ELSEVIER

Discrete Applied Mathematics 78 (1997) 251–282

DISCRETE APPLIED MATHEMATICS

A potential reduction approach to the frequency assignment problem ☆

J.P. Warners [a,*,1], T. Terlaky [a], C. Roos [a], B. Jansen [b,2]

[a] *Department of Statistics, Probability and Operations Research, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands*
[b] *CQM B.V., P.O. Box 414, 5600 AK, Eindhoven, The Netherlands*

Received 7 November 1995; revised 5 November 1996

Abstract

The frequency assignment problem is the problem of assigning frequencies to transmission links such that either no interference occurs, or the amount of interference is minimized. We present an approximation algorithm for this problem that is inspired by Karmarkar's interior point potential reduction approach to combinatorial optimization problems. A non convex quadratic model of the problem is developed, that is very compact as all interference constraints are incorporated in the objective function. Moreover, optimizing this model may result in finding multiple solutions to the problem simultaneouly. Several preprocessing techniques are discussed. We report on computational experience with both real-life and randomly generated instances.

Keywords: Interior point methods; Nonlinear optimization; Combinatorial optimization; Binary programming; Frequency assignment

1. Introduction

In this paper we are concerned with developing an interior point potential reduction approach to solve the Frequency Assignment Problem (*FAP*). This problem arises in practice when a network of radio links has to be established. Each radio link has to be assigned a frequency, subject to a number of *interference* constraints. An interference constraint gives the minimal required distance (in mHz) between the frequencies assigned to a couple of links. If this constraint is violated, communicaton using these links will be distorted. In practice, interference will occur when two links which are situated near each other, are assigned the same or close frequencies.

☆ This research was partially supported by the EUCLID program, SEPA 6 (Artificial Intelligence), RTP 6.4 (Combinatorial Algorithms for Military Applications). We are grateful to CELAR (Centre d'ELectrique de l'ARmement) for providing real-life test instances.

* Corresponding author. E-mail: j.p.warners@twi.tudelft.nl.

[1] Partially supported by the Dutch Organization for Scientific Research (NWO/SION) under grant 612-33-001.

[2] Supported by the Dutch Organization for Scientific Research (NWO) under grant 611-304-028.

PII S0166-218X(96)00139-4

We will consider instances of the *FAP* with the following objective. Given a set of frequencies, a set of links and a set of interference constraints:

- Try to assign each link a frequency such that the number of violated interference constraints is minimal.
- If a feasible assignment exists, try to assign each link a frequency such that all the interference constraints are satisfied, and the number of used frequencies is minimal.

The *FAP* is NP-complete, since the graph coloring problem, which can be considered as a special case of the *FAP*, is NP-complete [3, 12]. For NP-complete problems no polynomial-time algorithms are known; the worst-case complexity of exact algorithms is an exponential function of the problem size. However, in practice exact algorithms may behave much better than their worst case. For instance, for the *FAP* a branch-and-cut algorithm has been developed [1] which turned out to be quite effective. Still, finding a solution and proving its optimality requires substantial computation times. Hence approximation algorithms must be developed, i.e. algorithms that find a *good*, but not necessarily optimal solution within reasonable time. Examples of such algorithms (that also have been applied to the *FAP*) are local search, genetic and graph based algorithms [3, 12, 17, 18]. The interior point potential reduction method discussed in this paper also is an approximation algorithm. It is inspired by the research of Karmarkar et al. [6, 9, 10].

In 1984 Karmarkar showed that linear programming problems can be solved in polynomial time by an interior point method [8], improving upon the polynomial worst-case bound of Khachiyan's ellipsoid algorithm for linear programming [11]. More recently, he and his colleagues extended the interior point approach to combinatorial optimization problems. Promising results on the *satisfiability* problem [6] and the *set covering* problem [10] are reported. The idea is to formulate the combinatorial optimization problem under consideration as a binary feasibility problem with linear constraints, relax the integrality constraints, and add a (concave) quadratic objective function that forces the variables to binary values. Subsequently, a nonconvex potential function is introduced, whose minimizers are feasible solutions of the binary problem. An interior point method is used to sequentially minimize the potential function. Since the potential function is nonconvex, a local (nonbinary) minimum may be found. To generate feasible solutions more quickly, after each iteration the current interior point is rounded to a binary solution.

In this paper we will develop a nonconvex quadratic model for the *FAP*, in which all interference constraints are incorporated in the objective function. Optimizing this model may result in finding multiple feasible assignments simultaneously. In [19, 20] it is shown that this model may also be obtained from a binary linear model for the *FAP*. To solve the model, an appropriate potential function is designed and minimized by the interior point method Karmarkar et al. [6, 10] propose. In our algorithm we also incorporate preprocessing techniques, rounding schemes and techniques to escape from local minima.

This paper is organized as follows. In Section 2 we will construct various nonconvex quadratic models for the *FAP*. Furthermore, it is briefly indicated how these models can be derived from linear models for the *FAP*. In Section 3 the interior point method to solve the quadratic model is discussed, and specific details on the application of the algorithm to the *FAP* are given. Section 4 contains computational results on both real-life and randomly generated instances of the *FAP*. The real-life instances were provided by CELAR, the random instances were generated using the TU Delft developed test problem generator GRAPH [2]. Finally, Section 5 contains some concluding remarks.

2. Mathematical model of the frequency assignment problem

2.1. Definitions and notation

In order to model the *FAP*, the following notation is used:

$\mathscr{L}$: set of L radio links;

$\mathscr{F}$: set of F frequencies;

$\mathscr{F}_l$: frequency domain of link $l \in \mathscr{L}$ ($\mathscr{F}_l \subseteq \mathscr{F}$);

$f \to l$: the assignment of frequency $f \in \mathscr{F}_l$ to link $l \in \mathscr{L}$;

$\mathscr{D}$: set of pairs of links for which an interference constraint must be satisfied;

d_{lk}: minimal required frequency distance for the links $(l,k) \in \mathscr{D}$.

So

$$\mathscr{D} = \{(l,k) \mid \text{if } f \to l \text{ and } g \to k \text{ then } |f-g| \geqslant d_{lk},\ l,k \in \mathscr{L},\ f \in \mathscr{F}_l,\ g \in \mathscr{F}_k\}.$$

We define the following decision variables:

$$x_{lf} = \begin{cases} 1 & \text{if } f \to l, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall l \in \mathscr{L},\ \ \forall f \in \mathscr{F}_l.$$

To enable our model to deal with more sophisticated instances of the *FAP* we introduce some additional definitions.

p_{lk}: priority of the constraint concerning links $(l,k) \in \mathscr{D}$;

τ_i: penalty if a constraint with priority i is violated;

f_l^{pre}: preinstalled frequency for link l;

mob_l: mobility for link l;

θ_j: penalty if to a link l with mobility j a frequency other than f_l^{pre} is assigned.

