Path Relinking for the Fixed Spectrum Frequency Assignment Problem

Xiangjing Lai, Jin-Kao Hao*

LERIA, Université d'Angers, 2 bd Lavoisier, 49045 Angers, Cedex 01, France

Abstract

The fixed spectrum frequency assignment problem (FS-FAP) is a highly relevant application in modern wireless systems. This paper presents the first path relinking (PR) approach for solving FS-FAP. We devise four relinking operators to generate intermediate solutions (or paths) and a tabu search procedure for local optimization. We also adopt a diversity-and-quality technique to maintain population diversity. To show the effectiveness of the proposed approach, we present computational results on the set of 42 benchmark instances commonly used in the literature and compare them with the current best results obtained by any other existing methods. By showing improved best results (new upper bounds) for 19 instances, we demonstrate the effectiveness of the proposed PR approach. We investigate the impact of the relinking operators and the population updating strategy. The ideas of the proposed could be applicable to other frequency assignment problems and search problems.
Keywords: Path relinking; Population algorithm; Heuristics; Frequency assignment.

1. Introduction

Frequency assignment represents a key issue in wireless communications systems and has attracted a lot of attentions (Audhya et al., 2013; Castelino et al., 1996; Fischettti et al., 2000; Flood et al., 2013; Hale, 1980; Hao et al., 1998; Hurley et al., 1997; Kim et al., 2007; Král, 2005; Lai and Coghill, 1996; Maniezzo and Carbonaro, 2000; San Jose-Revuelta, 2007; Smith et al., 1998; Subramanian et al., 2008; Tiourine et al., 2000; Wang and Rushforth, 1996; Wu et al., 2013). The fixed spectrum frequency assignment problem (FS-FAP) studied in this study is among the most typical frequency assignment problems (Aardal et al., 2007; Eisenblätter and Koster, 2010; Montemanni and Smith, 2010).

Informally, given a radio network, FS-FAP aims to assign a set of available frequencies to each transmitter of the network such that the level of interference

*Corresponding author.

Email addresses: laixiangjing@gmail.com (Xiangjing Lai), hao@info.univ-angers.fr (Jin-Kao Hao)

of the frequency assignment is minimized. Technically, interference occurs when two close transmitters are assigned frequencies which are close in the frequency spectrum.

From a graph-theoretic point of view, FS-FAP can be defined as follows (Aardal et al., 2007; Montemanni and Smith, 2010). Given a double weight undirected graph $G = (V, E, D, P)$ with a vertex set V, an edge set E, and two edge weight sets D and P, as well as a set F of consecutive frequencies, FS-FAP is to find a mapping (i.e., a frequency assignment) f from V to F, (i.e., $f : V \rightarrow F$) such that an objective or cost function is minimized. In the present model, a vertex $v \in V$ corresponds to a transmitter of the network, an edge $e \in E$ represents a pair of transmitters for which the assigned frequencies are constrained, D is the set of edge weights (d_{ij}, the first type) defining the separation constraints between frequencies of any two adjacent vertices, and P is the set of edge weights (p_{ij}, the second type) representing the penalty values measuring the degree of interference defined as follows. Given a frequency assignment f and an edge $e(i, j) \in E$, the penalty p_{ij} is incurred if $|f(i)-f(j)| \leq d_{ij}$, i.e., if the separation constraint is violated. Then the cost of the frequency assignment f is defined as the total interference generated by f which is given by the summation of the incurred penalty values (p_{ij}). The goal of FS-FAP is then to determine a frequency assignment of minimum cost. Note that in the literature, there exist other similar FS-FAP models which differ mainly in the way the cost function is defined (Aardal et al., 2007; Hale, 1980; Hao et al., 1998; Hurley et al., 1997; Kendall and Mohamad, 2004; Mabed et al., 2011; Park et al., 2002).

Frequency assignment problems like FS-FAP are closely related to the bandwidth coloring problem (also called the restricted T-coloring problem) with k fixed colors (Dorne and Hao, 1998; Jin and Hao, 2014; Johnson et al., 2008; Lai and Lü, 2013; Malaguti and Toth, 2008; Roberts, 1991) and known to be computationally hard in general. Indeed, the FS-FAP problem can be shown to be NP-hard since it is reduced to the NP-hard graph k-coloring problem (Hale, 1980). Given the high complexity of FS-FAP, considerable efforts have been made to develop effective heuristic methods. These include single solution based approaches like tabu search (Castelino et al., 1996; Hao et al., 1998; Montemanni et al., 2003), adaptive local search (Kendall and Mohamad, 2004), heuristic manipulation technique (Montemanni and Smith, 2010). Population-based approaches are also very popular: ANTS algorithm (Maniezzo and Carbonaro, 2000), genetic algorithms (Dorne and Hao, 1996; Park et al., 2002; San Jose-Revuelta, 2007), hybrid evolutionary algorithms (Dorne and Hao, 1995; Jin et al., 2001; Kim et al., 2007; Lai and Coghill, 1996), multiple strategy method (Hurley et al., 1997). In addition to heuristic methods, approaches using integer programming were also proposed in the literature to determine lower bounds for the FS-FAP problem, such as those in (Montemanni et al., 2001, 2004).

Recently, the population-based path relinking approach (Glover, 1997; Glover et al., 2000) has attracted special attention in combinatorial optimization, and shows outstanding performances in solving a number of difficult problems, such as antibandwidth problem (Duarte et al., 2011), bandwidth coloring problem

(Lai et al., 2014), clustering (Martins de Oliveira et al., 2014), flow shop sequencing and scheduling (Costa et al., 2012; Reeves and Yamada, 1998), unconstrained binary quadratic optimization (Wang et al., 2012), and web services composition (Parejo et al., 2014). In this paper, we introduce an effective path relinking algorithm for solving FS-FAP with the following main contributions.

From the perspective of algorithm design, we base our approach on the general path relinking framework and devise dedicated relinking operators. These operators are used to create solution paths from one initiating solution to a guiding solution. For the purpose of local optimization, we develop a specific tabu search procedure. To maintain a healthy diversity of the solution population, we apply a quality-and-diversity technique to update the solution pool. The integration of these ingredients leads to an effective algorithm.

Indeed, the proposed algorithm is assessed on the set of 42 benchmark instances commonly used in the literature and shows a very competitive performance in comparison with the best performing method dedicated to the studied problem in the literature. In particular, the proposed algorithm is able to discover improved best known results (new upper bounds) for 19 instances and matches the best known results for 21 other cases.

The rest of this paper is organized as follows. In Section 2, we present the proposed algorithm. In Section 3, we show computational results by comparing them with the results of the best performing algorithm in the literature. In Section 4, we investigate some key ingredients of the proposed algorithm. In Section 5, we draw conclusions and discuss several research perspectives.

2. Path Relinking Algorithm for FS-FAP

The path relinking algorithm presented in this paper is a hybrid population algorithm that combines four essential ingredients: population initialization, local optimization method, relinking operator, and population updating strategy. In this section, we present the general path relinking algorithm and its ingredients.

2.1. Search Space and Evaluation Function

We first define the search space Ω explored by the algorithm and the evaluation function which is used to measure the quality of a candidate solution.

A solution of FS-FAP is a mapping from V to F ($f : V \rightarrow F$), thus a solution s can be represented by a one-dimensional vector where each variable represents a vertex and its value is the assigned frequency. The search space Ω explored by the algorithm is composed of all possible assignment, and hence the size of the search space is bounded by $O(|F|^{|V|})$.

To evaluate the quality of a frequency assignment $s = (f_1, f_2, \ldots, f_n) \in \Omega$, we calculate the degree of interference as follows:

$$Cost(s) = \sum_{e(i,j) \in E; |f_i - f_j| \leq d_{ij}} p_{ij} \tag{1}$$

where $d_{ij} \in D$ represents the distance constraint between the frequencies of two adjacent vertices i and j, and p_{ij} represents the resulting penalty when the distance constraint is violated, i.e., $|f_i - f_j| > d_{ij}$. This cost function (to be minimized) is also referred as the objective function in the rest of the paper.

2.2. Main Framework

Algorithm 1 Pseudo-code of our path relinking algorithm for FS-FAP

```
1: Input: Problem instance I, the size of population p
2: Output: the best solution s* found
3: repeat
4:    P = {s^1, ..., s^p} ← Population_Initialization(I, p)        /* Section 2.3 */
5:    if it is not in the first loop then
6:       s^w = arg max{f(s^i) : i = 1, ..., p}
7:       P ← P ∪ {s*} \ {s^w}
8:    end if
9:    s* = arg min{f(s^i) : i = 1, ..., p}
10:   PairSet ← {(s^i, s^j) : 1 ≤ i < j ≤ p}
11:   while PairSet ≠ ∅ do
12:      Randomly pick a solution pair (s^i, s^j) ∈ PairSet
13:      PairSet ← PairSet \ {(s^i, s^j)}
14:      s' ← PathRelinking(s^i, s^j), s'' ← PathRelinking(s^j, s^i)/* Section 2.5 */
15:      s' ← TabuSearch(s')                                       /* Section 2.4 */
16:      if f(s') < f(s*) then
17:         s* ← s', f(s*) ← f(s')
18:      end if
19:      P ← UpdatingPool(PairSet, P, s')
20:      PairSet ← UpdatingPool(PairSet, P, s')                    /* Section 2.6 */
21:      s'' ← TabuSearch(s'')                                     /* Section 2.4 */
22:      if f(s'') < f(s*) then
23:         s* ← s'', f(s*) ← f(s'')
24:      end if
25:      P ← UpdatingPool(PairSet, P, s'')
26:      PairSet ← UpdatingPool(PairSet, P, s'')                   /* Section 2.6 */
27:   end while
28: until a stop criterion is met
```

The general procedure of our path relnking algorithm is shown in Algorithm 1, where the following notations are used. s^* and s^w respectively represent the best solution found so far and the worst solution in the population, and $PairSet$ is the set of solution pairs (s^i, s^j) and is initially composed of all possible solution pairs (s^i, s^j) in the population.

