies are linked to the solution representation, the same SS procedure is used to search for solution to BP1 and MBP.

```
Diversification generator
Improvement
Initial RefSet
do
    Subset generator
    Combination
    Improvement
    RefSet update
until termination is criteria satisfied
```

Algorithm 1 Scatter search template.

Following the template in Algorithm 1, we implemented standard methods for creating the initial $RefSet$, for generating solution subsets, and for updating the $RefSet$. The diversity generator and the improvement method (which will be described in Sections 3 and 4, respectively) are applied first to produce a population of P feasible solutions to the problem. Then, the initial $RefSet$ (of size β) is built by first selecting the $\beta/2$ best (in terms of the objective function value) solutions from P. The selected

solutions are removed from P. Then, the $\beta/2$ most diverse solutions with respect to those currently in the $RefSet$ are selected from P. Because the selection of the most diverse group of elements out of a set is a hard problem (see e.g. [7]), the solutions are heuristically selected one at a time.

Any diversification strategy requires a measure of distance between any two solutions. The distance for each solution representation is measured as follows.

$$p(O, O') = \sum_{i=1}^{m} |O(i) - O'(i)|$$

$$c(S, S') = m - \text{number of common elements}$$

$p(O, O')$ represents the so-called positional distance in permutation vectors (see Das and Roberts [4]), where $O(i)$ is the wavelength in row i. To calculate $c(S, S')$, we count the number of common assignments between both solutions by matching the groups in S with the groups in S'. For each group and its match, we identify the wavelengths that are assigned to both (i.e., the common elements). The matching problem has the following form:

Maximize	$\sum_k \sum_l c_{kl} x_{kl}$	
Subject to	$\sum_l x_{kl} = 1$	$\forall k$
	$x_{kl} \in \{0,1\}$	$\forall k, l$

In this problem, $x_{kl} = 1$ when group k in solution S is matched with group l in solution S'. The c_{kl} value indicates the number of common elements in groups k and l. The total number of common elements is given by cx^*, where x^* is an optimal solution to the matching problem. Then, $c(S, S') = m - cx^*$.

For the sequential selection of the $\beta/2$ diverse solutions to be included in the reference set, a maxmin criterion is used. The goal at each step is to find the solution that has the maximum distance between itself and the solutions currently in the reference set. The distance between a solution in P and the solutions in the reference set depends on the solution representation, as follows:

$$d(O, RefSet) = \min_{O' \in RefSet} p(O, O')$$

$$d(S, RefSet) = \min_{S' \in RefSet} c(S, S')$$

Then, at each step of the process, the solution R from P that has the maximum distance between itself and the solutions currently in $RefSet$ is selected to be added to the reference set. Mathematically, R is identified as follows:

$$R = \arg\max_{X \in P} d(X, RefSet)$$

where X is either O or S, depending on the solution representation. Solution R is added to $RefSet$ and deleted from P. This is repeated $\beta/2$ times in order to complete the $RefSet$ with β solutions. As shown in Algorithm 1, the iterative process begins after the initial $RefSet$ has been created. This process consists of selecting subsets of reference solutions to perform combinations that will result in new trial solutions. These solutions are then considered for inclusion in the reference set once they have been subjected to the improvement method. As recommended in [20], we employ a standard subset generation method that consists of selecting all pairs of reference solutions that contain at least one new solution. A new solution is one that has been added in the previous iteration of the do-loop in Algorithm 1. In the first iterations, all the solutions are new and therefore the number of subsets generated the first time around equals $\binom{\beta}{2}$.

The updating of $RefSet$ occurs at the end of each iteration. This is where the new solutions that were generated by the combination method and potentially improved by the improvement method are considered for membership in the reference set. The goal of the updating procedures is to maintain both quality and diversity in the reference set. Therefore, a solution is admitted to the reference set if its objective function value is better than the objective function value of the worst solution in the current reference set. For the sake of diversity, a solution that passes the above quality test replaces the reference solution that is closest to it according to the distance measures described above. This means that a solution that is included in the reference set does not necessarily replace the worst reference solution, unless the worst solution in the current $RefSet$ happens to be the closest to this new reference solution.

3. Diversification Generation Method

The diversification generator is one of three scatter search methods that are problem-specific. That is, this method along with the combination and improvement methods take advantage of the problem context to create a customized solution procedure within the scatter search framework. Like it is often done to induce diversification, our method relies on a controlled randomization process. Controlled randomization refers to a strategy that makes reference to the objective function value while constructing solutions. This is in contrast to full randomization where solutions are constructed without any support or direction from the objective function.

Algorithm 2 summarizes the steps performed by the diversification generation method for solutions represented as an ordering of the wavelengths. The set U consists of all the wavelengths that have not been assigned to a row. Initially, U contains all m wavelengths in the problem and the solution O being constructed is empty. The iterative process assigns one wavelength at a time and ends when no more unassigned wavelengths are left (i.e., when U is empty). In each step, a wavelength k is randomly selected from U by the $Random(\,)$ function. The $BestRow(\,)$ function then searches for the best row i^* where to assign k in the partial solution O. In the first iteration, the selected wavelength is assigned to the first row, that is $i^* = 1$. In the i^{th} iteration, there are $i-1$ wavelengths in the partial solution and the search for the best row i^* consists of inserting wavelength k in the $i-1$ available positions, starting in row 1. The row that produces the largest increase in the objective function value (i.e., that increases the number of

bandpasses the most) is chosen as i^*. If no row assignment from 1 to $i-1$ is able to improve upon the current objective function value then $i^* = i$, meaning that wavelength k is assigned to the last row.

$U \leftarrow \{1, \dots, m\}$
$O \leftarrow \emptyset$
while $U \neq \emptyset$ **do**
    $k \leftarrow Random(U)$
    $i^* \leftarrow BestRow(O, k)$
    $O(i^*) = k$
    $U \leftarrow U \backslash \{k\}$
end while

Algorithm 2 Diversification generation method for solutions represented by O.

The controlled randomization aspect of the method is evident in Algorithm 2 by a random selection of a wavelength that is followed by a purposeful search for the best row. The random nature of the procedure enables it to generate the population of solutions needed to initiate the scatter search.

