Computers & Operations Research ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Heuristic manipulation, tabu search and frequency assignment

Roberto Montemanni[a,*], Derek H. Smith[b]

[a]Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), CH-6928 Manno, Switzerland
[b]Division of Mathematics and Statistics, University of Glamorgan, Pontypridd, CF37 1DL Wales, UK

ARTICLE INFO

Keywords:
Heuristic manipulation
Tabu search
Frequency assignment

ABSTRACT

Heuristic manipulation attempts to modify the search space of an optimization problem, using information provided by an underlying heuristic method. In this paper it is applied in combination with tabu search to the fixed spectrum frequency assignment problem.
The frequency assignment problem involves the assignment of discrete channels (or frequencies) to the transmitters of a radio network, such as a mobile telephone network. Frequency separation is necessary to avoid interference by other transmitters to the signal received from the wanted transmitter at the reception points. Unnecessary separation causes an excess requirement for spectrum. Good assignments minimize interference and the spectrum required. In fixed spectrum frequency assignment the frequency spectrum available is given and the target is to minimize the interference in the network.
Computational experiments confirm that the manipulation technique is able to drive the underlying tabu search algorithm towards improved solutions.

1. Introduction

Heuristic manipulation techniques (HMTs) are based on the idea that modifying the search space of an optimization problem, according to the information provided by an underlying heuristic method, could make the underlying method itself perform better, leading to a more effective algorithm. These concepts were first introduced in Montemanni et al. [1], where the idea is applied to the sequential ordering problem, working on top of an ant colony optimization metaheuristic (see Gambardella and Dorigo [2]). In this paper HMTs are tested in combination with tabu search for a frequency assignment problem.

Cellular telephone networks and other wireless communication networks have become widespread in recent decades. The problem of assigning frequencies to transmitters occurs when such a network is established or modified. The radio spectrum is a limited resource and the growth of demand has increased the importance of good network planning, aiming to use as few frequencies as possible. On the other hand, using fewer frequencies in a network generally leads to higher interference and thus a lower service quality. The aim of the fixed spectrum frequency assignment problem is to minimize (a measure of) the interference in the network when there is a given amount of spectrum available. Detailed reviews of frequency assignment problems and of methods for solving them can be found in Smith et al. [3], Murphey et al. [4], Aardal et al. [5] and in Section 4.2 of Correia [6].

Metaheuristic algorithms (i.e. iterative methods which, using particular strategies, drive a subordinate heuristic algorithm to explore the search space in an intelligent way) have been very successful for the frequency assignment problem. A complete collection of heuristic methods is provided by FASOFT (Hurley et al. [7]), where assorted constructive algorithms, local searches and metaheuristic algorithms are implemented.

One of the most successful of these metaheuristic algorithms for frequency assignment has been tabu search [8]. Some very effective tabu search algorithms have been presented by Hurley et al. [7], Hao et al. [9], Idoumghar and Debreux [10], Chiarandini and Stützle [11] and Montemanni et al. [12]. Their performance has been demonstrated on a variety of standard benchmark problems.

In the case of cellular problems (problems where more than one frequency has to be assigned to each node), assignments can be improved by interleaving the metaheuristic algorithm with exact cell optimization, or exact optimization of clusters of cells. This improved technique has been used by Hellebrandt and Heller [13], Montemanni et al. [12] and Mannino et al. [14]. The benefits of cell optimization combined with tabu search are precisely quantified in Montemanni et al. [12] for the case when run times are not specifically limited. Here we are interested in both cellular and non-cellular problems. We also wish to concentrate on the benefits of combining tabu search with heuristic manipulation when computation time is

* Corresponding author. Tel.: +41 91 610 8671; fax: +41 91 610 8661.
E-mail addresses: roberto@idsia.ch (R. Montemanni), dhsmith@glam.ac.uk (D.H. Smith).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.08.006

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

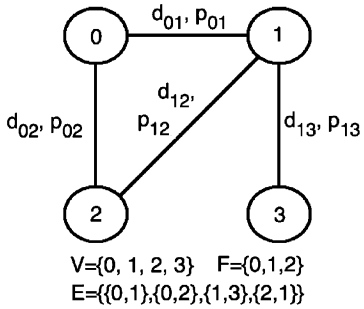

Fig. 1. Example of a graph for a fixed spectrum frequency assignment problem.

strictly limited. Thus cell optimization is not considered in this paper. In principle, however, all three techniques of tabu search, heuristic manipulation and cell optimization could be applied in combination.

The paper is organized as follows. In Section 2 the fixed spectrum frequency assignment problem is formally defined. The HMT we propose is described in Section 3, while the tabu search algorithm, used as the underlying method by the manipulation technique, is briefly summarized in Section 4. Extensive experimental results are presented and discussed in Section 5. Further experiments, carried out on some widely disseminated instances (with a somewhat different formulation), are reported in Section 6. Conclusions are drawn in Section 7.

2. The fixed spectrum frequency assignment problem

The fixed spectrum frequency assignment problem can be represented by a weighted undirected graph. Formally, a mapping $f : V \to F$ is required for a 5-tuple FS-FAP $= \{V, E, D, P, F\}$, where:

- V is a vertex set. Every vertex represents a transmitter of a radio network.
- E is a set of edges. Edges represent those pairs of transmitters for which the assigned frequencies are constrained. The constraint arises because one of the transmitters can interfere with signals received from the other.
- D is a set of labels for which $\forall\{i,j\} \in E\ \exists d_{ij} \in \mathcal{N}^+$. The value d_{ij} represents the highest separation between the frequency assigned to the transmitter i and the one assigned to j that generates unacceptable interference. If we indicate with $f(i)$ the frequency assigned to transmitter i, then if $|f(i) - f(j)| > d_{ij}$, the interference involving the two transmitters is acceptable.
- P is a set of labels for which $\forall\{i,j\} \in E\ \exists p_{ij} \in \mathcal{N}^+$. The value p_{ij} represents a cost to be paid if the separation between the frequencies of transmitters i and j is less than or equal to d_{ij}.
- F is a set of consecutive frequencies available for every vertex (transmitter) in V.