The proposed algorithm starts with an initial population P (line 4) which includes p different solutions, where each of them is randomly generated (Section 2.3) and then is optimized by the tabu search procedure (Section 2.4). Subsequently, the algorithm enters a while loop (lines 11 to 27) and at each iteration

a solution pair is randomly selected from $PairSet$ and then is used to generate new solutions by using a relinking operator (line 14) (Section 2.5) and the tabu search procedure (lines 15 and 21). Specifically, for the selected solution pair (s^i, s^j), two paths composed of intermediate solutions are built: one from s^i to s^j and one from s^j to s^i. Then two particular solutions $s^{'}$ and $s^{''}$ are identified from the two paths and are separately improved by tabu search. After that, the population updating strategy is used to update the population P with the improved offspring solution (lines 19 and 25). Simultaneously, the $PairSet$ is also updated accordingly as follows: First, the solution pair (s^i, s^j) is removed from $PairSet$ (line 13). Then, if the offspring solution $s^{'}$ replaces a solution s^r in the population, all solution pairs containing s^r are removed from $PairSet$ and all solution pairs that can be generated by combining $s^{'}$ with other solutions in the population are added into $PairSet$ (see Section 2.6). The while loop ends when $PairSet$ becomes empty, then the population is recreated, while retaining the best solution S^* found so far in the newly created population (lines 4 to 7).

Our algorithm terminates when the population has been rebuilt q times (a predetermined value) or a given timeout limit is reached.

In the following subsections, the ingredients of the proposed algorithm are respectively described.

2.3. Population Initialization

Within the proposed algorithm, a population is constructed as follows at the beginning of the search or when the reference set $PairSet$ becomes empty. First, we generate $3p$ random solutions from the search space Ω, where each variable (or vertex) of each solution is assigned a frequency randomly taken from the frequency domain F. Then, for each generated solution, the tabu search procedure (see Section 2.4) is applied to optimize the solution to a local optimum. Finally, we select the first p best solutions (according to the cost function given in (1)) to form the initial population.

2.4. Local Optimization with Tabu Search

In the present study, we employ a tabu search (TS) algorithm as the local optimization routine of our path relinking algorithm.

2.4.1. Neighborhood Structure

The TS procedure adopts the conflict-based neighborhood (Dorne and Hao, 1998; Hao et al., 1998; Montemanni et al., 2003). Given a solution s, a neighboring solution of s with respect to this neighborhood is obtained by changing the value (frequency) of one conflicting variable (vertex) u from its original frequency f_i to another frequency f_j (denoted by $< u, f_i, f_j >$), and the resulting neighborhood $N(s)$ is composed of all such neighboring solutions. Here, a vertex is considered to be conflicting if at least one of the distance constraints associated with this vertex is violated. Thus, the size of the neighborhood is bounded by $O(|CV| \times |F|)$, where $|CV|$ represents the number of conflicting vertices in s, and $|F|$ is the number of frequencies available.

2.4.2. Tabu List Management

In order to forbid the recently visited solutions to be revisited, a tabu list is maintained. During the search, each time a move $< u, f_i, f_j >$ is performed, (i.e., the frequency of u is changed from f_i to f_j), vertex u is forbidden to receive the frequency f_i for the next tt iterations unless the aspiration condition (see Section 2.4.3) is satisfied, where tt is called tabu tenure and is dynamically determined as follows:

$$tt(u) = \lceil R \times \frac{Freq[u][c_i]}{max_{0 \leq v < n, 0 \leq q < |F|}\{Freq[v][q]\}} \rceil + rand(10..12) \quad (2)$$

In this equation, R is a predetermined parameter, $Freq[v][q]$ represents the number of times of assigning frequency q to vertex v during the current TS round, and $rand(10..12)$ represents a random integer from 10 to 12. Thus, the tabu tenure has a minimum length of 10 which is dynamically adjusted by some historical information.

2.4.3. Aspiration Criterion and Stopping Condition

Since the attributes (vertices and frequencies used) of the solution instead of the solution itself are forbidden, some good neighboring solutions may be prevented from being visited by TS. To avoid this, we use the following aspiration criterion to revoke the tabu status of a forbidden move: A forbidden move is allowed as long as it leads to a solution better than the best solution found so far or the set of non-tabu neighborhood solutions becomes empty.

The stopping condition of our TS algorithm is the maximum number α of iterations during which the best solution has not been improved, and this number is called the depth of TS algorithm.

2.5. The Relinking Operator

The relinking operator is one of the most important components of the path relinking approach, and its goal is to generate, from two high-quality parent solutions, new promising solutions by creating solution paths connecting the two parent solutions. Our relinking operator includes two main operations. The first one is to construct a solution path that connects two parent solutions. The parent solutions located at the beginning and the end of the path are respectively called the initiating and guiding solutions, while the others are called intermediate (or path) solutions. The second operation is to choose from each constructed path one solution as the reference solution which will be further improved by the tabu search procedure.

To build a path connecting the initiating and guiding solutions, there are generally several strategies that can be used, leading to different relinking operators whose effectiveness may vary according to the specific problem structures. In this study, in order to identify the most appropriate relinking operators for FS-FAP, we devise four relinking operators: the greedy relinking operator (denoted by PR), the random relinking operator (denoted by rPR), the mixed

relinking operator (denoted by mPR), and the randomized and mixed relinking operator (denoted by mrPR). In the following, we describe these relinking operators.

The pseudo-code of the four different relinking operators are respectively described in Algorithms 2 to 5.

Algorithm 2 Pseudo-code of constructing a path for the greedy PR operator

Input: A pair of solutions (s^1, s^2)
Output: Path solutions $s(0), s(1), \ldots, s(r)$ from s^1 to s^2
$NC \leftarrow \{l : s^1_l \neq s^2_l, l = 1, 2, \ldots, n\}$
$s(0) \leftarrow s^1$, $s \leftarrow s^1$, $m \leftarrow 1$
while $|NC| > 0$ **do**
    $l^* \leftarrow argmin\{f(s \oplus l) : l \in NC\}$
    $NC \leftarrow NC \setminus \{l^*\}$
    $s \leftarrow s \oplus l^*$
    $s(m) \leftarrow s$, $m \leftarrow m + 1$
end while

Algorithm 3 Pseudo-code of constructing a path for the random PR operator

Input: A pair of solutions (s^1, s^2)
Output: Path solutions $s(0), s(1), \ldots, s(r)$ from s^1 to s^2
$NC \leftarrow \{l : s^1_l \neq s^2_l, l = 1, 2, \ldots, n\}$
$s(0) \leftarrow s^1$, $s \leftarrow s^1$, $m \leftarrow 1$
while $|NC| > 0$ **do**
    $l^* \leftarrow$ a randomly selected $l \in NC$
    $NC \leftarrow NC \setminus \{l^*\}$
    $s \leftarrow s \oplus l^*$
    $s(m) \leftarrow s$, $m \leftarrow m + 1$
end while

For the PR and rPR relinking operators (Algorithms 2 and 3), a solution sequence (i.e., a path) with a length of $r + 1$: $(s(0),\ s(1),\ s(2)\ \ldots\ s(r))$ is generated in a step by step way by starting from $s(0)$, where $s(m)$ differs from $s(m-1)$ $(m = 1, 2 \ldots r)$ by the value of only one variable, and $r + 1$ represents the size of the initial NC, where NC is a set of variables whose values are different with respect to the two solutions considered. In addition, $s(0)$ and $s(r)$ correspond respectively to the initiating solution s^1 and the guiding solution s^2, while the other solutions are intermediate (or path) solutions. Moreover $s \oplus l$ means that the lth variable of s is changed according to the guiding solution.

The only difference between the PR and rPR relinking operators lies in the way to select the variable l from NC for generating the next solution on the path. In rPR , based on the current solution $s(i)$, a random variable l $(l \in NC)$ is chosen to generate the next solution $s(i + 1)$, $i = 0, 1, \ldots, r - 1$, whereas in the PR relinking operator a variable l $(l \in NC)$ yielding the best objective value is always chosen.