Algorithm 3 shows the procedure for solutions represented by S that like Algorithm 2 is also semi-greedy [8] but that follows the tenets of GRASP [9]. For each candidate wavelength, the method calculates a greedy function $h_1(i, g)$ that consists of the number of the potential bandpasses that result from adding wavelength i to group g. A potential bandpass is a column within the group that contains all 1s after wavelength i has been added to the group. The function counts potential bandpasses because the final existence of a bandpass is not confirmed until B wavelengths are assigned to a group.

For instance, consider a problem with $n = 5$ and $B = 4$ and the two partial groups with two wavelengths each shown in Table 1. Wavelengths 4 and 8 are already assigned to group 1. Wavelengths 3 and 6 are assigned to group 2. The construction procedure is considering wavelength number 7 to be added to either group 1 or group 2. Each group in the example has two potential bandpasses, destinations 1 and 4 for group 1 and destinations 2 and 4 for group 2. Adding row 7 to group 1 preserves the potential of two bandpasses for that group (see blocks with bold lines in Table 1). Adding wavelength 7 to group 2 reduces the potential bandpasses to 1. Therefore, the greedy function evaluations for adding wavelength 7 to each group are:

$$h_1(7,1) = 2$$
$$h_1(7,2) = 1$$

Group	Wavelength	Destination				
		1	2	3	4	5
1	4	1	1	0	1	0
	8	1	0	1	1	0
	7	**1**	**0**	**0**	**1**	**1**
	?					
2	3	1	1	1	1	0
	6	0	1	0	1	0
	7	**1**	**0**	**0**	**1**	**1**
	?					

Table 1 Example of greedy group assignment.

To initiate the diversification generation procedure for solutions represented by S, the first wavelength is randomly chosen and is arbitrarily assigned to group 1 (see the first three lines in Algorithm 3). In the iterative loop, the greedy function is calculated for each pair of wavelength-group for all wavelengths in U and all groups with fewer than B wavelengths. The wavelength-group combination (i^*, g^*) is chosen among those pairs for which $h_1(i,g) \geq h_{min} + \alpha \cdot (h_{max} - h_{min})$, where h_{min} and h_{max} are the minimum and maximum values of h_1 among all the candidate wavelength-group combinations. As is customary in GRASP, α is a tunable parameter that varies between zero (random selection) and one (deterministic selection).

$U \leftarrow \{1, \dots, m\}$
$i \leftarrow Random(U)$
$S(i) \leftarrow 1$
$U \leftarrow U \backslash i$
while $U \neq \emptyset$ **do**
    $(i^*, g^*) \leftarrow SemiGreedy(U, S, \alpha)$
    $S(i^*) \leftarrow g^*$
    $U \leftarrow U \backslash \{i^*\}$
end while

Algorithm 3 Diversification generation method for solutions represented by S

A slight variant of this method is also tested in which the greedy function, $h_2(i,g)$, includes the number of bandpasses "eliminated" by adding a wavelength i to group g. That is, $h_2(i,g)$ is the difference between potential bandpasses and eliminated bandpasses. A wavelength eliminates a bandpass in a group if it adds a zero to a column that previously contained all 1s within the group. Under this variant, the h_2 values for the example in Table 1 are $h_2(7,1) = 2$ and $h_2(7,2) = 0$ because adding wavelength 7 to group 2 eliminates the potential bandpass at destination 2.

4. Improvement Method

In scatter search, solutions that are either constructed by the diversification generator or created by the combination method are subjected to an improvement process that is typically based on neighborhood searches. The search in an improvement method is local, meaning that it stops as soon as no improved solution is found in the neighborhood of the current solution. Four improvement methods were developed and tested for solutions represented by an ordering O of the wavelengths:

IM1 Best insertion
IM2 Best swap
IM3 Block merging
IM4 Variable neighborhood descent

All these improvement methods treat the bandpass problem as a search for the best ordering of a set of elements. The search neighborhood in IM1 is built via insertions of each wavelength in each row. Let O be the solution before the insertion of wavelength $O(i)$ in row k, and O' the solution after the insertion:

$$O = (O(1), \dots, O(i-1), O(i), O(i+1), \dots, O(k-1), O(k), O(k+1), \dots, O(m))$$

$$O' = (O(1), \dots, O(i-1), O(i+1), \dots, O(k-1), O(k), O(i), O(k+1), \dots, O(m))$$

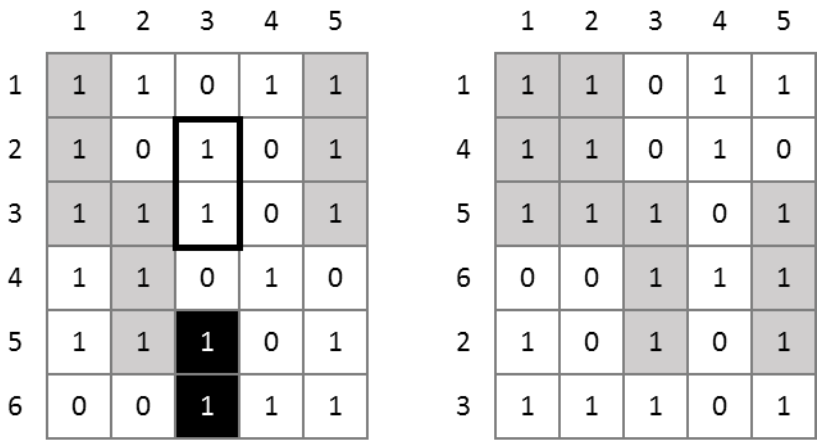
All insertions are attempted and the best is chosen. The procedure stops when no insertion results in an increase of the number of bandpasses in the current solution. The same best-move strategy is used by IM2 with the difference that swaps define the neighborhood instead of insertions. A swap between the wavelengths in rows i and k transforms O into O' as follows:

$$O = (O(1), \dots, O(i-1), O(i), O(i+1), \dots, O(k-1), O(k), O(k+1), \dots, O(m))$$

$$O' = (O(1), \dots, O(i-1), O(k), O(i+1), \dots, O(k-1), O(i), O(k+1), \dots, O(m))$$

IM3 is significantly different from IM1 and IM2. This improvement method analyzes the binary matrix associated with the current solution to identify higher-order exchanges. In particular, for each column in the matrix of the current solution it searches for two blocks of non-zero elements, one primary block with $B-2$ wavelengths and one secondary block with 2 wavelengths, where a block has non-zero elements in consecutive rows. Once these two blocks have been identified, the secondary block is merged "at the end" of the primary block by rearranging the row assignments of the wavelengths in the secondary block. This merger forms a new bandpass, however, the row reassignment of the two wavelengths in the secondary block could have eliminated one or more bandpasses in other columns. In an attempt to reestablish those bandpasses or finding new ones, an exhaustive search is performed among the rows corresponding to the new bandpass. This means that all permutations of the rows are explored in order to identify the best. The process is repeated for primary blocks with $B-3$ wavelengths, then $B-4$ wavelengths, and so on until the number of wavelengths in the primary block reaches 2. This local search procedure terminates when no improvement is possible after exploring all destinations.