An example of such a graph is given in Fig. 1.

The objective of the fixed spectrum frequency assignment problem is to find an assignment f which minimizes the sum of p_{ij} over all pairs $\{i,j\} \in E$ for which $|f(i) - f(j)| \leqslant d_{ij}$. In this version of the problem we are given a fixed span of frequencies and we seek an assignment that minimizes (a measure of) the total interference over the network.

3. A problem manipulation technique

It is easy to observe that adding constraints to a given problem reduces its search space, making the problem potentially easier to solve. Starting from this observation, the heuristic problem manipulation technique (HMT) described in the remainder of this section has been developed.

First an underlying heuristic method is selected. The basic idea is to monitor the solutions generated by this heuristic, and to identify features common to solutions with a low objective value. Specifically, in this application, the technique identifies pairs of transmitters that tend to be assigned different frequencies in good solutions. At specified intervals new *artificial constraints* are added to the original problem to ensure that such pairs of transmitters are assigned distinct frequencies. The manipulated problem is likely to be easier than the original one, as it has a reduced solution space.

Of course such problem manipulation may cut out all the optimal solutions of the original problem, leading to suboptimal solutions even when the best solution of the modified problem is retrieved. To overcome this side effect, during the execution of the algorithm artificial constraints will not be added permanently, but may be substituted by other artificial constraints.

Notice that any heuristic method that produces a sequence of feasible solutions to the original problem can be used as the underlying method for the proposed manipulation approach. Most methods work in this way. In our case we will use the tabu search algorithm originally described in Montemanni et al. [12], which is summarized in Section 4.

Formally, the proposed methodology is built on top of an existing algorithm and makes use of an additional set of variables $\{m_{ij}\}$ which are stored in a matrix $M = [m_{ij}]$. Variable m_{ij} will be an indicator for the "quality" of the solutions in which nodes i and j are assigned different frequencies. The following parameters also need to be defined:

- I_{UP}: the time between two consecutive updates to the set of active artificial constraints. It is also the time before the first introduction of artificial constraints.
- N_{AC}: the number of artificial precedence constraints active at any time after their first introduction;
- N_{SC}: the number of artificial constraints substituted at regular intervals I_{UP} after the first generation of these constraints;
- I_{SH}: the interval between two consecutive updates of matrix M during the running of the underlying heuristic algorithm. In the case of tabu search it is measured by the number of tabu search iterations.

The HMT can be summarized as follows. The matrix M is initialized as $m_{ij}=0\ \forall\{i,j\} \in E$ and the set R of active artificial constraints (edges) is initialized as $R=\emptyset$. Let $Cost_1$ denote the cost of the first solution found by the underlying heuristic algorithm before any artificial constraints are added. $Cost_1$ plays the role of a normalization factor, and is used to avoid numerical problems. At the end of each interval of length I_{SH} let $Cost_k$ denote the cost of the current solution $Asgn_k$ found by the underlying heuristic (notice that in case $Cost_k = 0$ we can stop the computation since the optimal solution has been retrieved). Let $f_k(i)$ be the frequency assigned to vertex i in solution $Asgn_k$. Then matrix $M = [m_{ij}]$ is updated as follows:

$$m_{ij} = m_{ij} + \frac{Cost_1}{Cost_k} \quad \forall i,j \in V,\ f_k(i) \neq f_k(j), \{i,j\} \notin R. \tag{1}$$

Eq. (1) reinforces the entry corresponding to a pair of vertices assigned to different frequencies in solution $Asgn_k$. The increase is inversely proportional to the cost of the solution itself. If $Asgn_k$ is a good solution (i.e. $Cost_k$ is low) the entries of M corresponding to vertices associated with different frequencies in $Asgn_k$ will be increased considerably. Conversely, if $Cost_k$ is high, the relevant entries of matrix M will only be moderately increased. Entries corresponding to vertices associated with the same frequency in $Asgn_k$ are not updated. Notice that entries of matrix M increase monotonically

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

```
For each pair i,j ∈ V, i ≠ j   m_ij := 0
R := ∅
k := 1
Start underlying algorithm with constraints E
While computation time < fixed time limit do
    If I_SH solutions have been generated by the underlying heuristic
       since the last matrix update (or since the start) then
        Asgn_k := Current solution returned by the underlying heuristic
        Cost_k := Cost of Asgn_k
        If Cost_k = 0 then
            Return Asgn_best and Cost_best
        End If
        For each node i,j ∈ V such that f_k(i) ≠ f_k(j) and {i,j} ∉ R do
            m_ij := m_ij + Cost_1/Cost_k            /* Equation (1) */
        End For
        k := k + 1
    End If
    If Underlying algorithm has run for time I_UP then
        If (R = ∅) then     /* First iteration*/
            For i:=1 to N_AC do
                (r,s) = argmax_{{j,k}∉E, {j,k}∉R}{m_jk}
                R := R ∪ {{r,s}}
            End For
        Else     /* After first iteration*/
            For i:=1 to N_SC do
                (r,s) = argmin_{{j,k}∈R}{m_jk}
                R := R\{{r,s}}
            End For
            For i:=1 to N_SC do
                (r,s) = argmax_{{j,k}∉E, {j,k}∉R}{m_jk}
                R := R ∪ {{r,s}}
            End For
        End If
    Restart underlying algorithm with constraints R ∪ E
    End If
End While
Asgn_best := the best assignment from the beginning
Cost_best := the cost of Asgn_best
Return Asgn_best and Cost_best
```

Fig. 2. Pseudo-code for the problem manipulation technique for the FAP.

during the computation. Entries of the matrix M corresponding to active artificial constraints are not updated. This produces variation of the set of active constraints. Considering solutions only at intervals determined by I_{SH} avoids the relative values of the variables m_{ij} from remaining fairly static as a result of updating by very similar solutions. If this happened the method would not be very effective in exploring the whole search space.