Algorithm 4 Pseudo-code of constructing a path for the mixed PR operator

Input: A pair of solutions (s^1, s^2)
Output: Path solutions $s(0), s(1), \ldots, s(r)$ from s^1 to s^2
$NC \leftarrow \{l : s_l^1 \neq s_l^2, l = 1, 2, \ldots, n\}$
$s(0) \leftarrow s^1$, $s \leftarrow s^1$, $m \leftarrow 1$
while $|NC| > 0$ **do**
  $l^* \leftarrow argmin\{f(s \oplus l) : l \in NC\}$
  $NC \leftarrow NC \setminus \{l^*\}$
  $s \leftarrow s \oplus l^*$
  $s(m) \leftarrow s$
  if m is odd **then**
    $s^1 \leftarrow s$, $s \leftarrow s^2$
  else
    $s^2 \leftarrow s$, $s \leftarrow s^1$
  end if
  $m \leftarrow m + 1$
end while

Algorithm 5 Pseudo-code of constructing a path for the randomized mixed PR operator

Input: A pair of solutions (s^1, s^2)
Output: Path solutions $s(0), s(1), \ldots, s(r)$ from s^1 to s^2
$NC \leftarrow \{l : s_l^1 \neq s_l^2, l = 1, 2, \ldots, n\}$
$s(0) \leftarrow s^1$, $s \leftarrow s^1$, $m \leftarrow 1$
while $|NC| > 0$ **do**
  $CVL \leftarrow$ the first λ best $l \in NC$ /* construct a candidate list for variables */
  $l^* \leftarrow$ a randomly selected $l \in CVL$
  $NC \leftarrow NC \setminus \{l^*\}$
  $s \leftarrow s \oplus l^*$
  $s(m) \leftarrow s$
  if m is odd **then**
    $s^1 \leftarrow s$, $s \leftarrow s^2$
  else
    $s^2 \leftarrow s$, $s \leftarrow s^1$
  end if
  $m \leftarrow m + 1$
end while

After the creation of a path, we choose one solution from this path as the reference solution such that the chosen solution is far enough from both the initiating and guiding solutions and has a good objective function value (Wang et al., 2012). Specifically, we first construct a candidate solution list (CSL) which is composed of the path solutions having a distance of at least $\xi \cdot HD$ (where ξ is a predefined parameter between 0 and 1.0, and HD is the Hamming distance between the initiating and guiding solutions) from both the initiating and guiding solutions, then the solution having the best objective value in CSL is chosen as the reference solution which will be further improved by the tabu search procedure.

As for the mixed relinking operator (mPR) described in Algorithm 4, the path solutions are alternately generated from s^1 to s^2 and from s^2 to s^1 in a greedy way, where s^1 and s^2 are alternately chosen as the guiding solution. Specially, let s be the current path solution, s^2 the current guiding solution, the mPR operator will pick the variable l^* ($l^* \in NC$) yielding the best objective value to generate the next path solution (i.e., $s \leftarrow s \oplus l^*$), and NC is updated accordingly (i.e., $NC \leftarrow NC \setminus \{l^*\}$). Then, to generate the next path solution, the current path solution s is updated and s^1 is chosen as the the guiding solution. The above process is repeated until the NC becomes empty. Finally, the last generated path solution is chosen as the reference solution.

The mrPR relinking operator described in Algorithm 5 is a randomized version of the mPR operator, and the difference between them lies in the way to select the variable from NC for generating the next path solution. In Algorithm 4, the variable yielding the best objective value is always selected at each step. However, in Algorithm 5, a candidate variable list (CVL) composed of the first λ best variables ($\lambda = Min\{\lambda_{max}, |NC|\}$) with respect to the objective value is first built from NC, then a variable l in CVL is chosen at random.

2.6. Population Updating

Within a population-based algorithm, when a new solution is generated, a population updating strategy is used to determine whether the new solution should be inserted into the population and which solution in the population should be replaced. In the present algorithm, we use a distance-and-quality based updating strategy to control the diversity of population and maintain simultaneously high-quality solutions in the population (Lü and Hao, 2010; Porumbel et al., 2010; Sörensen and Sevaux, 2006).

The population updating strategy used in our algorithm is described in Algorithm 6 and can be stated as follows: let s^o denote the offspring solution, s^r denote the closest solution to s^o, and s^w be the worst solution in the population, then the population is updated in the following two cases. First, if the distance between the solutions s^o and s^r is less than or equals to the cutoff distance $\beta \times n$ (β is a predetermined parameter and n is the number of vertices in the graph) and s^o is better than s^r in terms of the objective value, then s^o is inserted into the population, replacing the solution s^r. Second, if the distance between the solutions s^o and s^r is larger than or equals to the cutoff distance ($\beta \times n$) and s^o is better than s^w in terms of the objective value, then s^o is inserted into

the population, replacing the solution s^w. Finally, the reference set $PairSet$ is updated accordingly, as explained in Section 2.2.

Algorithm 6 Pseudo-code of Population Updating Strategy

Input: Population P, reference set $PairSet$, offspring solution s^o, distance cutoff factor β
Output: Population P , reference set $PairSet$
$s^w = arg\ max\{f(s^i) : i = 1, \ldots, p\}$
$s^r \leftarrow$ the closest solution to the offspring s^o
$Dis \leftarrow$ the Hamming distance between s^o and s^r
if $f(s^o) < f(s^r)$ and $Dis \leq \beta \times n$ **then**
    $P \leftarrow P \cup \{s^o\} \setminus \{s^r\}$
    $PairSet \leftarrow PairSet \cup \{(s^o, s^k) : s^k \in P\} \setminus \{(s^r, s^k) : s^k \in P\}$
else if $f(s^o) < f(s^w)$ and $Dis > \beta \times n$ **then**
    $P \leftarrow P \cup \{s^o\} \setminus \{s^w\}$
    $PairSet \leftarrow PairSet \cup \{(s^o, s^k) : s^k \in P\} \setminus \{(s^w, s^k) : s^k \in P\}$
end if

3. Experimental Results and Comparisons

This section is dedicated to experimental assessments of the proposed path relinking algorithm and comparisons with the state of the art methods in the literature.

3.1. Instances and Experimental Protocol

A set of 42 benchmarks is considered in our experiments, which is available online at `http://www.idsia.ch/~roberto/FAP08.zip`, and the characteristics of these instances are summarized in Table 1.

Our path relinking algorithm was programmed in C++ and compiled by g++ compiler without the optimization option [1], and all experiments are carried out on a computer with an Intel Xeon E5440 processor (2.83 GHz CPU and 2Gb RAM). Following the DIMACS machine benchmark, our machine requires 0.43s, 2.62s, and 9.85s respectively for graphs $r300.5$, $r400.5$, and $r500.5$ [2]. Table 2 gives the descriptions and settings of the parameters used in our algorithm. Note that these parameter values are determined according to a preliminary experiment.

In the present study, we conduct two experiments to evaluate systematically two versions of our path relinking algorithm, i.e., one with the rPR relinking operator, and another one with the mrPR relinking operator. In the first experiment, we use the maximum number q ($q = 5$) of rebuilding population as the

[1] The source code of our algorithm and the best solutions reported in this work will be available at `http://www.info.univ-angers.fr/pub/hao/prfap.html`

[2] fdmax: `ftp://dimacs.rutgers.edu/pub/dsj/clique/`. Note that the DIMACS machine benchmark procedure is compiled by gcc compiler without optimization option.

stopping condition of algorithm (see Section 3.2). Additionally, in the second experiment we use the time limit as the stopping condition to make a comparison with the reference algorithm (see Section 3.3). Given the stochastic nature of our path relinking algorithm, for each experiment each instance is independently solved 10 times by each tested version of the algorithm, without special parameter tuning.

Table 1: Characteristics of graph used in the instances

Graph	$\|V\|$	$\|E\|$	Average d_{vw}	Average p_{vw}
AC-45-17	45	482	0.29	1.00
AC-45-25	45	801	0.34	1.00
AC-95-9	95	781	0.00	1.00
AC-95-17	95	2298	0.15	1.00
GSM-93	93	1073	0.28	1.00
GSM-246	246	7611	0.32	1.00
Test95	95	1214	1.37	1.00
Test282	282	10430	1.38	1.00
P06-5	88	3021	0.58	1.00
P06-3	153	9193	0.59	1.00
P06b-5	88	3021	0.39	1.00
P06b-3	153	9193	0.4	1.00
GSM2-184	184	6809	0.2	8.95×10^6
GSM2-227	227	10088	0.18	9.10×10^6
GSM2-272	272	14525	0.16	7.95×10^6
1-1-50-75-30-2-50	75	835	0.26	10.81
1-2-50-75-30-4-50	75	835	0.62	11.01
1-3-50-75-30-0-50	75	835	0.00	10.97
1-4-50-75-30-2-1	75	835	0.25	1.00
1-5-50-75-30-2-100	75	835	0.26	21.35
1-6-50-75-30-0-1000	75	835	0.00	2068.48

Table 2: Settings of important parameters

Parameters	Section	Description	Values
p	2.2	population size	30
α	2.4	depth of TS	10^3
R	2.4	parameter used in tabu tenure	40
ξ	2.5	distance parameter used in relinking operator	0.33
λ_{max}	2.5	size of candidate variable list	3
q	2.5	number of times of rebuilding population	5
β	2.6	distance parameter used in the pool updating	0.35

3.2. Computational Results of the path relinking algorithms

The first experiment aims to evaluate the performance of two versions of the proposed algorithm with the rPR and mrPR relinking operators. The computational statistics on the 42 benchmark instances are summarized in Table 3. The first two columns of the table give the characteristics of instance, including the name of graph (name) and the span of frequencies ($|F|$), the third and fourth columns respectively give the lower bounds obtained by the integer

Table 3: Computational statistics of the rPR and mrPR algorithms on the 42 benchmark instances, including the best objective function value over 10 independent runs, average objective function value, average running time, and deviation of the best objective function from the best known result. The improved results are indicated in bold.