It is assumed that $\tau_i, \theta_j \in \mathbb{N}$ and that $\tau_i > \tau_{i+1}, \theta_j > \theta_{j+1}$. For the priorities p_{lk} and the mobilities mob_l the following holds:

$$p_{lk} \in \{0, 1, \ldots, p^{\min}\}, \qquad \forall (l,k) \in \mathscr{D};$$

$$mob_l \in \{0, 1, \ldots, mob^{\max}\}, \quad \forall l \in \mathscr{L}.$$

If a constraint has priority 0, it must be satisfied, if it has priority $i \in \{1, \ldots, p^{\min}\}$ it may be violated at cost τ_i. Similarly, if a link has mobility 0, its preinstalled frequency may not be changed, while if it has mobility $j \in \{1, \ldots, mob^{\max} - 1\}$ its preinstalled frequency may be changed at cost θ_j. A mobility of $mob^{\max}$ means that no frequency has been preinstalled.
Before going on, we introduce some more notation:

$|\mathscr{L}|$: the number of elements of set $\mathscr{L}$;

$l^* = \arg\max_{l \in \mathscr{L}} |\mathscr{F}_l|$: the element $l^* \in \mathscr{L}$ for which $|\mathscr{F}_l|$ is maximal;

$f^* \Rightarrow l$: $f^* \rightarrow l$ and the exclusion of interfering assignments, i.e. $x_{lf^*} := 1$; $x_{lf} := 0, \ \forall f \in \mathscr{F}_l \backslash \{f^*\}$; $x_{kg} := 0, \ \forall k \in \mathscr{L}$, $g \in \mathscr{F}_k$, such that $|f^* - g| < d_{lk}$;

sgn(x): equals 1, 0, -1 if x is positive, zero, negative.

e: all-one vector.

In the following, by an *assignment*, an *assignment* x or a *feasible assignment*, we usually mean a *full* assignment, i.e. each link is assigned exactly one frequency. Sometimes, by an assignment we mean a *single* assignment, i.e. $f \rightarrow l$ for some $l \in \mathscr{L}$ and $f \in \mathscr{F}_l$; this will be clear from the context. A *pair of assignments* refers to $f \rightarrow l$ and $g \rightarrow k$ for some $l, k \in \mathscr{L}$ and $f \in \mathscr{F}_l$, $g \in \mathscr{F}_k$. Furthermore, if an instance of the *FAP* allows a feasible assignment, we refer to it as a *feasible FAP*. Other instances of the *FAP* are referred to as *infeasible FAP*s.

2.2. A quadratic model

In this section we will derive quadratic models for the *FAP*. For each class we have the requirement that exactly one frequency is assigned to each link. This can be modelled as:

$$\sum_{f \in \mathscr{F}_l} x_{lf} = 1, \quad \forall l \in \mathscr{L}. \tag{1}$$

In the following subsection it is shown how the interference constraints are modelled and subsequently it is indicated in which way the objective to minimize the number of frequencies used is incorporated. Finally, a more general formulation of the *FAP* is given, which enables us to deal with interference constraints of different importances, and partially fixed assignments.

2.2.1. Feasibility version of the FAP

Let us define the coefficients q_{lfkg}:

$$q_{lfkg} = \begin{cases} 1 & \text{if } |f-g| < d_{lk}, \quad \forall (l,k) \in \mathscr{D},\ f \in \mathscr{F}_l,\ g \in \mathscr{F}_k, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Now we can write the feasibility version of the *FAP* as the minimization of a (nonconvex) quadratic function subject to the constraints that to every link exactly one frequency must be assigned (1) and all the variables must be binary:

$$(P_Q) \quad \begin{array}{ll} \min & \frac{1}{2} \sum_{l \in \mathscr{L}} \sum_{f \in \mathscr{F}_l} \sum_{k \in \mathscr{L}} \sum_{g \in \mathscr{F}_k} q_{lfkg} x_{lf} x_{kg} = \frac{1}{2} x^{\mathrm{T}} Q x \\ \text{s.t.} & \sum_{f \in \mathscr{F}_l} x_{lf} = 1, \quad \forall l \in \mathscr{L}, \\ & x_{lf} \in \{0,1\}, \quad \forall l \in \mathscr{L}, \quad \forall f \in \mathscr{F}_l. \end{array}$$

In (P_Q), Q is an $m \times m$ matrix, with $m = \sum_{l \in \mathscr{L}} |\mathscr{F}_l|$ the number of variables, containing the elements q_{lfkg}. The matrix Q has the following structure:

$$Q = \begin{pmatrix} 0 & Q_{12} & \dots & Q_{1L} \\ Q_{21} & 0 & \dots & Q_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{L1} & Q_{L2} & \dots & 0 \end{pmatrix}. \tag{3}$$

The $|\mathscr{F}_l| \times |\mathscr{F}_k|$ submatrix Q_{lk} represents the interference constraint concerning the links l and k. Obviously, if no such constraint exists we have $Q_{lk} = 0$. Furthermore, $Q_{lk} = Q_{kl}^{\mathrm{T}}$, for all $l, k \in \mathscr{L}$, so Q is symmetric. Since the diagonal of Q contains only zeroes, Q is indefinite.

The relaxation of (P_Q) will be called (R_Q). In (R_Q) the integrality constraints are replaced by $0 \leqslant x_{lf} \leqslant 1$, $\forall l \in \mathscr{L}$, $\forall f \in \mathscr{F}_l$.

Lemma 1. *The nonconvex quadratic optimization problem (R_Q) has the following properties:*
1. $x^{\mathrm{T}} Q x \geqslant 0$ *for any feasible solution x.*
2. *If x is a feasible assignment for the original FAP then it holds that $x^{\mathrm{T}} Q x = 0$.*
3. *If $x^{\mathrm{T}} Q x = 0$ then x yields one or more feasible assignments for the original FAP.*

Proof.
(1) As both Q and any feasible solution x of (R_Q) contain only nonnegative elements, we conclude that $x^{\mathrm{T}} Q x \geqslant 0$.
(2) If we are given a feasible assignment x, there is no combination l, f, k, g such that $q_{lfkg} = x_{lf} = x_{kg} = 1$, so $x^{\mathrm{T}} Q x = 0$.
(3) Given a solution x of (R_Q) such that $x^{\mathrm{T}} Q x = 0$. This implies that there is no combination l, f, k, g for which $q_{lfkg} = 1$ and both x_{lf} and x_{kg} are positive. Letting $\tilde{x} = \mathrm{sgn}(x)$ (thus $\tilde{x}$ is binary), $\tilde{x}^{\mathrm{T}} Q \tilde{x} = 0$. The assignment corresponding to $\tilde{x}$ may be *overfull*,

i.e. to a link more than one frequency might be assigned, but we can simply set variables from one to zero until each link is assigned exactly one frequency. This completes the proof. □

It may be worthwhile to observe that *any* assignment x is a feasible solution of (R_Q). As will be clear from the proof of Lemma 1, for a given assignment x the objective value $\frac{1}{2}x^{\mathrm{T}}Qx$ is equal to the number of violated interference constraints. On the other hand, the next lemma states that from a fractional (not necessarily optimal) solution x of (R_Q), a solution $\tilde{x}$ of (P_Q) can be constructed that has an objective value that is at least as good as the objective value of x.