It remains to clarify how the artificial constraints (which are edges $\{i,j\}$ with $d_{ij}=0$) are managed. After a first interval of length I_{UP}, an initial set of N_{AC} artificial constraints is added to the set R. The new constraints are selected as those not yet present in the graph with the highest entries in matrix M (with ties broken at random). These artificial constraints are added as *hard* constraints, i.e. with a cost $p_{ij}=+\infty$. Preliminary tests clearly suggested that this was the best choice in terms of performance of the technique. It should be noted that the parameters of the method are chosen so that it remains easy to find assignments which are feasible with respect to these hard constraints.

After the first artificial constraints have been added to the problem, at the end of subsequent intervals of length defined by I_{UP}, artificial constraints are updated by dropping N_{SC} constraints and substituting them by N_{SC} new constraints. The artificial constraints to drop are selected as those with the smallest entries in the matrix M (with ties broken at random). Conversely, the artificial constraints added are those with the highest entries in the same matrix M. Notice that, since entries of matrix M corresponding to active constraints are not reinforced (see Eq. (1)), it is likely that during the time they were active, entries corresponding to other inactive constraints have reached higher values. Such a strategy leads to a mechanism where artificial constraints are activated in turn. The mechanism should also prevent optimal solutions of the original problem from being permanently hidden by the active artificial constraints.

The pseudo-code in Fig. 2 summarizes the manipulation technique.

It is interesting to attempt to describe the method in terms of the ideas of intensification and diversification in tabu search (see

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

Gendreau [15]). Intensification is achieved in the method by the use of the artificial hard constraints to concentrate the search on promising regions of the search space. Continuous diversification is achieved in our approach by the rotation of the active artificial constraints. The method is compared with a multi-start (MS) application of the tabu search algorithm, which achieves a number of quite diverse solutions by starting from different randomly chosen starting assignments. It should finally be mentioned that, in contrast to the continuous diversification technique that is specific to tabu search, our method can run in combination with most other iterative heuristic algorithms.

4. The tabu search algorithm

The tabu search algorithm previously developed by Montemanni et al. [12] has been shown to be particularly effective and efficient for fixed spectrum frequency assignment problems. In this section we describe the general ideas of this tabu search algorithm for the *FS-FAP*.

Tabu search is a metaheuristic algorithm. It was first suggested in Glover [8]. The basic idea of the method is to partially explore the search space of all feasible solutions by a sequence of moves. At each iteration, the move carried out is the most promising among those available. A mechanism called a tabu list is present, which forbids a set of moves at each iteration. This aims to help the algorithm to escape from local (but not global) minima.

Formally, the main elements of the algorithm for the FS-FAP described in [12] are the following:

- *Solution representation*: The representation of a frequency assignment S is obtained by using a list $\{f_S(0), f_S(1), \ldots, f_S(|V|-1)\}$ where the vth element ($f_S(v)$) contains the frequency assigned to transmitter v.
- *Cost function*: The cost function Cost maps an assignment into the sum of the penalties paid in it. Formally we have

$$\text{Cost}(S) = \sum_{\{v,w\}\in E;\, |f_S(v)-f_S(w)| \leqslant d_{vw}} p_{vw}. \tag{2}$$

- *Neighbourhood*: An assignment S_N is in the neighbourhood of the current solution S_O if S_N differs from S_O in the frequency assigned to exactly one violating transmitter (defined as a transmitter incurring some penalty in assignment S_O) and the move which produces S_N from S_O is not in the tabu list. Defining $V^V_{S_O}$ as the set of violating transmitters in the assignment S_O, S_N is a neighbour of S_O if $\exists i \in V^V_{S_O} | f_{S_O}(i) \neq f_{S_N}(i)$ and $f_{S_O}(j) = f_{S_N}(j)\ \forall j \in V, j \neq i$ and the move $(i, f_{S_N}(i))$ is not in the tabu list.
- *Tabu list*: The tabu list contains the last T moves carried out. Moves in the tabu list are forbidden. A solution obtained from the current solution S with a move contained in the tabu list, cannot be a member of the neighbourhood of S. The tabu list contains pairs (i,f), where i is a transmitter and f is a frequency. Each time a move involving the assignment of frequency f to transmitter i is carried out, (i,f) is inserted into the tabu list, where it will remain for T iterations. Instead of using a tabu list with a fixed length T, as in the original scheme (see Glover [8]), T is varied during the running of the algorithm. In particular, the length of the tabu list starts from an initial value T_{init} and is reduced at regular intervals of I_{ts} iterations during the execution of the algorithm by the assignment $T := \beta T$ ($0 < \beta < 1$), where β, T_{init} and I_{ts} are user specified parameters.
- *Termination criterion*: The algorithm stops when T, the length of the tabu list, becomes smaller than a certain threshold $T_{\min}$, which is specified by the user.

At each iteration, the solution with the minimum cost among those in the neighbourhood becomes the new current solution.

It is important to notice that the algorithm described in Montemanni et al. [12] performs particularly well thanks to a sophisticated implementation, including the use of a *cost-change table*, which makes the method extremely fast. The reader is referred to [12] for further details.

5. Computational experiments with the formulation of Section 2

The aim of this section is to provide an experimental evaluation of the improvements generated by the problem manipulation technique described in Section 3, when it runs in combination with the tabu search algorithm, described in Montemanni et al. [12]. The combination will be denoted HMT. Tests with an MS version of the tabu search method, with a similar number of runs of the tabu search algorithm on unmodified problems, will allow a fair comparison to be made.