Graph	$\|F\|$	LB	C_{pre}	rPR				mrPR			
				C_{best}	C_{avg}	$\%Dev$	T_{avg}	C_{best}	C_{avg}	$\%Dev$	T_{avg}
AC-45-17	7	20	32	32	32.0	0.00	55	32	32.0	0.00	49
AC-45-17	9	10	15	15	15.0	0.00	63	15	15.0	0.00	55
AC-45-25	11	26	33	33	33.0	0.00	94	33	33.0	0.00	107
AC-95-9	6	27	31	31	31.0	0.00	125	31	31.0	0.00	117
AC-95-17	15	28	33	33	33.4	0.00	290	33	33.9	0.00	265
AC-95-17	21	9	10	10	10.0	0.00	207	10	10.0	0.00	247
GSM-93	9	17	32	32	32.2	0.00	170	32	32.1	0.00	185
GSM-93	13	4	7	7	7.3	0.00	127	7	7.5	0.00	155
GSM-246	21	50	79	80	80.8	1.27	882	79	79.7	0.00	1061
GSM-246	31	16	25	25	26.5	0.00	805	25	25.4	0.00	1017
Test95	36	7	8	8	8.0	0.00	138	8	8.0	0.00	180
Test282	61	21	51	54	56.1	5.88	56	54	55.5	5.88	4525
Test282	71	6	27	29	30.1	7.41	4258	28	29.8	3.70	4187
Test282	81	-	10	**9**	10.7	-10.0	3213	**9**	10.9	-10.0	3503
P06-5	11	121	133	133	133.0	0.00	204	133	133.0	0.00	152
P06-3	31	109	115	115	116.9	0.00	531	115	115.6	0.00	641
P06b-5	21	49	52	52	52	0.00	142	52	52.0	0.00	157
P06b-5	31	25	25	25	25	0.00	180	25	25	0.00	201
P06b-3	31	106	112	112	112.1	0.00	540	112	112	0.00	681
P06b-3	71	26	26	26	26	0.00	931	26	26	0.00	716
GSM2-184	39	4856	5447	**5250**	**5258.1**	-3.62	1529	**5258**	**5266.1**	-3.47	1578
GSM2-184	49	874	874	874	874.0	0.00	672	874	874.0	0.00	898
GSM2-184	52	-	162	162	162.0	0.00	295	162	162.0	0.00	425
GSM2-227	29	-	61586	**57731**	**59405.8**	-6.26	3131	**56955**	**59573.5**	-7.52	3091
GSM2-227	39	7445	10550	**8772**	**9121.6**	-16.85	2611	**8809**	**9201.1**	-16.50	2251
GSM2-227	49	1998	2459	**1998**	**2004.7**	-18.75	2230	**1998**	**2015.5**	-18.75	2396
GSM2-272	34	-	56128	**53080**	**54570.2**	-5.43	5513	**53688**	**54795.3**	-4.35	5344
GSM2-272	39	16144	27416	**26237**	27475.0	-4.3	6278	**26453**	28135.1	-3.51	5368
GSM2-272	49	6310	7785	**6997**	**7128.5**	-10.12	5798	**7056**	**7208.0**	-9.36	4480
1-1-50-75-30-2-50	5	806	1242	1242	1242.0	0.00	89	1242	1242.0	0.00	97
1-1-50-75-30-2-50	10	53	97	**96**	**96.4**	-1.03	241	**96**	**96.2**	-1.03	242
1-1-50-75-30-2-50	11	36	59	**55**	**57.1**	-6.78	223	**57**	**57.7**	-3.39	249
1-1-50-75-30-2-50	12	-	36	**32**	**32.1**	-11.11	184	**32**	**32.3**	-11.11	187
1-2-50-75-30-4-50	9	-	671	**665**	**665.0**	-0.89	234	**665**	**665.0**	-0.89	246
1-2-50-75-30-4-50	11	-	317	**313**	**313.8**	-1.26	282	**313**	**313.6**	-1.26	292
1-3-50-75-30-0-50	7	-	194	194	194.4	0.00	173	194	196.0	0.00	172
1-4-50-75-30-2-1	6	-	70	70	70.0	0.00	69	70	70	0.00	76
1-4-50-75-30-2-1	10	17	19	19	19.0	0.00	79	19	19.0	0.00	97
1-5-50-75-30-2-100	10	94	176	**168**	**168.9**	-4.55	295	**168**	**168.9**	-4.55	278
1-5-50-75-30-2-100	12	-	63	**57**	**57.1**	-9.52	229	**57**	**57.5**	-9.52	273
1-6-50-75-30-0-1000	10	6586	6840	**6777**	**6777.0**	-0.92	288	**6777**	**6785.7**	-0.92	301
1-6-50-75-30-0-1000	13	-	1207	**1190**	1213.7	-1.41	198	**1190**	**1203.1**	-1.41	235
#Better				17				17			
#Equal				22				23			
#Worse				3				2			

programming approaches (Montemanni et al., 2004) and the best known results in terms of objective or cost value (C_{pre}), the columns 5 to 8 report the computational results of the PR algorithm with the rPR relinking operator (denoted by rPR), including our best objective value (C_{best}) over 10 independent runs, the average objective value (C_{avg}), the percentage of deviation ($\%Dev$) of our best objective value relative to the best known results, which is computed as $((C_{best} - C_{pre})/C_{pre}) \times 100\%$, and the average computational time in seconds (T_{avg}) per run. Columns 9 to 12 report the results of the PR algorithm with the mrPR relinking operator (denoted by mrPR). The improved results are indicated in bold and the mark "-" means that the corresponding result is not available in the literature. Finally, the rows *Better, Equal, Worse* respectively show the number of instances for which the best result of our corresponding PR algorithm is better, equal, and worse compared to the best known result of the literature.

One observes from the table that the rPR algorithm is able to improve the best known results for 17 out of the 42 instances, and match the best known results for 22 instances. However, it fails to find the previous best known results for the 3 remaining instances, Test282(61),Test282(71),GSM-246(21). In terms of the computing speed, it can be seen that the average computing times are less than 1000 seconds except for the "GSM2" and "Test" instances for most of which the rPR algorithm is able to improve the best known results. Furthermore, the improvement of the objective value is large for some instances. For instance, the percentage of improvement reaches 18.75% for GSM2-227(49).

As for the mrPR algorithm, a similar performance can be observed when comparing with the rPR algorithm. Specifically, the mrPR algorithm is also able to improve the best known results for 17 instances and to match the best known results for 23 instances. Nevertheless, for the GSM2 instances, the mrPR algorithm performs slightly worse than the rPR algorithm in terms of both the best and average objective function values. It is worth noting that the mrPR relinking operator has a slightly higher computational complexity than the rPR relinking operator. However, the additional complexity of the mrPR relinking operator can be ignored since most of computing time of the proposed PR algorithms is consumed by the tabu search method.

Thus, the above results indicate that our PR algorithms (including rPR and mrPR) have a strong search capacity compared to the state-of-the-art results in the literature.

On the other hand, one also observes from the table that for the 4 instances, i.e., P06b-5(31), P06b-3(71), GSM2-184(49), GSM2-227(49), our best objective value is equal to the lower bound yielded by the integer programming approaches, meaning that the optimality of these results are proved. Moreover, it is worth noting that the optimality of the result of GSM2-227(49) is for the first time proved in this work.

3.3. Comparison of the PR algorithm with the best performing algorithm

In order to make a systematical comparison between the present PR algorithms and the best performing FS-FAP algorithm (i.e., the HMT algorithm

Table 4: Comparison of the rPR algorithm with the best performing algorithm (HMT) in terms of the best cost value, average cost value and worst cost value over 10 independent runs. The winner results between two algorithms are indicated in bold.