Figure 3 shows a simple example of one move in IM3. As in previous examples, the gray blocks indicate the bandpasses. Figure 3.a shows the current solution with an objective function value of 3. In this figure, the primary block identified by IM3 is the one with the black background, which corresponds to wavelengths (rows) 5 and 6 in destination (column) 3. The secondary block is shown as a solid bold outline, corresponding to wavelengths (rows) 2 and 3 in destination (column) 3. As prescribed by the IM3 logic, the secondary block is inserted at the end of the primary block. This means that rows in the secondary block are removed and inserted right below the rows in the primary block, and then all rows (except the first one) are shifted up. The result of this exchange is an increase in the number of bandpasses, as shown in Figure 3.b. It can be shown that no permutation of the last four rows (i.e., the ones that participated in the exchange) results in a better solution.



a) $O = (1,2,3,4,5,6)$ b) $O = (1,4,5,6,2,3)$

Figure 3 Example of a move in IM3.

IM4 combines IM2 and IM3 in the framework of a variable neighborhood descent (VND). VND employs multiple neighborhoods that are typically ordered in increasing level of complexity [24]. The first neighborhood is applied until no improvement is possible. At that time, the second neighborhood is explored until an improved solution is found at which point the search reverts back to the first neighborhood and the process starts again. If no improved solution is identified then the search moves to the next neighborhood. The entire VND ends when the last (and often the most complex) neighborhood is not able to identify an improved solution. In our case, we only use two neighborhoods: IM2 and IM3. The VND starts with IM2 and it identifies the first local optimum. Then, IM3 is applied and as soon as an improved solution is found it goes back to IM2. The search keeps alternating between IM2 and IM3 until IM3 fails to find an improved solution. The selection of IM2 followed by IM3 was supported by a full-factorial experimental designed where all orderings of two neighborhoods and also all orderings of the three neighborhoods were considered. The experiment yielded IM2 followed by IM3 as the best combination in terms of solution quality and exploration time.

A single improvement method was developed for solutions represented by S. This method, like IM2, is based on swaps, which preserve the feasibility of a solution. The neighborhood search is organized in such

a way that the groups are ordered by increasing number of bandpasses, and hence the first group is the one with the least number of bandpasses and the last group is the one with the largest number of bandpasses. The exploration is such that it starts by attempting swaps of wavelengths from the first group with wavelengths in group 2 to G. Then, wavelengths in group 2 are exchanged with wavelengths in groups 3 to G, and so on until wavelengths from group $G-1$ are exchanged with wavelengths in group G. If a trial swap results in an increase in the number of bandpasses, the swap is executed. In other words, a first-improving strategy is used in this neighborhood exploration. The process terminates when no more improving swaps can be identified.

5. Combination Method

The combination method is a procedure that generates new solutions from a given subset of solutions. As mentioned above, our implementation considers only subsets of size 2 (i.e., pairs of reference solutions). Path relinking (PR) is used as the method to create new solutions from the subsets created by the subset generation method. PR was originally proposed in [12] and formalized in [13]. The main idea behind PR consists of creating a path (i.e., a sequence of solutions) between two or more solutions. It is called relinking because the process was conceived in the context of a solution method that performs neighborhood searches to move from one solution to the next. In such a search, solutions are connected by the paths that the procedure created to reach them. The relinking in that context refers to creating a new path to connect solutions that are typically selected due to their elite status. In scatter search, solutions in the reference set may not have historical paths that connect them. Hence, in most cases, the application of PR to two reference solutions builds the first path between them. The PR principles have been successfully applied to multiple problem classes and in a variety of methodological frameworks, including tabu search [15], scatter search [29], and GRASP [3].

Our first combination method based on PR (referred to as CM1) operates on solutions represented by an ordering O. Given a pair of reference solutions, one is denominated the initiating solution O^i and the other the guiding solution O^g. PR consists of transforming the initiating solution into the guiding solution by a sequence of moves. Consider for example the initiating solution $O^i = (5,2,3,4,6,1)$ and the guiding solution $O^g = (5,3,2,1,4,6)$ shown in Figure 4. The gray cells represent wavelengths in positions that do not coincide with the positions that they occupy in the guiding solution. Swaps are used to make the transformation from the initiating solution to the guiding solution, which is achieved in three moves. The most-improving swap is chosen at each step as prescribed by a greedy path relinking process [28]. The moves are represented by arrows that are labeled with the pair of wavelengths that are swapping positions.

Initially, there are four swaps that transform the initiating solution into intermediate solutions that move toward to the guiding solutions. The (6,1) swap results in an intermediate solution with an improved objective function value. For this particular example, the relinking process could stop at this point because the objective function value of the intermediate solution matches the upper bound value of 5. However, for completeness, the figure shows that, at each step, the relinking process continues from the best intermediate solution. In this case, the solution resulting from the (4,1) swap is chosen and then the (2,3) swap completes the process.

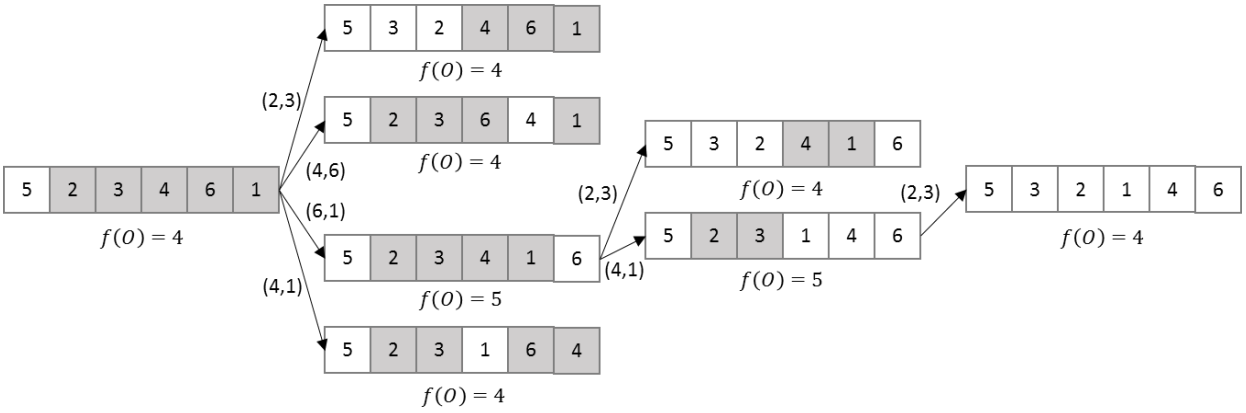


Figure 4 Path relinking example.