All the methods considered have been coded in ANSI C, starting from the original implementation of the tabu search algorithm (see [12]). All tests have been carried out on computers with a Dual AMD Opteron 250 2.6 GHz processor, with 4 GB of memory. In this section a maximum computation time of 2400 s has been allowed for each run. For the underlying tabu search algorithm, the same parameter settings indicated in [12] are retained (see also Section 5.2). These settings were producing runs of a maximum of 2500 s on the Intel Pentium II 400 MHz/128 MB used for the experiments reported in [12]. Since the machine we are using now is about 20 times faster (see Dongarra [16]), we expect to have at least 19 updates of artificial constraints for the HMT and a similar number of runs of the original tabu search algorithm in the MS case. These computation times should be adequate to reach a steady state, with further improvements to the best solution unlikely.

5.1. Description of the benchmarks adopted

The problems considered in this paper are those used in Montemanni [17] and Montemanni et al. [12], where an accurate description of the problems can be found.

In Table 1 we summarize the characteristics of the graphs on which the problem instances are based.[1] .

The meaning of the columns is as follows:

- Graph: names used to refer to the problems;
- $|V|$: number of vertices of the graph;
- $|E|$: number of edges of the graph;
- Avg d_{vw}: average frequency separation values in the graph;
- Avg p_{vw}: average penalty values in the graph.

From Table 1 we can notice that the graphs considered have heterogeneous characteristics, ranging from 45 to 282 nodes, with radically different edge densities and ranges for separation and penalty (d and p, respectively). We will undertake different tests on the same graph, changing the size of the spectrum available. For this reason there are more benchmarks (problem instances) than the number of graphs described in Table 1. Those problems for which optimal solutions can easily be found by the basic tabu search algorithm are not considered. The HMT has no possibility of improving tabu search solutions for these problems.

5.2. Parameter settings

The following parameter settings for the tabu search algorithm, described in Montemanni et al. [12] and Montemanni [17], are

[1] The graphs used in our experiments are available at http://www.idsia.ch/~roberto/FAP08.zip

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

Table 1
Problem characteristics

Graph	$\|V\|$	$\|E\|$	Avg d_{vw}	Avg 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	10 430	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	0.40	1.00
GSM2-184	184	6809	0.20	8.95×10^6
GSM2-227	227	10 088	0.18	9.10×10^6
GSM2-272	272	14 525	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

retained for the experiments reported in this paper. For all the problems we have fixed $T_{\min} = 10$, $\beta = 0.96$ and $I_{ts} = 5 \times 10^4$. The setting $T_{\text{init}} = 500$ is used for the problems based on AC-45-17, AC-45-25, AC-95-9, 1-1-50-75-30-2-50, 1-2-50-75-30-4-50, 1-3-50-75-30-0-50, 1-4-50-75-30-2-1, 1-5-50-75-30-2-100 and 1-6-50-75-30-0-1000. Similarly, $T_{\text{init}} = 1000$ is used for the problems based on AC-95-17, GSM-93, GSM-246, Test95, GSM2-184, GSM2-227 and GSM2-272. Finally $T_{\text{init}} = 2000$ for the remaining problems (based on Test282, P06-5, P06-3, P06b-5 and P06b-3).

As a comparison is made with an MS version of tabu search with the same parameters, all the improvements found by the proposed algorithm have to be due to the HMT.

The choice of parameter I_{UP} (interval between two consecutive updates of the artificial constraints) is immediate, since the parameter has to be the same as the length of a single run of the alternative MS algorithm. Each time a run of the tabu search is completed, the artificial constraints are updated.

In order to tune the remaining parameters I_{SH}, N_{AC} and N_{SC}, we proceed as follows. Six problems from the dataset are selected in such a way as to cover the different characteristics of the benchmarks. A reference value for each parameter is then selected. Starting from this standard setting, each parameter is then varied individually. Ten runs are completed for each parameter setting, and the results are examined. The columns of Tables 2–4 have the following meaning:

- Problem: Identifies the problem instances, which are obtained, in turn, by considering a graph (with separations and labels) and a number of consecutive available frequencies ($|F|$).
- Parameter (N_{AC}, N_{SC}, I_{SH}) setting: Identifies the values for the parameter under investigation considered for each problem.
- Average: Contains the average results (in term of cost of best solutions retrieved) obtained by the algorithm over 10 runs, given the parameter setting under investigation.
- Best: Contains the best results (in term of cost of best solutions retrieved) obtained by the algorithm over 10 runs, given the parameter setting under investigation.
- Worst: Contains the worst results (in term of cost of best solutions retrieved) obtained by the algorithm over 10 runs, given the parameter setting under investigation.

We start by tuning parameter N_{AC}, regulating the number of active artificial constraints. For these tests we set $I_{\text{SH}} = 1000$ and N_{SC} depending on N_{AC} itself, in such a way that $\lfloor 0.4 \cdot N_{\text{AC}} \rfloor \leqslant N_{\text{SC}} \leqslant \lfloor 0.5 \cdot N_{\text{AC}} \rfloor$. In Table 2 three different values of N_{AC} are considered. By considering the central value for N_{AC} in each case, it clearly emerges that only a small number of artificial constraints are necessary for the technique to work. The rationale is that if too many constraints are used, the technique tends to hide good solutions and does not converge to the best possible solutions. This happens because the selection of the active artificial constraints is heuristic, and therefore the probability of selecting the "wrong" constraints exists, and is increased when too many artificial constraints are active at the same time. On the other hand, if the number of artificial constraints is too small, the benefits of the HMT are not realized, making the resulting algorithm more or less equivalent to the underlying method adopted.