			C_{best}		C_{avg}		C_{worst}	
Graph	$\|F\|$	C_{pre}	HMT	rPR	HMT	rPR	HMT	rPR
AC-45-17	7	32	32	32	32.0	32.0	32	32
AC-45-17	9	15	15	15	15.0	15.0	15	15
AC-45-25	11	33	33	33	33.0	33.0	33	33
AC-95-9	6	31	31	31	31.0	31.0	31	31
AC-95-17	15	33	33	33	33.0	33.0	33	33
AC-95-17	21	10	10	10	10.0	10.0	10	10
GSM-93	9	32	32	32	32.2	32.0	33	32
GSM-93	13	7	7	7	7.0	7.0	7	7
GSM-246	21	79	79	79	**80.2**	80.6	**81**	82
GSM-246	31	25	**25**	26	26.1	26.1	27	27
Test95	36	8	8	8	8.0	8.0	8	8
Test282	61	51	**51**	56	**53.2**	56.8	**55**	58
Test282	71	27	**27**	29	**29.3**	30.5	**30**	33
Test282	81	10	10	**9**	11.9	**10.9**	**13**	14
P06-5	11	133	133	133	133.0	133.0	133	133
P06-3	31	115	115	115	115.0	115.0	115	115
P06b-5	21	52	52	52	52.0	52.0	52	52
P06b-5	31	25	25	25	25.0	25.0	25	25
P06b-3	31	112	112	112	112.0	112.0	112	112
P06b-3	71	26	26	26	26.0	26.0	26	26
GSM2-184	39	5447	5447	**5258**	5598.8	**5270.8**	5689	**5322**
GSM2-184	49	874	874	874	1043.6	**874.0**	1120	**874**
GSM2-184	52	162	162	162	260.6	**162.0**	287	**162**
GSM2-227	29	61586	61586	**57790**	66510	**59555.4**	70105	**61300**
GSM2-227	39	10550	10550	**8656**	10897.7	**9022.4**	1116	**9559**
GSM2-227	49	2459	2459	**1998**	2613.1	**1998.0**	2828	**1998**
GSM2-272	34	56128	56128	**53254**	58691.4	**55954.2**	64353	**57360**
GSM2-272	39	27416	**27416**	27503	28488.2	**28299.7**	29307	**28927**
GSM2-272	49	7785	7785	**7185**	7946.7	**7265.2**	8459	**7393**
1-1-50-75-30-2-50	5	1242	1242	1242	1253.9	**1242.0**	1260	**1242**
1-1-50-75-30-2-50	10	97	97	**96**	103.8	**96.0**	109	**96**
1-1-50-75-30-2-50	11	59	59	**55**	66.1	**55.0**	70	**55**
1-1-50-75-30-2-50	12	36	36	**32**	38.7	**32.**0	42	**32**
1-2-50-75-30-4-50	9	671	671	**665**	680.6	**665.0**	691	**665**
1-2-50-75-30-4-50	11	317	317	**313**	325.0	**313.0**	335	**313**
1-3-50-75-30-0-50	7	194	194	194	196.5	**194.0**	199	**194**
1-4-50-75-30-2-1	6	70	70	70	70.9	**70.0**	71	**70**
1-4-50-75-30-2-1	10	19	19	19	19.0	19.0	19	19
1-5-50-75-30-2-100	10	176	176	**168**	183.8	**168**	199	**168**
1-5-50-75-30-2-100	12	63	63	**57**	69.3	**57**	74	**57**
1-6-50-75-30-0-1000	10	6840	6840	**6777**	7064.3	**6777.0**	7267	**6777**
1-6-50-75-30-0-1000	13	1207	1207	**1190**	1365.2	**1190.0**	1440	**1190**
#Better				16		22		21
#Equal				22		17		17
#Worse				4		3		4

Table 5: Comparison of the mrPR algorithm with the best performing algorithm (HMT) in terms of the best cost value, average cost value and worst cost value over 10 independent runs. The winner results between two algorithms are indicated in bold.

			C_{best}		C_{avg}		C_{worst}	
Graph	$\|F\|$	C_{prev}	HMT	mrPR	HMT	mrPR	HMT	mrPR
AC-45-17	7	32	32	32	32.0	32.0	32	32
AC-45-17	9	15	15	15	15.0	15.0	15	15
AC-45-25	11	33	33	33	33.0	33.0	33	33
AC-95-9	6	31	31	31	31.0	31.0	31	31
AC-95-17	15	33	33	33	33.0	33.0	33	33
AC-95-17	21	10	10	10	10.0	10.0	10	10
GSM-93	9	32	32	32	32.2	**32.0**	33	**32**
GSM-93	13	7	7	7	7.0	7.0	7	7
GSM-246	21	79	79	**78**	80.2	**79.0**	81	**80**
GSM-246	31	25	25	**24**	26.1	**25.1**	27	**26**
Test95	36	8	8	8	8.0	8.0	8	8
Test282	61	51	**51**	56	**53.2**	57.1	**55**	58
Test282	71	27	**27**	29	**29.3**	30.6	**30**	32
Test282	81	10	10	10	11.9	**11.5**	13	13
P06-5	11	133	133	133	133.0	133.0	133	133
P06-3	31	115	115	115	115.0	115.0	115	115
P06b-5	21	52	52	52	52.0	52.0	52	52
P06b-5	31	25	25	25	25.0	25.0	25	25
P06b-3	31	112	112	112	112.0	112.0	112	112
P06b-3	71	26	26	26	26.0	26	26	26
GSM2-184	39	5447	5447	**5250**	5598.8	**5276.9**	5689	**5322**
GSM2-184	49	874	874	874	1043.6	**874**	1120	**874**
GSM2-184	52	162	162	162	260.6	**162**	287	**162**
GSM2-227	29	61586	61586	**58834**	66510	**59907.7**	70105	**61269**
GSM2-227	39	10550	10550	**8760**	10897.7	**9329.7**	11164	**9755**
GSM2-227	49	2459	2459	**1998**	2613.1	**2009.4**	2828	**2045**
GSM2-272	34	56128	56128	**54085**	58691.4	**56916.3**	64353	**58524**
GSM2-272	39	27416	**27416**	28074	**28488.2**	28880.4	**29307**	29491
GSM2-272	49	7785	7785	**7107**	7946.7	**7252.5**	8459	**7361**
1-1-50-75-30-2-50	5	1242	1242	1242	1253.9	**1242.0**	1260	**1242**
1-1-50-75-30-2-50	10	97	97	**96**	103.8	**96.0**	109	**96**
1-1-50-75-30-2-50	11	59	59	**55**	66.1	**55.0**	70	**55**
1-1-50-75-30-2-50	12	36	36	**32**	38.7	**32.0**	42	**32**
1-2-50-75-30-4-50	9	671	671	**665**	680.6	**665.0**	691	**665**
1-2-50-75-30-4-50	11	317	317	**313**	325.0	**313**	335	**313**
1-3-50-75-30-0-50	7	194	194	194	196.5	**194**	199	**194**
1-4-50-75-30-2-1	6	70	70	70	70.9	**70.0**	71	**70**
1-4-50-75-30-2-1	10	19	19	19	19.0	19.0	19	19
1-5-50-75-30-2-100	10	176	176	**168**	183.8	**168.0**	199	**168**
1-5-50-75-30-2-100	12	63	63	**57**	69.3	**57.0**	74	**57**
1-6-50-75-30-0-1000	10	6840	6840	**6777**	7064.3	**6777.0**	7267	**6777**
1-6-50-75-30-0-1000	13	1207	1207	**1190**	1365.2	**1190.0**	1440	**1190**
#Better				17		24		23
#Equal				22		15		16
#Worse				3		3		3

(Montemanni and Smith, 2010)), we carry out the second experiment on the 42 benchmark instances respectively using the rPR and mrPR algorithms with a cutoff time limit of 2400 seconds, which is the same as that used by the reference algorithm (HMT).

Like in Montemanni and Smith (2010), the rPR and mrPR algorithms are respectively run 10 times for each tested instance. The computational results are summarized in Tables 4 and 5 together with the results of the reference algorithm (HMT). The first three columns of Table 4 show the information as before. Columns 4 to 5 report the best results respectively for the HMT algorithm and the rPR algorithm. The average results are respectively reported in columns 6 to 7 for both algorithms, and the worst results are respectively given in columns 8 to 9. Additionally, the results of our mrPR algorithm are presented in Table 5, with the same notations as in Tables 3 and 4.

From Table 4, one can observe that our rPR algorithm improves the best known result for 16 out of the 42 instances, matches the best known results for 22 instances and misses the best known result for 4 instances. In terms of the average objective value, our rPR algorithm is able to find a better result for 22 instances compared with the reference algorithm, match the results obtained by the reference algorithm for the 17 easy instances, and obtain a worse result for the 3 remaining instances. In terms of the worst objective value, our rPR algorithm obtains a better result on 21 instances compared to HMT, and reaches an equal results for 17 instances.

On the other hand, it can be seen from Table 5 that the mrPR algorithm is able to improve the best known result for 17 out of the 42 instances, and obtain a worse result for only 3 instances. In addition, in terms of the average objective value, mrPR gets a better result than HMT for 24 instances, and obtains an equal result on 15 instances. As for the worst objective value, the mrPR algorithm obtains a better and an equal result respectively for 23 instances and 16 instances compared to the reference algorithm.

These results indicate that both our rPR and mrPR algorithms are very competitive compared with the best performing HMT algorithm for the studied problem.

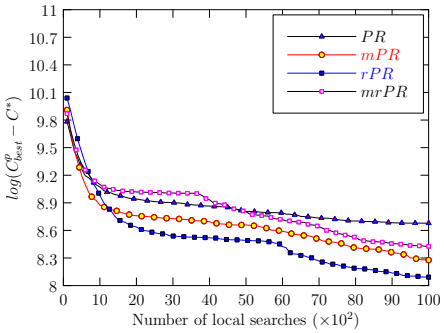
4. Analysis and Discussions

In this section, we analyze two essential components of the proposed algorithm, i.e., the relinking operator and the population updating strategy.