Lemma 2. *Let $\xi(x)=\frac{1}{2}x^{\mathrm{T}}Qx$. Given a feasible nonintegral solution $\bar{x}$ of (R_Q), we can construct a feasible integral solution $\tilde{x}$, such that $\xi(\tilde{x}) \leqslant \xi(\bar{x})$.*

Proof. We construct $\tilde{x}$ link by link. Let $x := \bar{x}$ and consider a link l^* for which two or more variables x_{l^*f}, $f \in \mathscr{F}_{l^*}$, are fractional. Using the symmetry of Q and the fact that $q_{lflg}=0, \forall l \in \mathscr{L}, \forall f, g \in \mathscr{F}_l$, (definition (2)), we can rewrite $\xi(x)$ as:

$$\xi(x) = \sum_{f \in F_{l^*}} c_f x_{l^*f} + K, \tag{4}$$

where the *cost coefficients* c_f are given by

$$c_f = \sum_{k \in \mathscr{L} \setminus \{l^*\}} \sum_{g \in \mathscr{F}_k} q_{l^*fkg} x_{kg} \geqslant 0, \qquad f \in \mathscr{F}_{l^*},$$

and

$$K = \frac{1}{2} \sum_{l \in \mathscr{L} \setminus \{l^*\}} \sum_{f \in \mathscr{F}_l} \sum_{k \in \mathscr{L} \setminus \{l^*\}} \sum_{g \in \mathscr{F}_k} q_{lfkg} x_{lf} x_{kg} \geqslant 0.$$

Note that both K and the cost coefficients are independent of the values of the variables x_{l^*f}. Therefore we can simply take f^* to be given by

$$f^* = \arg \min_{f \in \mathscr{F}_{l^*}} c_f.$$

and set

$$\tilde{x}_{l^*f^*} = 1; \qquad \tilde{x}_{l^*f} = 0, \ \forall f \in \mathscr{F}_{l^*} \setminus \{f^*\}; \qquad \tilde{x}_{lf} := x_{lf}, \ \forall l \in \mathscr{L} \setminus \{l^*\}, \ \forall f \in \mathscr{F}_l.$$

Then, using (4) and the definitions of $\tilde{x}$ and c_f,

$$\begin{aligned} \xi(\tilde{x}) - K &= \sum_{f \in \mathscr{F}_{l^*}} c_f \tilde{x}_{l^*f} = c_{f^*} = \min_{f \in \mathscr{F}_{l^*}} c_f = \left[\min_{f \in \mathscr{F}_{l^*}} c_f \right] \sum_{f \in \mathscr{F}_{l^*}} x_{l^*f} \\ &\leqslant \sum_{f \in \mathscr{F}_{l^*}} c_f x_{l^*f} = \xi(x) - K. \end{aligned}$$

If $\tilde{x}$ is not binary, we let $x := \tilde{x}$ and repeat this procedure. When $\tilde{x}$ is binary, we have constructed an integral solution such that $\xi(\tilde{x}) \leqslant \xi(\bar{x})$. □

Since all the input data are integral, an interesting corollary of this lemma is that the objective value of any *strict* minimizer (local or global) of (R_Q) is integral. Moreover, assuming that a (fractional) minimizer x has integral objective value ξ, using Lemma 2 one can construct multiple assignments, each of which violates exactly ξ constraints. The number of assignments N_x thus obtainable is given by the following formula:

$$N_x = \prod_{l \in \mathscr{L}} \left[\sum_{f \in \mathscr{F}_l} \operatorname{sgn}(x_{lf}) \right]. \tag{5}$$

2.2.2. *Minimizing the number of used frequencies*

The quadratic model has to be slightly extended in order to be able to minimize the number of used frequencies. We introduce the additional variables z_f:

$$z_f = \begin{cases} 0 & \text{if the frequency } f \text{ is assigned to at least one link,} \\ 1 & \text{otherwise,} \end{cases} \quad \forall f \in \mathscr{F}. \tag{6}$$

Furthermore, let us define the additional coefficients r_{glf} by

$$r_{glf} = \begin{cases} 1 & \text{if } f = g, \\ 0 & \text{otherwise,} \end{cases} \quad \forall g \in \mathscr{F},\ \forall l \in \mathscr{L},\ \forall f \in \mathscr{F}_l.$$

Let R be the $|\mathscr{F}| \times m$ matrix containing the elements r_{glf}. The matrix R has the following structure:

$$R = (R_1 \quad R_2 \quad \ldots \quad R_L). \tag{7}$$

The $|\mathscr{F}| \times |\mathscr{F}_l|$ submatrix R_l indicates which variable x_{lf}, $f \in \mathscr{F}_l$, corresponds to which frequency; it has a 1 in position f of row g if and only if $f = g$.

Lemma 3. *For an assignment x and corresponding z it holds that*

$$\sum_{g \in \mathscr{F}} \sum_{l \in \mathscr{L}} \sum_{f \in \mathscr{F}_l} r_{glf} z_g x_{lf} = z^{\mathrm{T}} R x = 0.$$

Proof. For a given assignment x, with $x_{lf} = 1$, $r_{glf} x_{lf} = 1$ if and only if $g = f$. Hence

$$\sum_{l \in \mathscr{L}} \sum_{f \in \mathscr{F}_l} r_{glf} x_{lf} = \# \text{ of times the frequency } g \text{ is used.}$$

The lemma follows immediately by the definition of z_g. □

Letting

$$\bar{Q} := \begin{pmatrix} Q & R^{\mathrm{T}} \\ R & 0 \end{pmatrix}, \qquad y := \begin{pmatrix} x \\ z \end{pmatrix}, \tag{8}$$

the *FAP* can be written as

$$
(P_{\overline{Q}}) \quad
\begin{aligned}
\min \quad & \tfrac{1}{2} y^{\mathrm{T}} \overline{Q} y \\
\text{s.t.} \quad & \sum_{f \in \mathscr{F}_l} x_{lf} = 1, \ \forall l \in \mathscr{L}, \\
& \sum_{f \in \mathscr{F}} z_f = M, \\
& z_f \in \{0,1\}, \ \forall f \in \mathscr{F}, \\
& x_{lf} \in \{0,1\}, \forall l \in \mathscr{L}, \ \forall f \in \mathscr{F}_l,
\end{aligned}
$$

where $F–M$ is the maximal number of frequencies to be used. (Recall that $F = |\mathscr{F}|$). The relaxation $(R_{\overline{Q}})$ of $(P_{\overline{Q}})$ is obtained by relaxing the integrality constraints to linear inequalities.

Lemma 4. *The nonconvex quadratic optimization problem* $(R_{\overline{Q}})$ *has the following properties*:
1. $y^{\mathrm{T}} \overline{Q} y \geqslant 0$ *for any feasible solution* y.
2. *If* y *yields a feasible assignment for the FAP that uses no more than* $F - M$ *frequencies then we have* $y^{\mathrm{T}} \overline{Q} y = 0$.
3. *If* $y^{\mathrm{T}} \overline{Q} y = 0$ *then* y *yields one or more feasible assignments for the original FAP that use no more than* $F - M$ *frequencies.*

Proof.
(1) As both $\overline{Q}$ and any feasible solution y of $(R_{\overline{Q}})$ contain only nonnegative elements, $y^{\mathrm{T}} \overline{Q} y \geqslant 0$.
(2) Using (8), we have that $y^{\mathrm{T}} \overline{Q} y = x^{\mathrm{T}} Q x + 2 z^{\mathrm{T}} R x$. If the assignment x uses *exactly* $F - M$ frequencies, we find by Lemmas 1 and 3 that $y^{\mathrm{T}} \overline{Q} y = 0$. If x is a feasible assignment that uses *less* than $F–M$ frequencies, we need to modify z. We have $x^{\mathrm{T}} Q x = z^{\mathrm{T}} R x = 0$, but by definition (6) we have that $\sum_{f \in \mathscr{F}} z_f \geqslant M + 1$, so strictly speaking y is not a feasible solution of $(R_{\overline{Q}})$. However, we may set some variables z_f from one to zero until y is feasible. Obviously, this does not change the value of $z^{\mathrm{T}} R x$.
(3) Suppose now that y is an optimal solution of $(R_{\overline{Q}})$ with zero value. As for a solution y of $(R_{\overline{Q}})$ we require that $\sum_{f \in \mathscr{F}} z_f = M$, we find that at least M of the z-variables are greater than zero, so at most $F - M$ of the z-variables are equal to zero. So, since there is no combination g, l such that both x_{lg} and z_g are positive, at most $F–M$ frequencies are used. For the term $x^{\mathrm{T}} Q x$, Lemma 1 applies. This proves the lemma. □

We observe that the proof of Lemma 2 can straightforwardly be extended to this situation.