Table 2
Tuning of parameter N_{AC} ($\lfloor 0.4 \cdot N_{\text{AC}} \rfloor \leqslant N_{\text{SC}} \leqslant \lfloor 0.5 \cdot N_{\text{AC}} \rfloor$; $I_{\text{SH}} = 1000$)

Graph	$\|F\|$	N_{AC}	Average	Best	Worst
GSM-93	9	3	32.4	32	33
		10	32.2	32	33
		50	33.1	32	34
GSM-246	31	5	26.2	26	27
		20	26.1	25	27
		100	26.2	25	27
Test282	81	10	11.9	10	13
		50	11.9	10	13
		250	12.2	10	13
GSM2-184	52	10	270.2	183	311
		50	260.6	162	287
		250	267.3	186	311
GSM2-272	34	10	60 637.9	57 672	64 353
		50	58 691.4	56 128	64 353
		250	60 770.9	58 169	64 353
1-5-50-75-30-2-100	10	3	186.8	176	199
		10	183.8	176	199
		50	188.0	177	199

Table 3
Tuning of parameter N_{SC} (N_{AC} takes the central value considered in Table 2; I_{SH}=1000)

Graph	$\|F\|$	N_{SC}	Average	Best	Worst
GSM-93	9	2	32.5	32	34
		5	32.2	32	33
		8	33.6	32	34
GSM-246	31	2	26.5	26	27
		10	26.1	25	27
		18	26.2	26	27
Test282	81	2	12.1	10	13
		20	11.9	10	13
		48	12.3	11	13
GSM2-184	52	2	277.1	186	311
		20	267.3	186	311
		48	292.7	217	342
GSM2-272	34	2	58 917.2	56 222	64 353
		20	58 691.4	56 128	64 353
		48	59 635.2	57 672	64 353
1-5-50-75-30-2-100	10	2	187.8	177	199
		5	183.8	176	199
		8	186.5	176	199

The second parameter under examination is N_{SC}, regulating the number of artificial constraints substituted at each iteration of the HMT. In Table 3 we report the results obtained on the selected problems when different values of N_{SC} are considered, while N_{AC} takes the central values considered in Table 2, and $I_{\text{SH}} = 1000$.

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

Table 4
Tuning of parameter I_{SH} (N_{AC} and N_{SC} take the central values considered in Tables 2 and 3)

Graph	$\|F\|$	I_{SH}	Average	Best	Worst
GSM-93	9	10	33.0	32	34
		1000	32.2	32	33
		100 000	32.4	32	34
GSM-246	31	10	26.3	25	27
		1000	26.1	25	27
		100 000	26.2	25	27
Test282	81	10	12.3	11	13
		1000	11.9	10	13
		100 000	12.3	10	13
GSM2-184	52	10	273.0	186	342
		1000	267.3	186	311
		100 000	271.1	186	342
GSM2-272	34	10	59 959.2	57 715	64 353
		1000	58 691.4	56 128	64 353
		100 000	58 917.2	56 222	64 353
1-5-50-75-30-2-100	10	10	187.1	177	196
		1000	183.8	176	199
		100 000	187.6	176	199

Examination of the table suggests that the central values considered (which are always in the range $\lfloor 0.4 \cdot N_{AC} \rfloor \leqslant N_{SC} \leqslant \lfloor 0.5 \cdot N_{AC} \rfloor$) lead to the best results. The rationale behind the result is the following: if the rotation is too slow, the examination of the search space remains too local; if the rotation is too fast, non-interesting parts of the search space tend to be examined.

Parameter I_{SH}, which regulates the number of iterations of the underlying heuristic between two consecutive updates of matrix M of the HMT, also has to be tuned. This parameter has to be fairly small in order to have an accurate view of the search space, but at the same time, it has to be large enough to prevent the relative values of the variables m_{ij} from remaining fairly static as a result of updating by very similar solutions. This would lead the manipulation technique to concentrate on a small subset of the search space only, and therefore to a myopic view of the whole search space. This last aspect is particularly important when using an underlying algorithm like tabu search, which tends to work around a local minimum for some iterations, before moving to another local optimum.

In Table 4 three different values of I_{SH} are considered, with N_{AC} and N_{SC} taking the values suggested by the previous experiments. The results support our choice of $I_{SH} = 1000$, and this value will be used for the remainder of the paper.

The parameter tuning adopted for the remaining results in this section are summarized in Table 5, where columns have the following meaning:

- Graph: name of the problem;
- N_{AC}: the number of active artificial constraints;
- N_{SC}: the number of artificial constraints substituted at each update;
- I_{SH}: the number of iterations of the underlying tabu search algorithm between two consecutive updates of matrix M.

A comparison of Tables 1 and 5 suggests that the number of artificial constraints is related to $|E|$, i.e. to the number of constraints of the problem. For $|E| \leqslant 3000$ we have $N_{AC} = 10$ and $N_{SC} = 5$. For $3000 < |E| \leqslant 9000$, $N_{AC} = 20$ and $N_{SC} = 10$, and for $|E| > 9000$ $N_{AC} = 50$ and $N_{SC} = 20$. The only exception to this rule is represented by problem GSM2-184, for which a setting with $N_{AC} = 50$ and $N_{SC} = 20$ gave better results, notwithstanding a number of edges smaller than 9000. Parameter I_{SH} is invariant and takes the value 1000 for all the graphs considered.