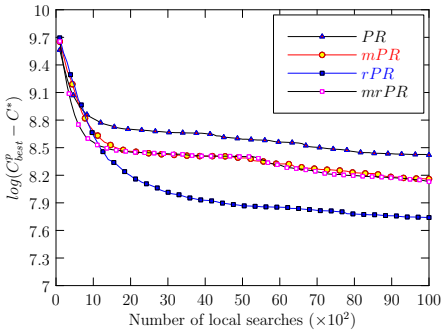
4.1. Performance Comparison among the Relinking Operators

First, in order to understand whether the relinking operator has a significant influence on the performance of the path relinking algorithm and which relinking operator is more appropriate for FS-FAP, we compare the four proposed relinking operators based on a set of 11 representative instances. For each tested relinking operator and each instance considered, the corresponding algorithm is independently performed 10 times with a maximum number θ of

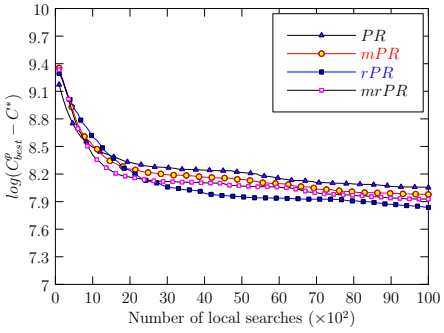
(a) GSM2-227-29



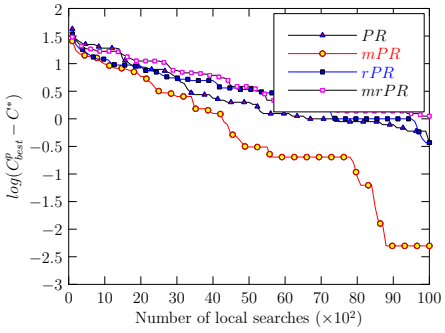
(b) GSM2-272-34



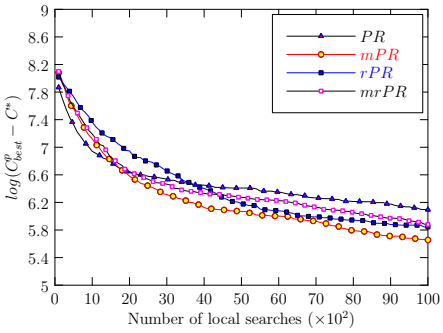
(c) GSM2-272-39



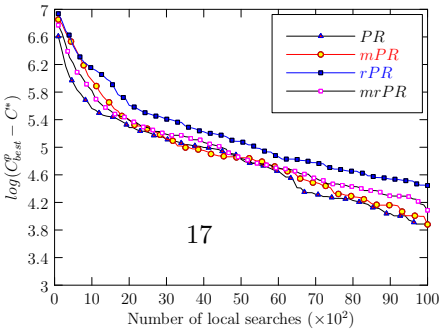
(d) P06-3-31



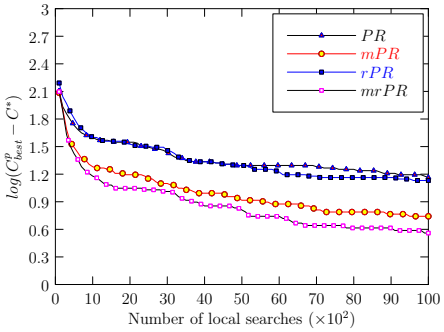
(e) GSM2-272-49



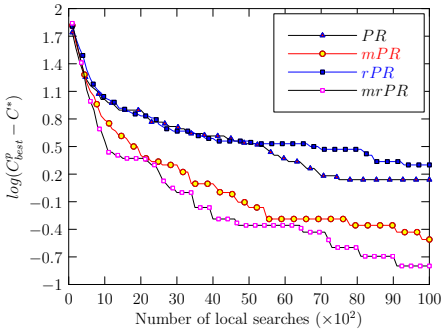
(f) GSM2-227-49



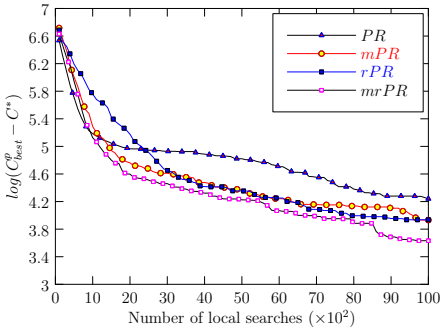
(g) GSM-246-21



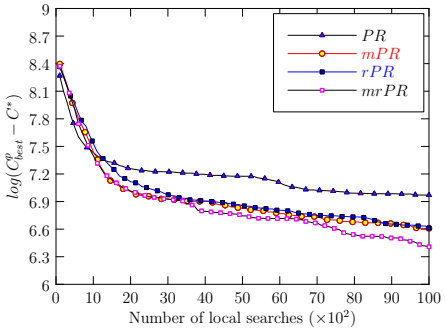
(h) GSM-246-31



(i) GSM2-184-39



(j) GSM2-227-39



(k) Test282-81

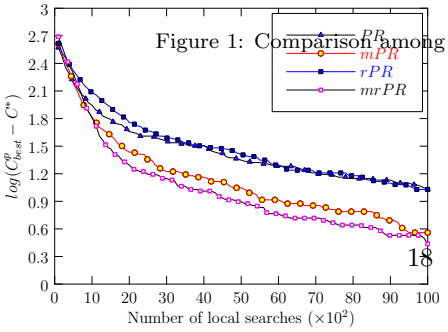


Figure 1: Comparison among the relinking operators

iterations (θ is set to 10^4), and the gap of the best objective value (C^p_{best}) in the population to the current best known result (C^*), i.e., $C^p_{best} - C^*$, is recorded as a function of the number of iterations. The evolution of the logarithm of $C^p_{best} - C^*$ is respectively plotted in Figure 1 for each instance and each relinking operator. Note that the results in this section are based on the average over 10 independent runs, and the logarithm of the gap is used just for the sake of the clarity of the curves.

One observes from Figure 1 that the relinking operators impact differently the performance of the path relinking algorithm. Specially, for the instances GSM2-227(29), GSMS-272(34), and GSMS-272(39), the rPR relinking operator performs the best while the PR relinking operator performs the worst. However, for P06-3(31) and GSM2-272(49), the mPR relinking operator is the best one, and the 3 remaining relinking operators differ significantly from each other. For GSM2-227(49), the PR relinking operator performs the best while the rPR relinking operator performs the worst. For the remaining 5 instances, the mrPR relinking operator performs the best.

This experiment shows clearly that for FS-FAP, the performance of relinking operator depends strongly on the structures of instances, and no relinking operator outperforms the other ones on all the instances. In this sense, the proposed relinking operators can be considered to be complementary and are useful to solve instances of different structures.

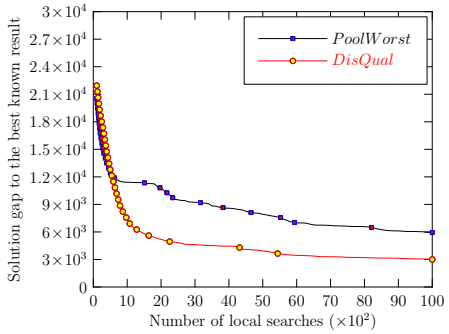
4.2. Importance of Diversity of Population

The second studied ingredient is the population updating strategy. In the present study, we employ a quality-and-distance based updating strategy (denoted by DisQual) to maintain the diversity of population. To show the effect of this strategy, we compare it with a traditional strategy (denoted by PoolWorst), that always replaces the worst solution in the population by the newly generated solution if it is better than the worst solution (s^w) in the population in terms of objective value.

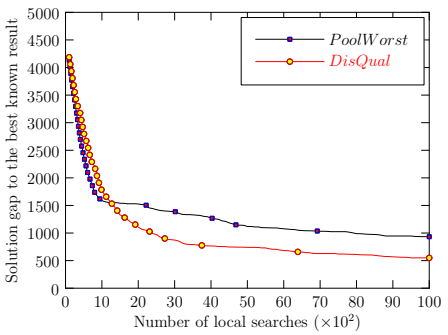
Similarly, taking the 11 selected instances as a test bed, the computational experiments are carried out as follows. First, the rPR algorithm with the DisQual strategy is independently run 10 times for each considered instance, and the best objective value (C^p_{best}) in the population is recorded as a function of the number of iterations at each run. Then, we displace the DisQual strategy by the PoolWorst strategy and keep other components unchanged, and the above experiment is again carried out by the resulting algorithm. The evolution of the gap of C^p_{best} to the current best known result is respectively plotted in Figure 2 for each instance and each strategy. Note that the results are based on the average values over 10 runs.