Note that, given a minimizer $y=(x,z)^{\mathrm{T}}$ of $(R_{\bar{Q}})$ for which two or more z_f variables are fractional, all assignments that can be constructed from x use *less* than $F-M$ frequencies. To compute the number of *assignments* with the same objective value, we can again use (5), since it is not necessary to take into account the number of fractional z_f variables. To compute the number of different *binary solutions* however, the number N_x obtained by (5) must be multiplied by $Z!/M!(Z-M)!$, where $Z=\sum_{f\in\mathscr{F}}\mathrm{sgn}(z_f)$.

2.2.3. Example

We give a small example of the construction of the matrices Q and R. Let

$$\mathscr{L}=\{1,2,3,4\} \quad \text{and} \quad \mathscr{F}=\{10,20,30,50\};$$

$$\mathscr{F}_1=\mathscr{F}_4=\{10,20,50\}, \qquad \mathscr{F}_2=\mathscr{F}_3=\mathscr{F};$$

$$\mathscr{D}=\{(1,2),(1,4),(2,3)\}.$$

The required frequency distances d_{lk} are given by the following matrix:

$$D=\begin{pmatrix} - & 12 & - & 25 \\ 12 & - & 7 & - \\ - & 7 & - & - \\ 25 & - & - & - \end{pmatrix}.$$

We now can construct the matrices Q and R submatrix by submatrix (see (3) and (7)).

- The submatrices Q_{13}, Q_{24}, Q_{34} (and their transposed counterparts) are all-zero matrices;
- $Q_{23}=I_{4\times 4}$;
- The other submatrices are given by

$$Q_{12}=\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad Q_{14}=\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Now we construct the submatrices R_l, $l=1,\dots,4$.
- $R_2=R_3=I_{4\times 4}$ since $\mathscr{F}_2=\mathscr{F}_3=\mathscr{F}$;
- Furthermore,

$$R_1=R_4=\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The third row contains all zeroes since $30\notin\mathscr{F}_1$, $30\notin\mathscr{F}_4$.

2.2.4. Minimizing the cost of interference

We now generalize our model to enable us to deal with more sophisticated instances of the *FAP*. If the interference constraints have different priorities, the different penalties can be taken into account in the following way. Instead of setting q_{lfkg} to one if its corresponding pair of assignments violates an interference constraint, we define the coefficients $\tilde{q}_{lfkg}$ to be equal to the penalty of the assignments $f \to l$ and $g \to k$. If a constraint has priority 0, i.e. it must be satisfied, we can set the corresponding penalty τ_0 to a large number, such that $\tau_0 \gg \tau_1$. So,

$$\tilde{q}_{lfkg} = \begin{cases} \tau_i & \text{if } |f-g| < d_{lk} \text{ and } p_{lk} = i, \quad \forall (l,k) \in \mathscr{D}, \ \forall f \in \mathscr{F}_l, \ \forall g \in \mathscr{F}_k, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Let $\widetilde{Q}$ be the matrix containing the elements $\tilde{q}_{lfkg}$; its structure is similar to that of Q in (3).

If preinstalled frequencies and mobilities are given, one may choose to add the mobility costs to the diagonal of the matrix $\widetilde{Q}$. Another possibility is to introduce a linear penalty term in the objective function. The second option results in a model which has more attractive properties, as then Lemma 2 straightforwardly can be extended. Thus the vector v is defined as follows:

$$v_{lf} = \begin{cases} \theta_j & \text{if } mob_l = j \text{ and } f \neq f_l^{\text{pre}}, \\ 0 & \text{if } f = f_l^{\text{pre}}, \end{cases} \quad \forall l \in \mathscr{L}, \quad \forall f \in \mathscr{F}_l. \tag{10}$$

Lemma 5. *The cost of a given assignment x is equal to $\frac{1}{2}x^{\mathrm{T}}\widetilde{Q}x + v^{\mathrm{T}}x$.*

Proof. The cost incurred by the pair of assignments $f \to l$, $g \to k$ equals $\tilde{q}_{lfkg}$. As $\widetilde{Q}$ is symmetric, the term $\tilde{q}_{lfkg}x_{lf}x_{kg}$ occurs twice, so to find the cost we have to divide the total sum by two. Furthermore, if $f \to l$, $l \in \mathscr{L}$, with $mob_l = j$ and $f \neq f_l^{\text{pre}}$, then a penalty $\theta_j = v_{lf}$ has to be paid. □

Now we can write the *FAP* as:

$$(P_{\widetilde{Q}}) \quad \begin{array}{ll} \min & \frac{1}{2}x^{\mathrm{T}}\widetilde{Q}x + v^{\mathrm{T}}x \\ \text{s.t.} & \sum_{f \in \mathscr{F}_l} x_{lf} = 1, \quad \forall l \in \mathscr{L}, \\ & x_{lf} \in \{0,1\}, \quad \forall l \in \mathscr{L}, \ \forall f \in \mathscr{F}_l. \end{array}$$

The relaxation of $(P_{\widetilde{Q}})$ will be called $(R_{\widetilde{Q}})$. The minimum of $(P_{\widetilde{Q}})$ is the optimal value of the original *FAP* (Lemma 5). Due to the following lemma, the optimal values $(R_{\widetilde{Q}})$ and $(P_{\widetilde{Q}})$ are equal.

Lemma 6. *Let $\xi(x) = \frac{1}{2}x^{\mathrm{T}}\widetilde{Q}x + v^{\mathrm{T}}x$. Given a feasible nonintegral solution $\bar{x}$ of $(R_{\widetilde{Q}})$, we can construct a feasible integral solution $\tilde{x}$, such that $\xi(\tilde{x}) \leqslant \xi(\bar{x})$.*

Proof. Essentially the same as the proof of Lemma 2. The only change is that a linear term is added to the cost coefficients c_f, i.e.

$$c_f := v_{l^*f} + \sum_{k \in \mathscr{L}\setminus\{l^*\}} \sum_{g \in \mathscr{F}_k} \tilde{q}_{l^*fkg}x_{kg} \geqslant 0.$$

Again, the values of c_f are independent of the values $x_{l^* f}$, so the proof of Lemma 2 applies to this situation. □

Using (5), one can again compute the number of assignments with the same objective value that can be constructed from a minimizer with integral objective value.

Note that if instead of introducing a linear term, we add the mobility costs to the diagonal of the matrix $\tilde{Q}$, the minima of $(P_{\tilde{Q}})$ and $(R_{\tilde{Q}})$ are in general not equal.

2.3. An alternative way to derive the quadratic model

In the previous section we have constructed the nonconvex quadratic model for the *FAP* in a straightforward manner. In this section we indicate how it can be derived from a linear model for the *FAP*. This linear model has certain properties which make it possible to give an alternative expression for the matrices Q and $\overline{Q}$ (see also [19, 20]). The matrix $\tilde{Q}$ can also be derived from a linear model, but this requires a little more effort.