We developed a second combination method (referred to as CM2) based on a PR strategy that has been recently suggested and that has been labeled *Exterior Path Relinking* (EPR) [6][14]. EPR is a diversification strategy because, instead of moving the initiating solution toward the guiding solution, it searches for intermediate solutions that move away from the guiding solution. Figure 5 shows an EPR example applied to a $O^i = (5,2,3,4,6,1)$ and the guiding solution $O^g = (5,2,6,4,3,1)$. The process starts by randomly selecting one of the elements in O^i that occupies the same position in O^g, these are the cells with a gray background (i.e., 5, 2, 4, and 1 in the initiating solution). Assume that the random selection is 5. Then another random selection is made among those elements that, if swapped with 5, do not produce an increase in the number of wavelengths that the resulting intermediate solution will have in the same position as in the guiding solution.

In Figure 5, two possible paths are shows, where the left path is created by first swapping 5 and 6, while the right path is initiated by the swap of 5 and 3. Clearly, this process relies heavily on randomization to create a diverse set of solutions that move away from the guiding solution. A large number of paths could be created from any O^i and O^g pair. As done in [14], we generate additional solutions by reversing the roles of the initiating and guiding solutions.

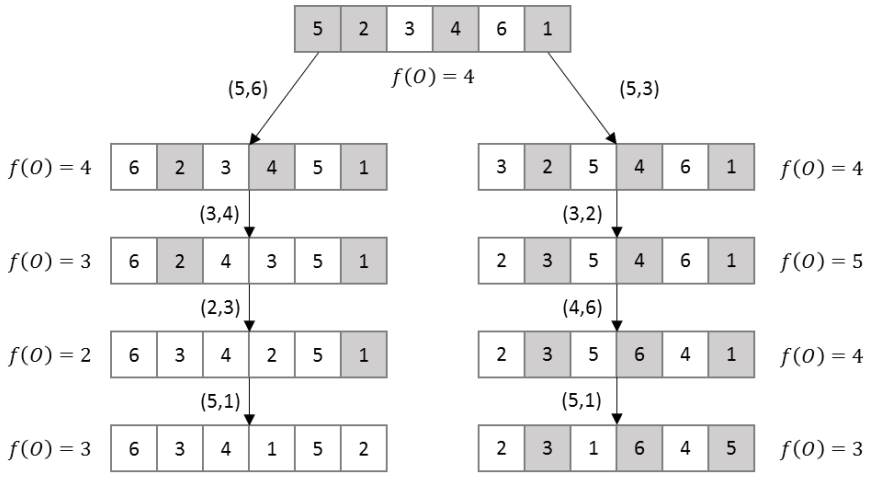


Figure 5 Exterior path relinking example.

The path relinking for solutions represented by a group assignment S starts with the solution of the matching problem described in Section 2. Recall that, given two solutions, the problem matches the groups in one solution with the groups in the other solution in order to maximize the total number of wavelengths assigned to the same group (also referred to as the common elements). The optimal matching identifies the common elements between an initiating solution S^i and a guiding solution S^g. Then, swaps are performed to move S^i toward S^g. As done in CM1, the swap with the largest improvement of the objective function is chosen. The process terminates when the guiding solution is reached.

Table 2 shows the group assignments for two reference solutions for a BP2 instance with $m = 18$ and $B = 6$, and therefore $G = 3$.

Solution	Group 1	Group 2	Group 3
1	1, 3, 6, 9, 12, 17	2, 5, 8, 10, 11, 18	4, 7, 13, 14, 15, 16
2	1, 4, 5, 10, 11, 18	2, 3, 6, 12, 13, 15	7, 8, 9, 14, 16, 17

Table 2 Two reference solutions for a BP2 instances with $m = 18$ and $B = 6$.

The matching problem associated with the solutions in Table 2 is:

$$\text{Maximize} \quad x_{11} + 3x_{12} + 3x_{13} + 4x_{21} + x_{22} + x_{23} + x_{31} + 2x_{32} + 3x_{33}$$

$$\text{Subject to} \quad x_{11} + x_{12} + x_{13} = 1$$
$$x_{21} + x_{22} + x_{23} = 1$$
$$x_{31} + x_{32} + x_{33} = 1$$

An optimal solution to this problem is $x_{12}^* = x_{21}^* = x_{33}^* = 1$, resulting in the following common elements:

Group 1: 3, 6, 12
Group 2: 5, 10, 11, 18
Group 3: 7, 14, 16

The PR process then starts by switching the labels of the first two groups in solution 2 (i.e., the original group 1 becomes group 2 and the original group 2 becomes group 1). Since $x_{33}^* = 1$, the original label for group 3 remains unchanged. Then, a sequence of swaps must be performed to move the wavelengths 1, 2, 4, 8, 9, 13, 15, and 17 from their current group in solution 1 to their group in solution 2. For instance wavelength 1 must be moved to group 2 and wavelength 2 must be moved to group 1. Therefore, a swap between wavelength 1 and 2 is considered since it moves solution 1 closer to solution 2.

6. Computational Experiments

Before engaging in competitive testing, we performed a series of scientific tests to determine the contribution of the various elements that we have designed for the two problem classes defined by the solution representations. These experiments are also used to determine the best SS configuration for each problem class. Henceforth, SS1 refers to the implementation for which solutions are presented by an ordering O of the wavelengths and therefore equipped to tackle BP1 and MBP instances. The SS2 procedure refers to the scatter search for which solutions are represented as an assignment S of

wavelengths to groups and that it is configured to solve BP2 instances. Table 3 summarizes the SS components for each SS implementation.