One can clearly observe that the DisQual strategy outperforms the traditional PoolWorst strategy for all the considered instances, which indicates that the adopted population updating strategy is relevant for the performance of the proposed algorithm. By considering both the quality and the distance to the existing solutions, the DisQual strategy is able to maintain the diversity of pop-

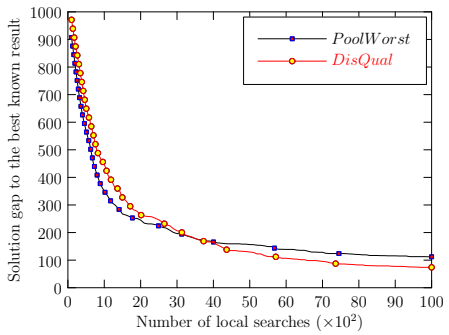
(a) GSM2-227-29



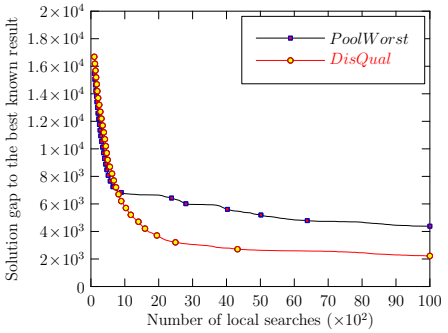
(b) GSM2-227-39



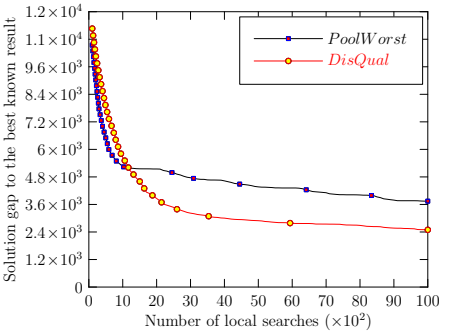
(c) GSM2-227-49



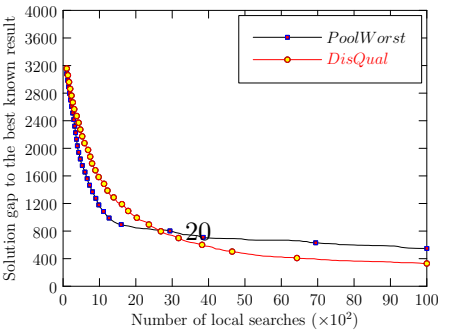
(d) GSM2-272-34



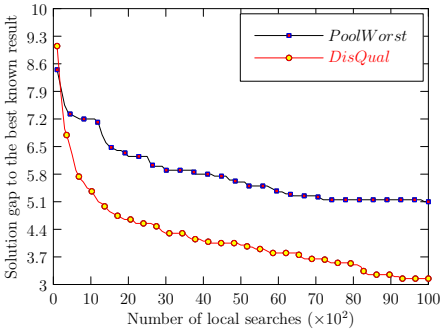
(e) GSM2-272-39



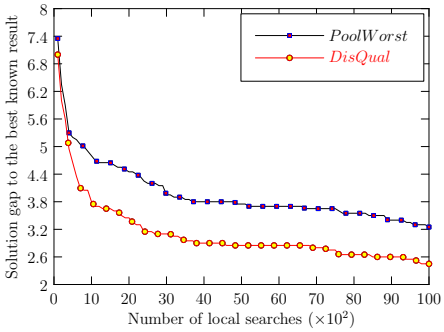
(f) GSM2-272-49



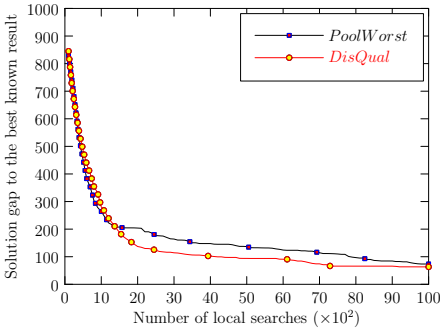
(g) GSM-246-21



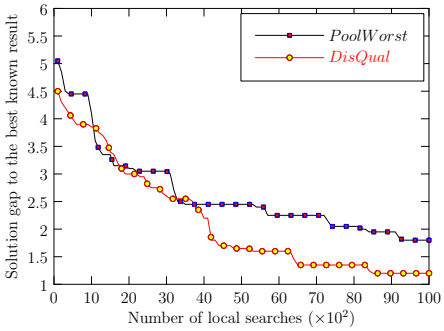
(h) GSM-246-31



(i) GSM2-184-39



(j) P06-3-31



(k) Test282-81

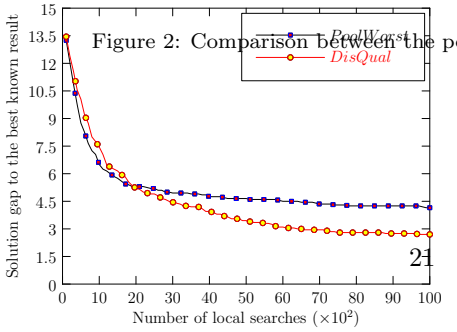
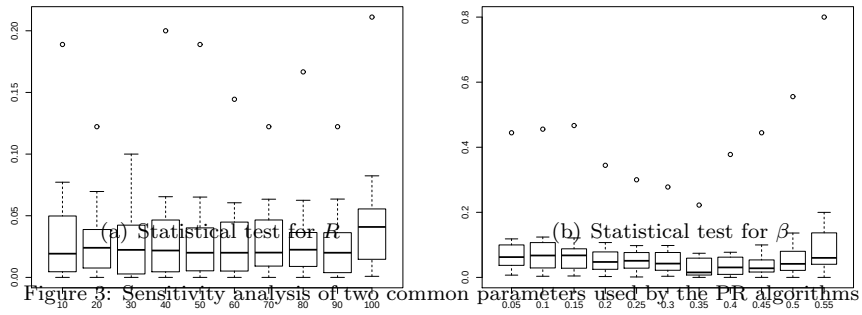


Figure 2: Comparison between the population updating strategies

ulation to avoid efficiently the premature convergence of population and thus enhance the performance of the proposed algorithm.

4.3. Sensitivity Analysis of Parameters



(a) Statistical test for R (b) Statistical test for β

Figure 3: Sensitivity analysis of two common parameters used by the PR algorithms

Our PR algorithms use several parameters. Hence, a sensitivity analysis of parameters is useful to know whether a given parameter is sensitive and to find an appropriate values for each parameter. In this study, we focus on two most important parameters, i.e., R which is used to determine the tabu tenure and β which is used in the population updating rule to control the population diversity.

For each of these parameters, we vary its value within a reasonable range and study their influences on our algorithm, while keeping the other parameters with their default values (as shown in Table 2). To compare the results in terms of solution quality, we employ the popular box and whisker plots based on the rPR algorithm and the 11 representative instances mentioned above.

First, we test the parameter R and show in the subfigure (a) of Figure 3 the results obtained with 10 different values $R \in [10, 100]$, where X-axis indicates the tested R values and Y-axis shows the solution quality (expressed as the percentage deviation of the average objective value over 10 independent runs from the best objective value yielded in this work). One observes from the figure that the different values of R lead to similar performances, indicating that the performance of the algorithm is not sensitive to the setting of R.

For the parameter β, we show in the subfigure (b) of Figure 3 the results achieved with 11 different $\beta \in [0.05, 0.55]$, where X-axis and Y-axis have the same meaning as for subfigure (a). From subfigure (b), one observes that the performance of the algorithm is significantly influenced by the value of parameter β, a too large or too small value will make the algorithm performs badly, and the rPR algorithm with a medium β value ($\beta = 0.35$) performs the best. Therefore, $\beta = 0.35$ is adopted as the default value by the PR algorithms.

5. Conclusions

In this paper, we proposed the first path relinking algorithm for solving the fixed spectrum frequency assignment problem. The main contribution of the proposed algorithm concerns the introduction of four relinking operators which are specifically designed to generate solution paths from existing solutions. A simple tabu search routine is also developed for the purpose of local optimization and a quality-and-distance strategy is introduced to update the population. We assessed the proposed algorithm on the set of 42 benchmark instances commonly used in the literature and obtained improved best results for 19 instances, while matching the best known results for 21 other instances. These improved results become new references for future algorithm comparisons and constitute our second contribution.

Although the proposed PR approach is very competitive in terms of both solution quality and computational efficiency compared with the state-of-the-art algorithms in the literature, there are some limitations. First, for several hard instances, the proposed PR algorithms missed the previous best known results, indicating the search capacity of our PR algorithms needs to be further enhanced. Second, the present study shows that the performance of the relinking operators depends largely on the structure of problem instances and no single relinking operator dominates other relinking operators on all instances, which means that it is difficult to obtain a high-quality result for all instances by means of a single path relinking operator.

Based on the present work and the identified limitations of our PR algorithms, we advance some research perspectives.

First, as mentioned before, the performance of the relinking operators depends on the structure of problem instances. Hence, in order to make the PR approach more efficient for various types of instances, it is interesting to integrate different relinking operators within a single PR algorithm. Second, learning is shown to be very efficient for the related bandwidth coloring problem Jin and Hao (2014). To reinforce the PR approach, it would be interesting to investigate learning mechanisms within important search components like the evaluation function or relinking operators. Finally, from a more general perspective, it is worth noting that the basic principle of the proposed PR approach is independent of the FS-FAP problem considered in this paper. Consequently, it would be valuable to examine its application to other frequency assignment problems and constraint satisfaction problems.