2.3.1. Linear models for the FAP

We use the notation introduced in Section 2.1. As described in Section 2.2, we need to assign exactly one frequency to each link. This results in constraints of the form (1). The interference constraints can be modelled as follows:

$$x_{lf} + x_{kg} \leqslant 1, \quad \forall (l,k) \in \mathscr{D},\ \forall f \in \mathscr{F}_l,\ g \in \mathscr{F}_k: \quad |f-g| < d_{lk}. \tag{11}$$

Note that the number of constraints may be reduced significantly by combining constraints of the form (11), thus obtaining:

$$x_{lf} + \sum_{g \in \mathscr{F}_k : |f-g| < d_{lk}} x_{kg} \leqslant 1, \quad \forall (l,k) \in \mathscr{D},\ \forall f \in \mathscr{F}_l. \tag{12}$$

Let the equality constraints (1) now be denoted by $Bx = e$, and let either the disaggregated (11) or aggregated (12) inequality constraints be denoted by $Ax \leqslant e$. Then we can write the *FAP* as a $\{0,1\}$ feasibility problem:

(P_L) find $x \in \{0,1\}^m$ such that $Ax \leqslant e$, $Bx = e$.

To extend the model for minimizing the number of used frequencies, we use the variables z_f (6). A number $F - M$ of frequencies to be used is determined and we add the constraint

$$\sum_{f \in \mathscr{F}} z_f = M.$$

Furthermore, the following constraints are added to the model:

$$x_{lf} + z_f \leqslant 1, \quad \forall l \in \mathscr{L},\ \forall f \in \mathscr{F}_l. \tag{13}$$

Now let $y=(x,\ z)^{\mathrm{T}}$ and let the set of linear inequalities (11) or (12) and (13) be denoted by $\bar{A}y \leqslant e$, then the new feasibility problem becomes:

$(P_{\bar{L}})$ find $y \in \{0,1\}^{m+F}$ such that $\bar{A}y \leqslant e,\ \bar{B}y=d$,

where

$$\bar{B}=\begin{pmatrix} B & 0 \\ 0 & e^{\mathrm{T}} \end{pmatrix}, \qquad d=\begin{pmatrix} e \\ M \end{pmatrix}.$$

2.3.2. A special structure of (P_L) and $(P_{\bar{L}})$

The models (P_L) and $(P_{\bar{L}})$ have the following properties:

Property 1. *All elements of A and $\bar{A}$ are binary.*

Property 2. *All elements of B and $\bar{B}$ are binary.*

Property 3. *The columns of B and $\bar{B}$ contain exactly one nonzero element.*

Property 4. *The right-hand sides of all (but one) constraints equals one.*

For the moment, let us assume that the interference constraints are modelled according to (11). Then (P_L) and $(P_{\bar{L}})$ also have the following property:

Property 5. *Two variables that occur in the same equality constraint, do not occur simultaneously* in any *inequality constraint.*

Note that, independent of the way the interference constraints are modelled, the only constraint in $(P_{\bar{L}})$ that has a right-hand side not equal to one (provided $M \neq 1$, which will usually be the case), always has Property 5. From now on, we will consider only (P_L), (P_Q) and (R_Q); for $(P_{\bar{L}})$, $(P_{\bar{Q}})$ and $(R_{\bar{Q}})$ similar theorems apply (see [19, 20]).

We can give the following expression for the matrix Q:

$$Q=\operatorname{sgn}[A^{\mathrm{T}}A-\operatorname{diag}(A^{\mathrm{T}}A)], \tag{14}$$

where $\operatorname{diag}(A^{\mathrm{T}}A)$ denotes the diagonal matrix containing the diagonal entries of the matrix $A^{\mathrm{T}}A$. Note that due to Property 5 the sgn-function is superfluous.

Lemma 7. *The matrices Q as defined in* (14) *and* (2) *are the same.*

Proof. Let the columns of A be indiced by the lf-pairs, $l \in \mathscr{L}$, $f \in \mathscr{F}_l$, and the rows by $i=1,\ldots,n$. Using the Properties 1, 4 and 5, the following equivalencies hold:

$$\begin{aligned}[A^{\mathrm{T}}A-\operatorname{diag}(A^{\mathrm{T}}A)]_{lfkg}=1 \quad &\Leftrightarrow \quad \sum_{i=1}^{n} a_{ilf}a_{ikg}=1 \\ &\Leftrightarrow \quad \exists(!)\ t\in\{1,\ldots,n\} \text{ such that } a_{tlf}=a_{tkg}=1.\end{aligned}$$

So constraint t is $x_{lf}+x_{kg} \leqslant 1$. Therefore, $f \to l$ and $g \to k$ will violate constraint t. This implies that $|f-g|<d_{lk}$. Also, by definition (2), we have that $q_{lfkg}=1$ if and only if $|f-g|<d_{lk}$. $\square$

Now we can prove the following theorems concerning (P_L) and (R_Q) (see for a proof [20]). The first is a generalization of Lemma 1; its proof uses the Properties 1–4.

Theorem 1. *The following statements hold*:
1. *If x is a feasible solution of (P_L), then x is an optimal solution of (R_Q).*
2. *If x is an optimal solution of (R_Q) with $x^{\mathrm{T}}Qx=0$, then x is either a (binary) solution of (P_L), or we can trivially construct multiple solutions of (P_L) from x.*

The second is a rephrase of Lemma 2. Again, we need the Properties 1–4 to prove it.

Theorem 2. *Given a feasible nonintegral solution $\bar{x}$ of (R_Q), one can construct a feasible integral solution $\tilde{x}$ of (P_Q), such that $\tilde{x}^{\mathrm{T}}Q\tilde{x} \leqslant \bar{x}^{\mathrm{T}}Q\bar{x}$.*

We observe that if we model the interference constraints according to (12), Lemma 7 does not hold. This is due to the fact that Property 5 does not hold for (all) equality constraints; as a consequence not all q_{lflg}, $l \in \mathscr{L}$, $f,g \in \mathscr{F}_l$, are equal to zero. However, even if $q_{lflg}=1$ for all $l \in \mathscr{L}$, $f,g \in \mathscr{F}_l$, $f \neq g$ (then $Q=\mathrm{sgn}[A^{\mathrm{T}}A+B^{\mathrm{T}}B-\mathrm{diag}(A^{\mathrm{T}}A+B^{\mathrm{T}}B)]$; note that now the function sgn is needed to ensure that Q is binary) both theorems apply. In this case, the minimizers of $x^{\mathrm{T}}Qx$ will be *strictly binary*.

The reduction of problem size in terms of the number of constraints, when comparing (P_L) and (R_Q) is enormous. The number of constraints required to model the interference constraints according to (12) (which is more concise than (11)), is approximately $|\mathscr{D}|F$. For example, in the test problems that we have considered, $|\mathscr{D}|$ is approximately 5 times the number of links, while 48 frequencies are available. So for a 200-link problem, the reduction is almost 50 000 constraints, leaving only 200 equality constraints and the bounds on the (appr. 4000) variables.

Theorems 1 and 2 also hold if the matrix Q is replaced by a nonnegative matrix $\tilde{Q}$ with the same nonzero structure. Thus they also can be applied to derive $(P_{\tilde{Q}})$. For a given (infeasible) instance of the *FAP*, we can construct the matrix A as discussed in the previous section. Subsequently, Q can be computed as discussed above. If we replace each 'one' by the penalty that has to be paid if the corresponding constraint is violated this will result in the matrix $\tilde{Q}$.