Component	SS1 (BP1 and MBP)	SS2 (BP2)
Solution representation	Ordering of wavelengths (O)	Assignment of wavelengths to groups (S)
Distance measure	Positional distance	Number of wavelengths assigned to different groups
Subset generation method	All pairs of reference solutions	All pairs of reference solutions
Initial reference set	Half of the reference solutions chosen by quality and half by diversity	Half of the reference solutions chosen by quality and half by diversity
Reference set update	Eligible new trial solution replaces the reference solution closest to it.	Eligible new trial solution replaces the reference solution closest to it.
Diversification generation	Semi-greedy (random selection of a wavelength followed by best insertion)	GRASP construction with two greedy functional forms (h_1 and h_2)
Improvement method	Four variants (IM1 to IM4) based on exchanges of positions	First-improvement swap of group assignments
Combination method	PR (CM1) and EPR (CM2) based on swapping positions	PR based on swapping group assignments

Table 3 Summary of SS components for the two bandpass problem classes.

Our scientific testing focuses on determining the best combination of components for each SS. All components were programmed in Java 8 and the experiments were executed on an Intel Core i7 2600 (3.4 GHz) processor with 4 GB of RAM. We use the Library of Bandpass Problems (BPLIB)[3] for BP1 and BP2 and the MBP library (MBPLIB)[4] for the MBP. The BPLIB contains 90 instances, 45 referred to as OS and 45 referred to as BKS. The OS instances were constructed in such a way that the optimal solutions are known for BP1 but not for BP2. The BKS problems were randomly generated and optimal solutions are not known for either BP1 or BP2. There are 45 instances with known optimal solutions in the MBP library. We use the BKS instances for the scientific experimentation and reserve the OS and the MBP instances for the competitive testing. This means that the experiments used to configure SS1 include only BP1 instances and that the configuration is expected to generalize to the MBP instances.

A two-factor factorial design is employed to configure SS1 by fixing the population size $|P|$ to 100 and the reference set size β to 10, the default values suggested in the literature. Also, the stopping criterion (see last line of Algorithm 1) is set as the iteration when no trial solution is admitted to the reference set. Since there are four improvement methods and two combination methods, the factorial design consists of 8 treatment combinations with 45 replications each. For this experiment, the deviation from the best-known solution is used as the response variable. The deviation is calculated as $\left(\frac{f^*-f}{f^*}\right)$, where f^* is the

[3] http://sci.ege.edu.tr/~math/BandpassProblemsLibrary/
[4] http://fen.ege.edu.tr/arifgursoy/mbpopt/

objective function value of the best-known solution and f is the objective function value of the solution obtained by the treatment combination. The ANOVA associated with the experiment revealed that both main effects are significant, with p-values smaller than 10^{-9}. However, the experiment did not detect a significant interaction effect between CM and IM, as indicated by a p-value of 0.881. Figure 6, a plot of the average deviation values (y-axis) for each improvement method (x-axis), confirms this finding, where it can be observed that the CM1 line (solid) does not cross the CM2 line (dotted).

Figure 6 shows that CM2 is more effective than CM1 regardless of the IM. It also shows that the response obtained by the IM alternatives does not change with the choice of CM. This means that the selection of the CM and the IM could have been done independently. The experiment indicates that the best treatment combination and therefore the SS1 configuration that is expected to yield the best results is CM2 with IM4.

In the process of conducting this experiment, 39 solutions were found that improved upon the best-known solution currently published in the BPLIB website. (We point out that the website does not indicate how those solutions were found given that neither [1] nor [2] reports results on these instances.) The objective function values of the new best-known solutions, which improve upon the current best-known solutions by an average of 16.2%, are shown in Appendix A.

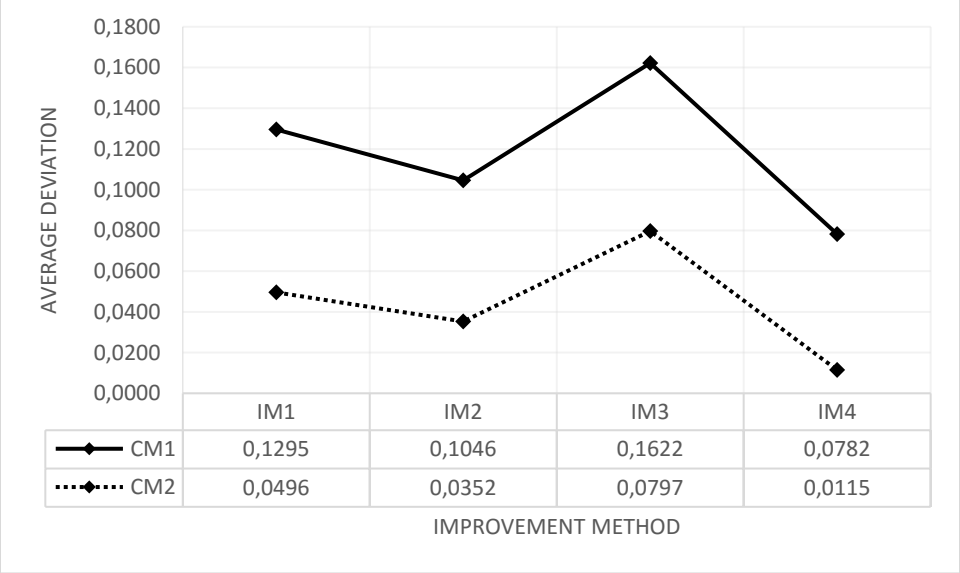
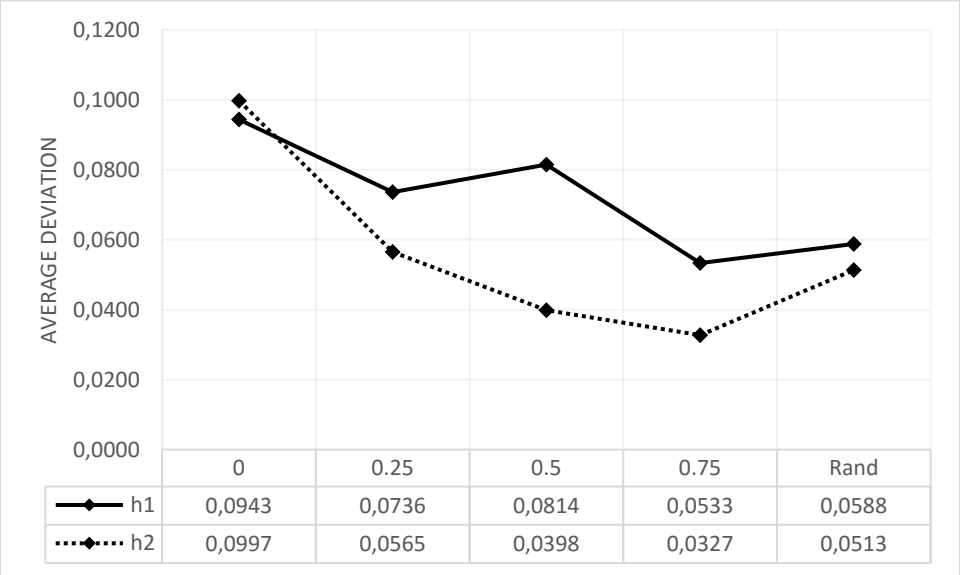