Table 5
Parameter setting

Graph	N_{AC}	N_{SC}	I_{SH}
AC-45-17	10	5	1000
AC-45-25	10	5	1000
AC-95-9	10	5	1000
AC-95-17	10	5	1000
GSM-93	10	5	1000
GSM-246	20	10	1000
Test95	10	5	1000
Test282	50	20	1000
P06-5	20	10	1000
P06-3	50	20	1000
P06b-5	20	10	1000
P06b-3	50	20	1000
GSM2-184	50	20	1000
GSM2-227	50	20	1000
GSM2-272	50	20	1000
1-1-50-75-30-2-50	10	5	1000
1-2-50-75-30-4-50	10	5	1000
1-3-50-75-30-0-50	10	5	1000
1-4-50-75-30-2-1	10	5	1000
1-5-50-75-30-2-100	10	5	1000
1-6-50-75-30-0-1000	10	5	1000

5.3. Comparison between MS and HMT

In this section we compare the results obtained by the MS tabu search algorithm and those retrieved with the use of the HMT in combination with the tabu search algorithm. Column headings of Table 6 have the following meaning:

- Problem: Identifies the problem instances, which are obtained, in turn, by considering a graph (with separations and labels) and a number of consecutive available frequencies ($|F|$).
- Average: Contains the average results (in term of cost of best solutions retrieved) obtained by the algorithms (MS and HMT) over 10 runs. In column %Imp the percentage improvement of HMT over MS (average case) is reported.
- Best: Contains the best results (in term of cost of best solutions retrieved) obtained by the algorithms (MS and HMT) over 10 runs. In column %Imp the percentage improvement of HMT over MS (best case) is reported.
- Worst: Contains the worst results (in term of cost of best solutions retrieved) obtained by the algorithms (MS and HMT) over 10 runs. In column %Imp the percentage improvement of HMT over MS (worst case) is reported.
- NT: Problem instances for which not all the runs of the two algorithms produce a solution with the same cost are marked here with an asterisk (NT denotes non-trivial).

Some lines of Table 6 are in italics. They correspond to problem instances not considered in [12,17], but that have been added here (by considering new values for $|F|$) in order to have a wider and more significant set of benchmarks. Some entries of the table are in bold: they correspond to new best solution costs, that improve previously best-known solutions (or best solutions to new problems).

The last but one line of Table 6 reports the average improvements (over all instances) of the average, best and worst results over the 10 runs considered, given by the HMT. The very last line of the table is analogous to the previous one, but averages are computed only for instances marked with an asterisk in column NT. This excludes instances that can be conjectured to be solved to optimality already. This last line should give an estimate of the real improvements guaranteed by the HMT.

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

Table 6
Computational results

Problem		Average			Best			Worst			NT
Graph	$\|F\|$	MS	HMT	%Imp	MS	HMT	%Imp	MS	HMT	%Imp	
AC-45-17	7	32.0	32.0	–	32	32	–	32	32	–	
AC-45-17	9	15.0	15.0	–	15	15	–	15	15	–	
AC-45-25	11	33.0	33.0	–	33	33	–	33	33	–	
AC-95-9	6	31.0	31.0	–	31	31	–	31	31	–	
AC-95-17	15	33.0	33.0	–	33	33	–	33	33	–	
AC-95-17	21	10.0	10.0	–	10	10	–	10	10	–	
GSM-93	9	33.0	32.2	2.42	32	32	–	34	33	2.94	*
GSM-93	13	7.0	7.0	–	7	7	–	7	7	–	
GSM-246	21	80.6	80.2	0.50	79	79	–	82	81	1.22	*
GSM-246	31	26.3	26.1	0.76	25	25	–	27	27	–	*
Test95	36	8.0	8.0	–	8	8	–	8	8	–	
Test282	61	53.4	53.2	0.37	51	51	–	56	55	1.79	*
Test282	71	29.4	29.3	0.34	27	27	–	30	30	–	*
Test282	*81*	*12.2*	*11.9*	*2.46*	*11*	***10***	*9.09*	*13*	*13*	–	*
P06-5	11	133.0	133.0	–	133	133	–	133	133	–	
P06-3	31	115.0	115.0	–	115	115	–	115	115	–	
P06b-5	21	52.0	52.0	–	52	52	–	52	52	–	
P06b-5	31	25.0	25.0	–	25	25	–	25	25	–	
P06b-3	31	112.0	112.0	–	112	112	–	112	112	–	
P06b-3	71	26.0	26.0	–	26	26	–	26	26	–	
GSM2-184	39	5642.8	5598.8	0.78	5481	**5447**	0.62	5758	5689	1.20	*
GSM2-184	49	1073.4	1043.6	2.78	999	874	12.51	1143	1120	2.01	*
GSM2-184	*52*	*277.9*	*260.6*	*6.23*	*186*	***162***	*12.90*	*311*	*287*	*7.72*	*
GSM2-227	*29*	*68 077.7*	*66 510*	*2.30*	***61 586***	***61 586***	–	*70 105*	*70 105*	–	*
GSM2-227	39	11 170.3	10 897.7	2.44	10 979	**10 550**	3.91	11 276	11 164	0.99	*
GSM2-227	49	2649.1	2613.1	1.36	2459	2459	–	2828	2828	–	*
GSM2-272	*34*	*60 473.2*	*58 691.4*	*2.95*	*57 715*	***56 128***	*2.75*	*67 025*	*64 353*	*3.99*	*
GSM2-272	39	28 484.3	28 488.2	−0.01	27 416	27 416	–	29 323	29 307	0.05	*
GSM2-272	49	8043.8	7946.7	1.21	7785	7785	–	8411	8459	−0.57	*
1-1-50-75-30-2-50	5	1254.1	1253.9	0.02	1242	1242	–	1260	1260	–	*
1-1-50-75-30-2-50	10	105.3	103.8	1.42	101	**97**	3.96	109	109	–	*
1-1-50-75-30-2-50	11	65.5	66.1	−0.92	59	59	–	69	70	−1.45	*
1-1-50-75-30-2-50	*12*	*39.6*	*38.7*	*2.27*	*38*	***36***	*5.26*	*42*	*42*	–	*
1-2-50-75-30-4-50	*9*	*680.9*	*680.6*	*0.04*	***671***	***671***	–	*691*	*691*	–	*
1-2-50-75-30-4-50	11	326.6	325.0	0.49	323	**317**	1.86	337	335	0.59	*
1-3-50-75-30-0-50	*7*	*197.1*	*196.5*	*0.30*	*196*	***194***	*1.02*	*198*	*199*	*−0.51*	*
1-4-50-75-30-2-1	*6*	*71.0*	*70.9*	*0.14*	*71*	***70***	*1.41*	*71*	*71*	–	*
1-4-50-75-30-2-1	10	19.0	19.0	–	19	19	–	19	19	–	
1-5-50-75-30-2-100	10	191.6	183.8	4.07	186	**176**	5.38	197	199	−1.02	*
1-5-50-75-30-2-100	*12*	*70.5*	*69.3*	*1.70*	*65*	***63***	*3.08*	*74*	*74*	–	*
1-6-50-75-30-0-1000	10	7123.4	7064.3	0.83	6942	**6840**	1.47	7279	7267	0.16	*
1-6-50-75-30-0-1000	*13*	*1389.6*	*1365.2*	*1.76*	*1318*	***1207***	*8.42*	*1490*	*1440*	*3.36*	*
Average				0.93			1.75			0.54	
Average over non trivial (NT)				1.44			2.73			0.83	