3. Solving the frequency assignment problem

The models constructed in Section 2.2 can be uniformly denoted in the following way.

$$(FAP) \quad \begin{array}{ll} \min & \frac{1}{2}x^{\mathrm{T}}Qx + v^{\mathrm{T}}x \\ \text{s.t.} & Bx = d \\ & 0 \leqslant x \leqslant e, \end{array}$$

for some indefinite symmetric matrix $Q \in \mathbb{R}^{m\times m}$, $v \in \mathbb{R}^m$, $B \in \mathbb{R}^{p\times m}$, $d \in \mathbb{R}^p$, $x \in \mathbb{R}^m$, m and p appropriate. Note that the quadratic formulation (FAP) is quite similar to existing models of other well known combinatorial optimization problems such as the max clique problem [13], the quadratic assignment problem and the graph partitioning problem [7]. The latter have been approximated using semidefinite relaxations. The quadratic formulations also have been used to derive bounds on the optimal solution, by solving their dual problems. Since our model has the advantage of a known optimal value (provided the instance under consideration is feasible; see Theorem 1) and there exists an efficient procedure for rounding fractional to binary solutions (Theorem 2), we choose to directly solve (FAP). To this end, we apply the interior point method developed by Karmarkar et al. [6, 10] to an appropriate potential function. In the following sections we discuss a suitable potential function, an interior point method to minimize the potential function, and further algorithmic details such as preprocessing techniques, starting points, ways to deal with local minima and rounding schemes.

3.1. A potential function for the FAP

To obtain a problem in which only inequality constraints occur, we relax the equality constraints $Bx = d$ to inequality constraints $Bx \geqslant d$. This is valid, since B, Q, x and d are all nonnegative. The interpretation is that we require that *at least* one frequency is assigned to each link. Note that one could also choose to deal with the equality constraints by using a projection onto the null space of B [19]. From now on, let

$$A := \begin{pmatrix} -B \\ I \\ -I \end{pmatrix}, \qquad b := \begin{pmatrix} -d \\ e \\ 0 \end{pmatrix}$$

(so this is *not* the matrix A of Section 2). A is an $n \times m$ matrix, with $n = 2m + p$. To solve (FAP), we introduce the *weighted logarithmic barrier* potential function:

$$\psi(x) = \frac{1}{2}x^{\mathrm{T}}Qx + v^{\mathrm{T}}x - \sum_{i=1}^{n} w_i \log s_i, \tag{15}$$

where w_i are positive weights, and the variables s_i are the slacks of the constraint set $Ax \leqslant b$. Instead of (FAP) we solve the equivalent nonconvex minimization problem

$$(FAP_\psi) \quad \min\ \psi(x).$$

3.2. An interior point method to minimize a nonconvex function

In this section we discuss the interior point method developed by Karmarkar et al. [6, 10]. The treatment is similar to that given in [10], although the notation used is somewhat different. Furthermore, the extension of the algorithm proposed by Shi et al. [15] is mentioned.

3.2.1. An approximate problem

Solving (FAP_ψ) is NP-complete. Therefore, in each iteration (FAP_ψ) is approximated by a quadratic optimization problem over an ellipsoid, that can be solved in polynomial time. Define

$$\mathscr{P} = \{x \in \mathbb{R}^m \mid Ax \leqslant b\};$$
$$\mathscr{P}^0 = \mathrm{Int}(\mathscr{P}) = \{x \in \mathbb{R}^m \mid Ax < b\}.$$

The algorithm starts with an initial interior point $x^0 \in \mathscr{P}^0$. It generates a sequence of points $\{x^k\}$, $k = 0, 1, \ldots$, in the interior $\mathscr{P}^0$ of the polytope $\mathscr{P}$.

Let $x^k \in \mathscr{P}^0$ be the kth iterate and let $s^k = b - Ax^k$ be its slack vector. Using the notation $S = \mathrm{diag}(s_1^k, \ldots, s_n^k)$ and $W = \mathrm{diag}(w_1, \ldots, w_n)$, the Hessian and gradient of ψ in x^k can be expressed as

$$h_\psi = Qx^k + v + A^{\mathrm{T}} S^{-1} w; \tag{16}$$
$$H_\psi = Q + A^{\mathrm{T}} S^{-1} W S^{-1} A. \tag{17}$$

Note that the density of H_ψ is determined by the density of the matrix $Q + B^{\mathrm{T}}B$. The quadratic approximation of ψ around x^k is given by

$$\mathscr{Q}(x) = \tfrac{1}{2}(x - x^k)^{\mathrm{T}} H_\psi (x - x^k) + h_\psi^{\mathrm{T}}(x - x^k) + \psi(x^k).$$

As approximation of the polytope $\mathscr{P}$, the *Dikin ellipsoid* [4] is used. The Dikin ellipsoid around $x^k \in \mathscr{P}^0$ is given by

$$\mathscr{E}(r) = \{x \in \mathbb{R}^m \mid (x - x^k)^{\mathrm{T}} A^{\mathrm{T}} S^{-2} A (x - x^k) \leqslant r^2\},$$

where for $r < 1$ the ellipsoid is inscribed in $\mathscr{P}$. Denoting $\Delta x = x - x^k$ we obtain the following optimization problem:

$$(FAP_\mathscr{E}) \qquad \begin{array}{ll} \min & \tfrac{1}{2}(\Delta x)^{\mathrm{T}} H_\psi (\Delta x) + h_\psi^{\mathrm{T}}(\Delta x) \\ \text{s.t.} & (\Delta x)^{\mathrm{T}} A^{\mathrm{T}} S^{-2} A (\Delta x) \leqslant r^2. \end{array}$$

$(FAP_\mathscr{E})$ can be solved in polynomial time (see e.g. [16, 21, 5]). It is known in the literature as the trust region subproblem. The classical algorithm has been proposed by Sörensen [16], and requires the (Cholesky) factorization of an $m \times m$ matrix. More recently, Lanczos-type algorithms have been introduced [14], that require the computation of the smallest eigenvalue of an $m \times m$ matrix. Since we are dealing with sparse matrices and we need not fix the value of r^2, we use (following Karmarkar et al. [10]) a Sörensen-type algorithm. It is explained in the next section.

3.2.2. Computing the descent direction

We start with formulating the optimality conditions of $(FAP_\mathscr{E})$ (see [5, 10, 16]). The vector Δx^* is an optimal solution of $(FAP_\mathscr{E})$ if and only if $\mu \geqslant 0$ exists, such that

$$(H_\psi + \mu A^T S^{-2} A)\Delta x^* = -h_\psi \tag{18}$$

$$\mu\,((\Delta x^*)^T A^T S^{-2} A(\Delta x^*) - r^2) = 0 \tag{19}$$

$$H_\psi + \mu A^T S^{-2} A \text{ is positive semidefinite.} \tag{20}$$

To find a solution Δx^* that satisfies the optimality conditions, one first needs to find a $\mu \geqslant 0$ such that (20) holds. Since A has full rank m, the matrix $A^T S^{-2} A$ is positive definite. Let U be its Cholesky factor, i.e. $U^T U = A^T S^{-2} A$. Ye [21] established the following lower and upper bound for μ.