Figure 6 Plot of average response for all (CM, IM) treatment combinations.

To configure SS2 we also fix $|P|$ and β to their default values and execute the procedure (Algorithm 1) a total of 10 times on each instance. Recall that the standard SS stopping criterion (which we also use for SS1) terminates the search when the reference set does not change, that is, when in a given iteration, no trial solution is admitted to the reference set. In SS2, the procedure is allowed to rebuild the reference set by seeding it with the incumbent solution. That is, the incumbent solution is placed in the new reference set first and the remaining $b - 1$ solutions are added following the construction processes within SS2. A two-factor factorial design is employed to study the effects of various α values and the

greedy functions h_1 and h_2. The α values included in the experiment are 0, 0.25, 0.5, and 0.75. We also include a variant where the value of α is selected at random during the search. This factorial design has a total of 10 treatment combinations and 45 replications. We again use the average deviation —calculated as described above— as the response variable. The ANOVA yielded p-values of 0.03 and 0.00015 for the main effects associated with the greedy functions and the α values, respectively. Hence, we consider both main effect significant. The interaction effect between the two factors has a p-value of 0.366. Therefore, we conclude that there is no significant interaction between the choice of h and α. Figure 7 is a graphical representation of the average deviations obtained by each treatment combination.



	0	0.25	0.5	0.75	Rand
h1	0,0943	0,0736	0,0814	0,0533	0,0588
h2	0,0997	0,0565	0,0398	0,0327	0,0513

Figure 7 Plot of average response for all (h, α) treatment combinations.

It can be concluded from Figure 7 that h_2 is generally more effective than h_1, except when $\alpha = 0$. Also in general, increasing the value of α —thus making the construction less random— improves performance (except for h_1 and $\alpha = 0.5$). Allowing α to change randomly during the search seems to be better than using small α values. Given these results, the combination h_2 and $\alpha = 0.75$ is selected.

We now turn our attention to the competitive testing for which we use the OS and the MBP instances. The competitors for BP1 are the DSR (Different Start Rows) method of Babayev, Bell, and Nuriyev [1] and the four variants of a genetic algorithm (GA1 to GA4) proposed by Berbeler and Gürsoy [2]. Results of the OS instances associated with DSR and the GA variants were obtained from [1] and [2] respectively. The articles do not report computational time. For future reference, we report that the SS1 results shown in Table 4 were obtained in an average of 875 seconds. In Table 4, OF Value refers to the average objective function value and #Opt indicates the number of optimal solutions that each competitor is able to find out of the 45 in the set. The column labeled GAP contains the average relative optimality gap (i.e., the absolute gap divided by the optimal objective function value and averaged over all instances).

Procedure	OF Value	GAP	#Opt
DSR	37.00	0.3407	0
GA1	40.53	0.2516	3
GA2	40.53	0.2550	3
GA3	40.24	0.2613	2
GA4	40.29	0.2575	3
SS1	46.07	0.1027	16

Table 4 Summary of competitive testing for BP1.

The results in Table 4 indicate that there is little difference in performance among the GA variants. DSR produces results that are inferior to those produced by all other methods while SS1 dominate all competitors. In order to support this conclusion, we performed a Friedman test. This test assigns, for each instance in the test set, the value of 1 to the procedure that obtains the best result, the value 2 to the second best, and so forth. The average ranking is then calculated for each procedure and a p-value is calculated to assess the strength of the ranking. The rankings shown in Table 5 result in a p-value of less than 0.001.

Procedure	Ranking
SS1	1.12
GA2	3.33
GA1	3.34
GA4	3.61
GA3	3.73
DSR	5.86

Table 5 Friedman test ranking.

The next competitive test compares the performance of SS1 with four variants of a genetic algorithm (GA1 to GA4) proposed by Gürsoy and Nuriyev [17] for MBP. The 45 MBPLIB instances are employed for this comparison and the summary of the results is presented in Table 6. Gürsoy and Nuriyev [17] do not report solution times but for future reference we note that the SS1 solutions summarized in Table 6 were obtained in an average of 267 seconds. Table 6 does not include the #Opt column because none of the procedures is capable of finding a single optimal solution. This is in agreement with the large optimality gap shown in Table 6. Nonetheless, the average objective function value of the SS1 solutions is at least 47.6% better than the average objective function values of all the GA variants, resulting in optimality gap that is less than half of those corresponding to the GAs. Although this is a significant improvement over the existing GA procedures, it is clear that the strategies designed for BP1 and embedded in SS1 lose some of their effectiveness when applied to MBP instances. Additional strategies that directly exploit the structure of the MBP instances are necessary to produce improved outcomes.

Procedure	OF Value	GAP
GA1	20.38	0.5153
GA2	19.49	0.5326
GA3	19.69	0.5324
GA4	18.76	0.5458
SS1	30.09	0.2349

Table 6 Summary of competitive testing for MBP instances.

Our third and last competitive testing compares the performance of SS2 to LocalSolver, a commercial metaheuristic optimizer, and Gurobi, a commercial MIP solver. To the best of our knowledge, there is no specialized procedure for BP2 in the literature. The Gurobi solutions were found with the IP formulation presented in [1], and the solver was run with its default optimization settings. This means that the search took advantage of the 8 processors on the machine that was used for testing. The LocalSolver model that we used is included in Appendix B. The model is a simplified version of the IP formulation of BP2 presented in [1]. The LocalSolver model is simpler because it requires of only one set of binary variables, namely, the `x[i][g]` variables that indicate whether or not wavelength i is assigned to group g. We perform two experiments, one with a relatively short search time limit of 600 seconds per instance and one with a longer search horizon of 3000 seconds. As described above, SS2 is set up to rebuild the reference set every time the stopping criterion is satisfied and therefore is capable of continuing the search until any specified time limit. The results of this experiments are summarized in Table 7.