The results of Table 6 confirm that adding artificial precedence constraints is advantageous. First of all, it virtually never leads to worse average results, and even when the average is worse, it is an insignificant deterioration. This fact is particularly important because it indicates that the use of artificial precedence constraints can be recommended: in the worst case the quality of the solutions is (almost) the same, but often the additional implementation effort is justified by improved results.

If we concentrate our attention on the columns reporting the best results obtained by the methods considered over 10 runs, we can see that the same conclusions drawn for the average case, are still valid. The only difference is that in the latter case the improvement seems to be amplified. This confirms that the use of the manipulation technique really leads to better results when the search space is properly modified (i.e. the manipulation technique works as we expect).

The section of Table 6 concerning the worst results over the 10 runs considered shows that the improvements guaranteed by artificial precedence constraints are in line with those obtained for the average results (although slightly worse). This aspect is important because it indicates that artificial precedence constraints do not tend to hide "good" solutions. Notice that in a very small number of cases this is not true; in these cases the HMT was probably unable to correctly interpret the search-space, and drove the underlying heuristic to suboptimal solutions.

Statistical tests on the results provided by the two algorithms considered have also been performed. A paired t-test indicates that the difference between the results of the two methods is extremely statistical significant ($p < 0.0001$), and is clearly in favour of *HMT*.

6. Experiments on COST 259 problems

In this section heuristic manipulation is evaluated on some of the COST 259 benchmarks.[2] The formulation of these benchmarks is more complex than that described in Section 2. The modifications necessary to handle the COST 259 problems will now be indicated. Some constraints are *hard* and must be satisfied. This is handled by giving them a large penalty, much larger than the likely final solution (10 000 is used here). Other constraints with $d_{vw} = 1$ have two different levels of penalty, with the higher penalty being applied if v and w are assigned the same frequency. Penalties need not be integers. Finally, some frequencies may not be used, and the unusable frequencies may vary from transmitter to transmitter. Thus the set F of consecutive frequencies available for every vertex (transmitter)

[2] The instances are available at http://fap.zib.de/problems/COST259.

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

Table 7
Computational results on COST 259 instances

Problem	Average			Best			Worst		
	MS	HMT_a		MS	HMT_a		MS	HMT_a	
	Val	Val	%Imp	Val	Val	%Imp	Val	Val	%Imp
Siemens 1	2.8979	2.8492	1.68	2.8937	2.7642	4.48	2.9411	2.8475	3.18
Siemens 2	15.1973	15.0578	0.92	15.1392	14.9360	1.34	15.2634	15.1276	0.90
Siemens 3	6.9403	6.7358	2.95	6.8592	6.6496	3.06	7.0515	6.8150	3.35
Siemens 4	114.2577	112.4817	1.55	113.3240	110.9725	2.08	115.3547	113.7577	1.38
K	0.5209	0.4886	6.20	0.5125	0.4647	9.33	0.5316	0.5089	4.27
Average			2.66			4.06			2.62
	MS	HMT_b		MS	HMT_b		MS	HMT_b	
	Val	Val	%Imp	Val	Val	%Imp	Val	Val	%Imp
Siemens 1	2.8979	2.8697	0.97	2.8937	2.7889	3.62	2.9411	2.9119	0.99
Siemens 2	15.1973	15.2879	−0.60	15.1392	15.1756	−0.24	15.2634	15.1392	0.81
Siemens 3	6.9403	6.7273	3.07	6.8592	6.6884	2.49	7.0515	6.8196	3.29
Siemens 4	114.2577	112.7155	1.35	113.3240	112.4765	0.75	115.3547	113.1545	1.91
K	0.5209	0.5153	1.08	0.5125	0.5057	1.33	0.5316	0.5248	1.28
Average			1.17			1.59			1.66

is replaced by a set $F = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{|V|}\}$ of frequency domains. Vertex i must be assigned a frequency in $\mathcal{F}_i$.

The problems *Siemens* 1, *Siemens* 2, *Siemens* 3, *Siemens* 4 and *K* are all cellular GSM problems and have 929, 977, 1623, 2785 and 267 transmitters, respectively. They all are cellular problems (i.e. more than one frequency has to be assigned to some node locations). As observed in Section 1, for this kind of problem an important improvement can be achieved by using cell optimization (see, for example, Mannino et al. [14]). The use of this technique is outside the scope of this paper. Moreover, it has to be observed that the best known assignments for these problems have typically been found after very long computations (sometimes measured in weeks). As previously observed, the aim of this paper is to show that the use of HMTs can improve the performance of a tabu search algorithm, given a limited run time. For these reasons we do not expect to retrieve the assignments that match the best available in the tests presented in this section. The same machine used for the experiments reported in Section 5 is used, and a maximum run time of 7200 s is considered.