Lemma 8. *Let $\lambda_{\min} < 0$ be the smallest eigenvalue of $\tilde{H} = U^{-T} H_\psi U^{-1}$. Then μ satisfies*

$$-\lambda_{\min} < \mu \leqslant \frac{\|U^{-T} h_\psi\|}{r} + m \max_{i,j} |\tilde{H}_{ij}|.$$

In the algorithm these bounds are not used explicitly. The idea is to choose a value for the multiplier μ such that (20) holds, and subsequently compute the corresponding r. From (18) it follows that

$$\Delta x^*(\mu) = -(H_\psi + \mu A^T S^{-2} A)^{-1} h_\psi. \tag{21}$$

The *S-norm* of Δx^* is defined as

$$\|\Delta x^*(\mu)\|_S = \sqrt{(\Delta x^*(\mu))^T A^T S^{-2} A(\Delta x^*(\mu))}. \tag{22}$$

The S-norm of $\Delta x(\mu)$ is a strictly decreasing convex function of μ, in the interval given in Lemma 8 [10, 21]. Since $\mu > 0$, from (19) we conclude that $\|\Delta x^*(\mu)\|_S = r$. To find a solution Δx^* that both satisfies the optimality conditions and lies on an appropriate ellipsoid, the following search strategy is employed. The linear system (18) is solved for different values of μ until a solution is obtained such that $\underline{\lambda} \leqslant \|\Delta x^*(\mu)\|_S \leqslant \overline{\lambda}$, where $\underline{\lambda}$ and $\overline{\lambda}$ are a lower and upper bound on the radius of the ellipsoid. Karmarkar et al. [10] call the interval $(\underline{\lambda}, \overline{\lambda})$ the *acceptable length region*. The procedure to compute the descent direction is given in Algorithm 3.1. Each run through this procedure is called a *minor* iteration. Some explanation is given below.

The procedure needs as input the current iterate x^k, an initial multiplier μ and the acceptable length region $(\underline{\lambda}, \overline{\lambda})$. Three logical keys are set:

- ID_{key}, which is true if during the process an indefinite matrix is encountered,
- $\overline{\mu}_{\text{key}}$ ($\underline{\mu}_{\text{key}}$), which is true if an upper (lower) bound is found for the multiplier μ. An upper (lower) bound is found if for a given μ the corresponding S-norm is too small (too large).

The procedure runs as follows. If necessary μ is increased until $H_\psi + \mu A^T S^{-2} A$ is positive definite. Subsequently the descent direction and its S-norm are computed.

- If the S-norm is too small, and ID_{key} is *false*, an upper bound on μ has been found. μ is decreased, either by multiplying it by $\|\Delta x^*\|_S$ or if a lower bound on μ exists, by taking the geometrical

Algorithm 3.1. The descent_direction algorithm

Procedure descent_direction $(\mu, x^k, \underline{\lambda}, \overline{\lambda})$

$\|\Delta x^*\|_S := 0$; ID_{key} := **false**; $\overline{\mu}_{\text{key}}$:= **false**; $\underline{\mu}_{\text{key}}$:= **false**;
while $(\|\Delta x^*\|_S > \overline{\lambda}$ **or** $(\|\Delta x^*\|_S < \underline{\lambda}$ **and** ID_{key} =**false**)) **do**
while $H_\psi + \mu A^{\mathrm{T}} S^{-2} A$ is not PSD **do**
    $\mu := \mu\sqrt{2}$;
    ID_{key} := **true**;
  endwhile
  $\Delta x^* := -(H_\psi + \mu A^{\mathrm{T}} S^{-2} A)^{-1} h_\psi$;
  $\|\Delta x^*\|_S := \sqrt{(\Delta x^*)^{\mathrm{T}} A^{\mathrm{T}} S^{-2} A (\Delta x^*)}$;
  if $(\|\Delta x^*\|_S < \underline{\lambda}$ **and** ID_{key}= **false**) **then**
    $\overline{\mu} := \mu$; $\overline{\mu}_{\text{key}}$:= **true**;
    if $\underline{\mu}_{\text{key}}$ = **true then** $\mu := \sqrt{\underline{\mu} \cdot \overline{\mu}}$ **else** $\mu := \mu\|\Delta x^*\|_S$;
  endif
  if $(\|\Delta x^*\|_S > \overline{\lambda})$ **then**
    $\underline{\mu} := \mu$; $\underline{\mu}_{\text{key}}$:= **true**;
    if $\overline{\mu}_{\text{key}}$= **true then** $\mu := \sqrt{\underline{\mu} \cdot \overline{\mu}}$ **else** $\mu := \mu\|\Delta x^*\|_S$;
    if ID_{key} = **false then** ID_{key} := **true**;
  endif
endwhile
if $(\|\Delta x^*\|_S < \underline{\lambda}$ **and** ID_{key}= **true**) **then** $\underline{\lambda} := \gamma\underline{\lambda}$;
if $(\|\Delta x^*\|_S < \frac{\underline{\lambda}+\overline{\lambda}}{2}$ **and** ID_{key} = **false**) **then** $\mu := \mu\|\Delta x^*\|_S$;
return $(\Delta x^*, \mu, \underline{\lambda}, \overline{\lambda})$

---

mean of μ and the lower bound; with the new value for μ, the new descent direction and S-norm are computed.

- If the S-norm is too large, a lower bound on μ has been found. μ is increased, either by multiplying it by $\|\Delta x^*\|_S$ or if an upper bound on μ exists, by taking the geometrical mean of μ and the upper bound; with the ne value for μ, the new descent direction and S-norm are computed.
- If the S-norm is too, small and ID_{key} is *true*, decreasing μ will lead to an indefinite matrix; the acceptable length region is adjusted, i.e. $\underline{\lambda} := \gamma\underline{\lambda}$, $\gamma < 1$, and the descent direction is accepted.
- If the S-norm is satisfactory the descent direction is accepted. If the S-norm is below a certain bound, for example $(\underline{\lambda} + \overline{\lambda})/2$, the current multiplier μ is multiplied by the S-norm. The final multiplier of this iteration is the initial multiplier of the next iteration.

Karmarkar et al. [6, 10] choose, when the S-norm is too large (too small), to multiply (divide) μ by a constant $\mu_r = \sqrt{2}$. During our experiments we observed that in each iteration the product $\mu\|\Delta x^*\|_S$ is more or less constant for each μ. Therefore we choose to modify μ as described above.

After a descent direction Δx^* has been found, a line search is applied to find the minimum of the potential function along the line $\{\alpha \in \mathbb{R}^+ \mid x^k + \alpha\Delta x^*\}$. Shi et al. [15] use a *golden section search*. The new iterate x^{k+1} is computed by

$$x^{k+1} = x^k + \alpha_{\text{opt}} \Delta x^*.$$

If the potential value does not improve, i.e. $\psi(x^{k+1}) \geqslant \psi(x^k)$, the acceptable length region and hence the radius of the ellipsoid are decreased. It can be shown [10] that always a descent direction can be found if the radius of the ellipsoid is sufficiently decreased. In our implementation, we consider x^{k+1} to be a *local minimum* if the potential value does not sufficiently improve, i.e. $\psi(x^{k+1}) \geqslant \psi(x^k) - \varepsilon_1$.

3.3. Further algorithmic details

3.3.1. Preprocessing

Since the test problems are of considerable size, methods to reduce the size of the problem may be used, thus decreasing computational effort. Some straightforward preprocessing methods, which can be used separately or in combination, are discussed below.