Procedure	OF Value	GAP	#Opt	#Best	#Outright	#Ties	#Worst
Time limit of 600 seconds							
Gurobi	31.044	0.0434	19	25	1	24	18
LocalSolver	32.577	0.0644	11	31	7	24	12
SS2	32.711	0.0119	17	33	1	32	2
Time limit of 3000 seconds							
Gurobi	31.733	0.0291	19	27	0	27	17
LocalSolver	32.778	0.0516	12	37	8	29	8
SS2	32.956	0.0050	19	37	1	36	2

Table 7 Summary of results for the competitive testing on BP2 instances.

The GAP column in Table 7 is calculated against the best-known solution for each instance, 19 of which are optimal, as confirmed by the 3000-second Gurobi runs. The #Opt is the number of optimal solutions, out of the 19 that are known, that each procedure is able to match. The #Best column contains the number of best-known solutions matched by each procedure. The #Best values are divided into two groups, the number of outright best solutions (#Outright) and the number of ties with at least one of the competing procedures (#Ties). The #Worst column indicates the number of times that the solution procedure yielded the worst solution

The following observations can be made from the analysis of the results in Table 7:

- SS2 exhibits the most robust behavior of the three competing procedures. Its #Best values are at the top in both the short and the long runs. At the same time, its #Worst values are the lowest in both runs.
- The GAP values of 0.0119 and 0.005 for SS2 supports the robustness argument. That is, SS2 produces high-quality solutions in most runs and avoids arbitrarily inferior solutions. SS2's worst deviation from the best-known solutions is 0.067 in both, short and long runs.
- LocalSolver is highly competitive and a viable option for finding high-quality solutions to BP2 instances. This solution alternative is particularly attractive when one takes into consideration the development effort of creating a customized solution methods such as SS2. The only caveat is the variability of the solution quality produced by Local Solver. Note that for the long runs LocalSolver has the same number of outright best solutions (8) as the number of worst solutions. This variability causes LocalSolver to have the worst average deviation.
- Gurobi's GAP values are reasonably low because, even though it has the highest #Worst counts in both runs, its deviations from the best-known solutions are never more than 0.2

Additional search strategies could widen the gap between the performance of a specialized procedure like SS2 for the BP2 and a commercial software LocalSolver. For instance, strategic oscillation proved critical in boosting performance in a tabu search for a grouping problem [11].

7. Conclusions

The bandpass problem is somewhat new to the OR literature in the sense that, although it was originally introduced at a conference more than 10 years ago, the first publication did not appear until 2009. Several versions of the problem have been introduced in subsequent publications along with additional test data. One of our goals in this project was to gain understanding of the current state of knowledge associated with the bandpass problem and to identify avenues for advancing this area. Through our investigation, we identified two problem classes and a total of three problem variants within those classes. We then designed a common SS solution framework and developed the individual strategies for each problem class.

We performed meticulous scientific testing that provided some interesting insights on the behavior and interaction of the search components. These experiments produced new benchmarks that will help future researchers test solution methods that could prove even more effective than those described here. We were also able to show that although BP1 and MBP belong to the same problem class, the effectiveness of the search strategies designed for BP1 diminish when applied to MBP instances. Nonetheless, competitive testing showed that the proposed SS1 method applied (without customization) to both BP1 and MBP instances produces results that are significantly better than those produced by the existing procedures. Finally, our experiments show that the adaptation of the SS to BP2 exhibits a robust behavior when compared to commercial software.

Acknowledgments

This research was partially supported by the Ministerio de Economía y Competitividad of Spain (Project Number TIN2012-35632-C02) and the Comunidad de Madrid (Project Number S2013/ICE-2894).

References

[1] V. Babayev, G. I. Bell, and U. G. Nuriyev, "The bandpass problem: combinatorial optimization and library of problems," *Journal of Combinatorial Optimization*, 18(2):151–172, 2009.

[2] M. E. Berberler and A. Gürsoy, "Genetic algorithm approach for bandpass problem," in *24th Mini EURO Conference*, 2010.

[3] V. Campos, R. Martí, J. Sánchez-Oro, and A. Duarte, "GRASP with path relinking for the orienteering problem," *Journal of the Operational Research Society*, In press, 2013.

[4] A. Das and M. Roberts, "Metric distances of permutations," www.cra.org/Activities/craw_archive/dmp/awards/2004/Das/paper.ps, 2004.

[5] A. Duarte, R. Martí, F. Glover, and F. Gortázar, "Hybrid scatter tabu search for unconstrained global optimization," *Annals of Operations Research*, 183(1):95–123, 2009.

[6] A. Duarte, J. Sánchez-Oro, M. G. C. Resende, F. Glover, and R. Martí, "GRASP with exterior path relinking for differential dispersion minimization," *Information Sciences*, In press, 2014.

[7] A. Duarte and R. Martí, "Tabu search and grasp for the maximum diversity problem," *European Journal of Operational Research*, 178(1):71–84, 2007.

[8] T.A. Feo and M.G.C. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," *Operations Research Letters*, 8:67–71, 1989.

[9] T.A. Feo and M.G.C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, 6:109–133, 1995.

[10] M. Gallego, A. Duarte, M. Laguna, and R. Martí, "Hybrid heuristics for the maximum diversity problem," *Computational Optimization and Applications*, 44(3):411–426, 2009.

[11] M. Gallego, M. Laguna, R. Martí, and A. Duarte, "Tabu search with strategic oscillation for the maximally diverse grouping problem," *Journal of the Operational Research Society,* 64(5):724-734.

[12] F. Glover, "Heuristics for integer programming using surrogate constraints," *Decision Sciences*, 8(1):156–166, 1977.

[13] F. Glover, "A template for scatter search and path relinking," *Lecture Notes in Computer Science*, 1363, 1998.

[14] F. Glover, "Exterior path relinking for zero-one optimization," *International Journal of Applied Metaheuristic Computing*, In press, 2014.