Based on the experience documented in Montemanni et al. [12], and on a calibration phase, the following parameter setting for the tabu search parameters were chosen: $T_{\text{init}} = 5000$ (Siemens problems), $T_{\text{init}} = 3000$ K, $\beta = 0.96$ and $I_{ts} = 5 \times 10^4$. These settings guarantee at least 10 runs of the tabu search method for each problem in the given time. Some preliminary tests also suggested two promising values for parameters of the HMT. In the reminder of this section we will refer to HMT_a as the configuration with $N_{AC} = 200$, $N_{SC} = 100$ and $I_{SH} = 1000$, and to HMT_b as the alternative configuration with $N_{AC} = 400$, $N_{SC} = 200$ and $I_{SH} = 1000$. Notice that the parameter setting rules identified in Section 5.2 do not apply for this different family of problems.

Column headings of Table 7 have the following meaning:

- Problem: Identifies the problem instances.
- Average: Contains the average results (in term of cost of best solutions retrieved) obtained by the algorithms (MS, HMT_a or HMT_b) over five runs. In columns Val the results found by the methods are summarized, while columns %Imp contain the percentage improvement of HMT_a or HMT_b over MS (average case).
- Best: Contains the best results (in term of cost of best solutions retrieved) obtained by the algorithms (MS, HMT_a or HMT_b) over five runs. In columns Val the results found by the methods are summarized, while columns %Imp contain the percentage improvement of HMT_a or HMT_b over MS (best case).
- Worst: Contains the worst results (in term of cost of best solutions retrieved) obtained by the algorithms (MS, HMT_a or HMT_b) over five runs. In columns Val the results found by the methods are summarized, while columns %Imp contain the percentage improvement of HMT_a or HMT_b over MS (worst case).

The rows of Table 7 labelled "Average" report the average improvements (over all instances) of the average, best and worst results over the 10 runs considered, given by the two configurations of the HMT considered.

All the observation and conclusions drawn in Section 5.3 for the experiments reported in Table 6 are valid also for the experiments summarized in Table 7, notwithstanding the improvements guaranteed by the HMT are more substantial in Table 7. This suggests that the technique we propose is potentially more useful when used on difficult instances.

7. Conclusion

A manipulation technique, aiming at enhancing the performance of known heuristic methods, has been applied to a frequency assignment problem.

Experimental results show that the technique, which is inherently simple and capable of being applied on top of most heuristic algorithms, improves the results obtained by the underlying tabu search algorithm considered here.

It should finally be stressed that the proposed problem manipulation method is not only applicable to tabu search methods for frequency assignment problems. It can be adapted to other combinatorial optimization methods. The issue in each case is to identify proper artificial constraints (based on the structure of the problem under investigation) that can usefully modify the solution space in the right manner.

References

[1] Montemanni R, Smith DH, Gambardella LM. A heuristic manipulation technique for the sequential ordering problem. Computers and Operations Research 2008;35:3931–44.
[2] Gambardella LM, Dorigo M. An ant colony system hybridized with a new local search for the sequential ordering problem. INFORMS Journal on Computing 2000;12(3):237–55.
[3] Smith DH, Allen SM, Hurley S, Watkins WJ. Frequency assignment: methods and algorithms. In: Proceedings of the NATO RTA SET/ISET symposium on frequency assignment, sharing and conservation in systems (aerospace), Aalborg, Denmark, October 1998, NATO RTO-MP13, 1999. p. K.1–K.18.

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006

[4] Murphey RA, Pardalos PM, Resende MGC. Frequency assignment problems. In: Du D-Z, Pardalos PM, editors. Handbook of combinatorial optimization, Supplement Volume A. Dordrecht: Kluwer Academic Publishers; 1999. p. 295–397.
[5] Aardal KI, van Hoesel SPM, Koster AMCA, Mannino C, Sassano A. Models and solution techniques for frequency assignment problems. 4OR 2003;1(4): 261–317.
[6] Correia LM, editor. Wireless flexible personalized communications—COST 259: European co-operation in mobile radio research. New York: Wiley; 2001. COST Action 259—Final Report.
[7] Hurley S, Smith DH, Thiel SU. FASoft: a system for discrete channel frequency assignment. Radio Science 1997;32(5):1921–39.
[8] Glover F. Heuristics for integer programming using surrogate constraints. Decision Science 1977;8:156–66.
[9] Hao J-K, Dorne R, Galinier P. Tabu search for frequency assignment in mobile radio networks. Journal of Heuristics 1998;4:47–62.
[10] Idoumghar L, Debreux P. New modeling approach for the frequency assignment problem in broadcasting. IEEE Transactions on Broadcasting 2002;48(4):293–8.
[11] Chiarandini M, Stützle T. Stochastic local search algorithms for graph set T-coloring and frequency assignment. Constraints 2007;12(3):371–403.
[12] Montemanni R, Moon JNJ, Smith DH. An improved tabu search algorithm for the fixed spectrum frequency assignment problem. IEEE Transactions on Vehicular Technology 2003;52(4):891–901.
[13] Hellebrandt M, Heller H. A new heuristic method for frequency assignment. Technical Report TD(00)003, COST259, Valencia, Spain; January 2000.
[14] Mannino C, Oriolo G, Ricci F, Chandran S. The stable set problem and the thinness of a graph. Operations Research Letters 2007;35(1):1–9.
[15] Gendreau M. An introduction to tabu search. In: Glover F, Kochenberger GA, editors. Handbook of metaheuristics. New York: Springer; 2003. p. 37–54.
[16] Dongarra JJ. Performance of various computers using standard linear algebra software in a fortran environment. Technical Report CS-89-85, University of Tennessee; January 2008.
[17] Montemanni R. Upper and lower bounds for the fixed spectrum frequency assignment problem. PhD thesis, University of Glamorgan; 2001 ⟨http://www.idsia.ch/~roberto/papers/Th2side.pdf⟩.

Please cite this article as: Montemanni R, Smith DH. Heuristic manipulation, tabu search and frequency assignment. Computers and Operations Research (2008), doi:10.1016/j.cor.2008.08.006