Acknowledgments

We are grateful to the reviewers for their useful comments which helps us to improve the paper. The work is partially supported by the LigeRo project (2009-2013, Region of Pays de la Loire, France), the PGMO project (2013-2015, Jacques Hadamard Mathematical Foundation) and a post-doc grant (for X.J. Lai) from the Region of Pays de la Loire (France).

References

Aardal, K.I., Van Hoesel, S.P.M., Koster, A.M.C.A., Mannino, C. & Sassano, A. (2007). Models and solution techniques for the frequency assignment problem. *Annals of Operations Research* 153(1), 79–121.

Audhya, G.K., Sinha, K., Mandal, K., Dattagupta, R., Ghosh, S.C., & Sinha, B.P. (2013). A new approach to fast near-optimal channel assignment in cellular mobile networks. *IEEE Transactions on Mobile Computing* 12(9), 1814–1827.

Castelino, D.J., Hurley, S. & Stephens, N.M. (1996). A tabu search algorithm for frequency assignment. *Annals of Operations Research* 63(2), 301–319.

Costa, W.E., Goldbarg, M.C. & Goldbarg, E.G. (2012). Hybridizing VNS and path-relinking on a particle swarm framework to minimize total flowtime. *Expert Systems with Applications* 39(18), 13118–13126.

Dorne, R. & Hao, J.K. (1995). An evolutionary approach for frequency assignment in cellular radio networks. *Proceedings of IEEE International Conference on Evolutionary Computation*, Perth, Australia, Nov.-Dec. IEEE Press., pp. 539–544.

Dorne, R. & Hao, J.K. (1996). Constraint handling in evolutionary search: A case study of the frequency assignment. *Parallel Problem Solving from Nature (PPSN IV), Lecture Notes in Computer Science* 1141, 801–810.

Dorne, R. & Hao, J.K. (1998). Tabu search for graph coloring, T-colorings and set T-colorings. *In S. Voss, S. Martello, I.H. Osman, C. Roucairol (Eds.)*, Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization. Chapter 6, pp. 77–92, Kluwer.

Duarte, A., Martí, R., Resende, M.G.C. & Silva, R.M.A (2011). GRASP with path relinking heuristics for the antibandwidth problem. *Networks* 58(3), 171–189.

Eisenblätter, A. & Koster, A. (2010). Frequency assignment websit. *http://fap.zib.de/index.php.*

Fischettti, M., Lepsch, C., Minerva, G., Romanin-Jacur, G. & Toto, E. (2000). Frequency assignment in mobile radio systems using branch-and-cut techniques. *European Journal of Operational Research* 123(2), 241–255.

Flood, I.D., & Allen, S.M. (2013). The fixed links frequency assignment problem with equipment selection. *Wireless Personal Communications* 71(1), 181–194.

Glover, F. (1997). A template for scatter search and path relinking. *Lecture Notes in Computer Science* 1363, 1–51.

Glover, F., Laguna, M. & Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control Cybernetics* 39, 653–684.

Hale, W.K. (1980). Frequency assignment: theory and applications. *Proceedings of the IEEE* 68(12), 1497–1514.

Hao, J.K., Dorne, R. & Galinier, P. (1998). Tabu search for frequency assignment in mobile radio networks. *Journal of Heuristics* 4(1), 47–62.

Hurley, S., Smith, D.H. & Thiel, S.U. (1997). FASoft: A system for discrete channel frequency assignment. *Radio Science* 32(5), 1921–1939.

Jin, M., Wu, H., Horng, J. & Tsai, C. (2001). An evolutionary approach to fixed channel assignment problems with limited bandwidth, *Proceeding of IEEE International Conference on Communications*, pp. 2100–2104.

Jin, Y., Hao, J.K. (2014) Effective learning-based hybrid search for bandwidth coloring. Accepted to *IEEE Transactions on Systems, Man and Cybernetics: Systems*, DOI:10.1109/TSMC.2014.2360661.

Johnson, D.S., Mehrotra, A. & Trick, M.A. (2008). Special issue on computational methods for graph coloring and its generalizations. *Discrete Applied Mathematics* 156(2), 145–156.

Kendall, G. & Mohamad, M. (2004). Solving the fixed channel assignment problem in cellular communications using an adaptive local search. *Proceedings of the 5th international conference on the Practice and Theory of Automated Timetabling (PATAT)*, Pittsburgh, USA, 18-20 August, 219–231.

Kim, S.S., Smith, A.E. & Lee, J.H. (2007). A memetic algorithm for channel assignment in wireless FDMA systems. *Computers and Operations Research* 34(6),1842–1856.

Král, D. (2005). An exact algorithm for the channel assignment problem. *Discrete Applied Mathematics* 145(2), 326–331.

Lai, X.J. & Lü, Z.P. (2013). Multistart iterated tabu search for bandwidth coloring problem. *Computers and Operations Research* 40(5) 1401–1409.

Lai, X.J., Lü, Z.P., Hao, J.K., Glover, F. & Xu, L.P. (2014). Path Relinking for Bandwidth Coloring Problem. http://arxiv.org/abs/1409.0973.

Lai, W.K. & Coghill, G.G. (1996). Channel assignment through evolutionary optimization. *IEEE Transactions on Vehicular Technology* 45(1), 91–95.

Lü, Z.P. & Hao, J.K. (2010). A memetic algorithm for graph coloring. *European Journal of Operational Research* 203(1), 241–250.

Mabed, H., Caminada, A. & Hao, J.K. (2011). Genetic Tabu Search for robust channel assignment under dynamic traffic data. *Computational Optimization and Applications* 50(3), 483–506.

Malaguti, E. & Toth, P. (2008). An evolutionary approach for bandwidth multicoloring problems. *European Journal of Operational Research* 189(3), 638–651.

Martins de Oliveira, R., Nogueira Lorena, L.A., Chaves, A.A. & Mauri, G.R. (2014). Hybrid heuristics based on column generation with path-relinking for clustering problems. *Expert Systems with Applications* 41(11), 5277–5284.

Maniezzo, V. & Carbonaro, A. (2000). An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems* 16(8), 927–935.

Montemanni, R., Smith, D.H. & Allen, S.M., (2001). Lower bounds for fixed spectrum frequency assignment problem. *Annals of Operational Research* 107(1-4), 237–250.

Montemanni, R., Moon J.N.J. & Smith, D.H. (2003). An improved tabu search algorithm for the fixed spectrum frequency assignment problem. *IEEE Transactions On Vehicular Technology* 52(3), 891–901.

Montemanni, R., Smith, D.H. & Allen, S.M., (2004). An improved algorithm to determine lower bounds for the fixed spectrum frequency assignment problem. *European Journal of Operational Research* 156(3), 736–751.

Montemanni, R. & Smith, D.H. (2010). Heuristic manipulation, tabu search and frequency assignment. *Computers and Operations Research* 37(3), 543–551.

Parejo, J.A., Segura, S., Fernandez, P. & Ruiz-Cortés, A. (2014). QoS-aware web services composition using GRASP with Path Relinking. *Expert Systems with Applications* 41(9), 4211–4223.

Park, E.J., Kim, Y.H. & Moon, B.R. (2002). Genetic search for fixed channel assignment problem with limited bandwidth. *In Genetic and Evolutionary Computation Conference*, pp. 1172–1179.

Porumbel, D.C., Hao, J.K. & Kuntz, P. (2010). An Evolutionary Approach with Diversity Guarantee and Well-Informed Grouping Recombination for Graph Coloring. *Computers and Operations Research* 37(10), 1822–1832.

Reeves, C.R. & Yamada, T. (1998). Genetic algorithms, path relinking and the flowshop sequencing problem. *Evolutionary Computation* 6(1), 230–234.

Roberts, F.S. (1991). T-colorings of graphs: recent results and open problems. *Discrete Mathematics* 93(2-3), 229–245.

San José-Revuelta, L.M. (2007). A new adaptive genetic algorithm for fixed channel assignment. *Information Sciences* 177(13), 2655–2678.

Smith, D.H., Hurley, S. & Thiel, S.U. (1998). Improving heuristics for the frequency assignment problem. *European Journal of Operational Research* 107(1),76–86.

Sörensen, K. & Sevaux, M. (2006). MA|PM: memetic algorithms with population management. *Computers and Operations Research* 33, 1214–1225.

Subramanian, A.P., Gupta, H., Das, S.R. & Jing, C. (2008). Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing* 7(12), 1459–1473.

Tiourine, S.R., Hurkens, C.A.J. & Lenstra, J.K. (2000). Local search algorithms for the radio link frequency assignment problem. *Telecommunication systems* 13, 293–314.

Wang, W. & Rushforth, C.K. (1996). An adaptive local-search algorithm for the channel-assignment problem (CAP). *IEEE Transactions on Vehicular Technology* 45(3),459–466.

Wang, Y., Lü, Z.P., Glover, F. & Hao, J.K. (2012). Path relinking for unconstrained binary quadratic programming. *European Journal of Operational Research* 223(3), 595–604.

Wu, J., Dai, Y., Zhao, Y.C. (2013). Effective channel assignments in cognitive radio networks. *Computer Communications* 36(4), 411–420.