[15] F. Glover and M. Laguna, *Tabu search,* New York: Springer, 1997.

[16] W. J. Goralski, *SONET, a guide to synchronous optical networks*. New York: McGraw-Hill, 1997.

[17] A. Gürsoy and U. Nuriyev, “Genetic algorithm for multi bandpass problem and library of problems,” in Proceedings of the IV International Conference on Problems of Cybernetics and Informatics (PCI), Baku, Azerbaijan, September 12-14, 2012.

[18] I. P. Kaminow, A. A. M. Saleh, R. E. Thomas, V. W. S. Chan, and M. L. Stevens, “A wideband all-optical WDM network,” *IEEE Journal on selected areas in communications*, 14(5):780–799, 1996.

[19] M. Kurt, H. Kutucu, A. Gürsoy, and U. Nuriyev, “The optimization of the bandpass lengths in the multi-bandpass problem,” in Proceedings of the Seventh International Conference on Management Science and Engineering Management: Focused on Electrical and Information Technology, J. Xu, J. A. Fry, B. Lev, and A. Hajiyev, Editors, (1): 115-123, Berlin Heidelberg: Springer-Verlag, 2014.

[20] M. Laguna and R. Martí, *Scatter search: Methodology and implementations in C*. Boston, USA: Kluwer Academic Publisher, 2003.

[21] Z. Li and G. Lin, “The three column bandpass problem is solvable in linear time,” *Theoretical Computer Science,* 412(4-5):281-299, 2011.

[22] G. Lin, “On the bandpass problem,” *Journal of Combinatorial Optimization,* 22(1):71-77,2011.

[23] R. Martí, A. Duarte, and M. Laguna, “Advanced scatter search for the max-cut problem,” *INFORMS Journal on Computing*, 21(1):26–38, 2009.

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

[25] U. G. Nuriyev, H. Kutucu, and M. Kurt, “Mathematical models of the bandpass problem and OrderMatic computer game,” *Mathematical and Computer Modeling,* 53: 1282-1288, 2011.

[26] J. J. Pantrigo, R. Martí, A. Duarte, and E. G. Pardo, “Scatter search for the cutwidth minimization problem,” *Annals of Operations Research*, 199(1):285–304, 2011.

[27] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective*. San Francisco, USA: Morgan Kaufmann, 1998.

[28] M. Resende, R. Martí, M. Gallego, and A. Duarte, “GRASP and path relinking for the max-min diversity problem,” *Computers and Operations Research*, 37:498-508, 2010.

[29] J. Sánchez-Oro, M. Laguna, A. Duarte, and R. Martí, “Scatter search for the profile minimization problem,” *Networks*, In press, 2014.

Appendix A

Table 10 shows the objective function values of the new best-known solutions for the BKS instances. A dash indicates that no solution better than the current best known was found in our experimentation.

Instance	Previous	New	%Improve	Instance	Previous	New	%Improve
P01-A1B8	10	11	10.00	P24-A10B16	10	12	20.00
P02-A1B16	3	-	-	P25-A11B5	54	59	9.26
P03-A2B8	14	16	14.29	P26-A11B8	23	29	26.09
P04-A2B16	5	6	20.00	P27-A11B16	6	9	50.00
P05-A3B8	20	23	15.00	P28-A12B5	78	83	6.41
P06-A3B16	8	-	-	P29-A12B8	37	40	8.11
P07-A4B5	27	30	11.11	P30-A12B16	9	13	44.44
P08-A4B8	12	15	25.00	P31-A13B5	71	-	-
P09-A4B16	4	-	-	P32-A14B5	83	87	4.82
P10-A5B5	40	41	2.50	P33-A15B8	53	56	5.66
P11-A5B8	18	21	16.67	P34-A16B16	29	35	20.69
P12-A5B16	5	7	40.00	P35-A17B5	104	106	1.92
P13-A6B5	62	67	8.06	P36-A18B5	215	223	3.72
P14-A6B8	31	34	9.68	P37-A19B16	71	-	-
P15-A6B16	9	11	22.22	P38-A20B5	132	134	1.52
P16-A7B8	17	20	17.65	P39-A22B8	74	78	5.41
P17-A8B8	25	27	8.00	P40-A22B25	17	22	29.41
P18-A8B16	9	10	11.11	P41-A23B5	161	-	-
P19-A9B5	44	47	6.82	P42-A24B8	93	95	2.15
P20-A9B8	18	23	27.78	P43-A25B16	51	53	3.92
P21-A9B16	5	8	60.00	P44-A26B5	245	248	1.22
P22-A10B5	62	68	9.68	P45-A27B5	28	41	46.43
P23-A10B8	33	35	6.06				

Table 10 New best solutions found during the scientific experimentation for SS1.

Appendix B

LocalSolver input and model functions for the experiments with BP2.

```
function input()
{
  usage = "\nUsage: localsolver BP2.lsp "
        + "B=BandpassNumber fileName=dataFile "
        + "[lsTimeLimit=timeLimit] [lsVerbosity=0 or 1]\n";

  if (fileName == nil) error(usage);
  dataFile = openRead(fileName + ".txt");
  m = readInt(dataFile);
  n = readInt(dataFile);
  a[i in 1..m][j in 1..n] = readInt(dataFile);
  close(dataFile);
  G = floor(m/B);
  b = m-G*B;
  GP = (b > 0) ? G+1 : G;
}

function model()
{
  // x[i][g] equal to 1 if wavelength i is assigned to group g
  x[1..m][1..GP] <- bool();

  // each group from 1 to G has exactly B rows
  for [g in 1..G] constraint sum[i in 1..m](x[i][g]) == B;

  // group G+1 if it exists has b rows
  if (GP > G) constraint sum[i in 1..m](x[i][GP]) == b;

  // each row is included in exactly one group
  for [i in 1..m] constraint sum[g in 1..GP](x[i][g]) == 1;

  // y[j][g] is equal to 1 if there is a bandpass in destination j of group g
  y[j in 1..n][g in 1..G] <- sum[i in 1..m] (a[i][j]*x[i][g]) == B;
  if (GP > G) y[j in 1..n][GP] <- sum[i in 1..m] (a[i][j]*x[i][GP]) == b;

  // objective function is the sum of bandpasses
  obj <- sum[g in 1..GP][j in 1..n] (y[j][g]);
  maximize obj;
}
```