3.3.1.1. Removing variables from the problem. In some cases, it is easily seen that a variable included in the problem is redundant, and may be removed. For instance, given a frequency domain consisting of frequencies in the range [10, 100], and for a given pair of links the minimal required frequency distance is, say, 70, then frequencies in the range [30, 80] may not be assigned to either of these links, and the variables corresponding to these assignments may be removed. So, if for a given frequency–link pair the following holds:

$$\text{given } l \in \mathscr{L},\ f \in \mathscr{F}_l : \exists k \in \mathscr{L} \text{ such that } \forall g \in \mathscr{F}_k : |f-g| < d_{lk} (\Leftrightarrow q_{lfkg} = 1),$$

then the variable x_{lf} may be removed from the problem. Removing a variable x_{lf} from the problem and updating its frequency domain may lead to other variables becoming redundant. We remove variables from the problem until no redundant variables remain. Note that this method may identify infeasibility of the problem under consideration. This occurs if for a link all variables are removed. Furthermore, after preprocessing, some links may have only one available frequency. Then the corresponding assignment can be fixed, and all variables corresponding to interfering assignments may be removed from the problem. This, in turn, may lead to some variables becoming redundant or to more fixed assignments. The process is repeated until no further reduction is possible.

The infeasible instances of the *FAP* also can be preprocessed based on this idea. A variable x_{lf} is removed if there exists a $k \in \mathscr{L}$ such that for all $g \in \mathscr{F}_k$, $q_{lfkg} > C_{\max}$, where $C_{\max}$ is the maximum cost we allow for any pair of assignments. Removing variables in this way, may result in increasing the optimal solution value of the problem, since usually nothing is known about the optimal solution. Note that if an upper bound *UB* is known, taking $C_{\max} = UB$ obviously does not lead to changing the optimal value.

3.3.1.2. Removing frequencies from the problem. This strategy is only valid for the feasible instances of the *FAP*. It is attempted to find a feasible assignment for the problem using a straightforward heuristic method, and subsequently the potential reduction method is used to find a better solution, using only the frequencies that are required in the initial assignment generated by the heuristic method. A heuristic method that appeared to be quite successful in finding reasonably good feasible solutions for the given test problems, is the following. The idea is to select the link that has the lowest number of frequencies available, and assign to it the frequency that has the highest number of links to which it may be assigned. This frequency is assigned as many times as possible, before another frequency is chosen to be assigned, according to the same rule.

We introduce some more notation:

$$\begin{aligned}&\mathscr{L}^0\text{: the set of links to which no frequency has been assigned;}\\&\mathscr{F}_l^-\text{: the set of available frequencies for link } l\text{;}\\&\mathscr{F}^0\text{: the set of unused frequencies.}\end{aligned} \tag{23}$$

The heuristic is formalized below.

$\mathscr{A}_f := \{l \in \mathscr{L} \mid f \in \mathscr{F}_l^-\},\ \forall f \in \mathscr{F}.$

$l^* := \operatorname{argmin}_{l \in \mathscr{L}^0} |\mathscr{F}_l^-|.$

while $\mathscr{F}_{l^*}^- \neq \emptyset$

  $f^* := \operatorname{argmax}_{f \in \mathscr{F}_{l^*}^-} |\mathscr{A}_f|.$

  while $\mathscr{F}_{l^*}^- \neq \emptyset$

    $f^* \Rightarrow l^*.$

    Update $\mathscr{A}_f,\ \forall f \in \mathscr{F}^0;\ \mathscr{L}^0$ and $\mathscr{F}_l^-,\ \forall l \in \mathscr{L}^0.$

    $l^* = \operatorname{argmin}_{l \in \mathscr{A}_{f^*}} |\mathscr{F}_l^-|.$

  endwhile

  $l^* = \operatorname{argmin}_{l \in \mathscr{L}^0} |\mathscr{F}_l^-|.$

endwhile

The heuristic either finds a feasible assignment, or it stops when it detects an infeasibility caused by the partial assignment it has generated. If a full assignment is found, the preprocessed problem is obtained by removing all frequencies that are not used in this assignment. Note that the resulting problem may have a worse optimal assignment than the original, but is clearly feasible.

3.3.2. Starting points

Since we choose to relax the equality constraints to inequality constraints, we need to find an interior starting point x^0 satisfying $0 < x^0 < e,\ Bx^0 > d$. We simply take (assuming $|\mathscr{F}_l| \geqslant 3,\ \forall l \in \mathscr{L}$)

$$x_{lf}^0 = \frac{1}{|\mathscr{F}_l| - 1}, \quad \forall l \in \mathscr{L},\ \forall f \in \mathscr{F}_l.$$

If required, for the variables z_f we take

$$z_f^0 = \frac{M}{F-1}, \quad \forall f \in \mathscr{F},$$

where $F - M$ is (again) an upper bound on the number of frequencies to be used.

3.3.3. Local minima

If the algorithm ends up in a local minimum, some measures have to be taken to prevent the algorithm from running into the same local minimum again after the process has been restarted. To accomplish this, we use the following combination of methods. Let x_{lm} denote the final interior solution, yielding a local minimum, and let x^0 be the starting point.

- *Change the weights* in the barrier of the potential function, according to the following rule: determine all constraints i which are 'near-active' for the final interior solution x_{lm}, i.e. $s_i < \varepsilon_2$, where $\varepsilon_2 > 0$ is small. Increase the corresponding weights w_i.
- *Add a cut* (see also Karmarkar et al. [10]). Let $\tilde{x}$ be the final rounded solution. Suppose $\tilde{x}$ is infeasible. Let $\mathscr{I} = \{lf \mid \tilde{x}_{lf} = 1\}$, then the cut

  $$\sum_{lf \in \mathscr{I}} x_{lf} \leqslant |\mathscr{I}| - 1$$

  does not cut off any feasible solution. Of course, any infeasible rounded solution may be used to generate a cut.
- *Restart the process* from a new starting point, taking the new starting point x^0_{new} as:

  $$x^0_{\text{new}} = x_{lm} + \alpha_{\max}(x^0 - x_{lm}),$$

  where $\alpha_{\max} = \max\{\alpha \in \mathbb{R}^+ \mid x_{lm} + \alpha\,(x^0 - x_{lm}) \in \mathscr{P}^0\}$. Clearly, if $\alpha = 1$ then $x^0_{\text{new}} \equiv x^0$, so we are looking for an $\alpha > 1$.

3.3.4. Rounding schemes

In each iteration, the new iterate x^{k+1} is rounded to one or more binary solutions. A number of rounding schemes has been developed and tested. Here we only describe the rounding schemes that were the most successful in finding feasible assignments, according to our experiments. Table 1 gives an indication which rounding schemes are applicable to which classes of problems. In this section we also use the notation (23).

3.3.4.1. Rounding scheme I. This is a straightforward rounding scheme: to each link, assign the frequency that has the largest x_{lf}-value. It yields a full (not necessarily feasible) assignment.

Given a fractional solution x:
$f_l^* := \operatorname{argmax}_{f \in \mathscr{F}_l} x_{lf}, \quad \forall l \in \mathscr{L}.$
$f_l^* \to l, \quad \forall l \in \mathscr{L}.$

Table 1
Applicability of rounding schemes

	Feas. Prob.	Min. # Frqs.	Min. Cost
I	Y	Y	Y
II	Y	Y	Y
IIIa	Y	Y	Y
IIIb	N	Y	N
IVa	Y	N	N
IVb	N	Y	N
Va	N	N	Y
Vb	N	N	Y

3.3.4.2. Rounding scheme II. The idea of this rounding scheme is to first make the assignment corresponding to the largest x_{lf} value which has not been rounded yet, then determine which assignments are not allowed as a consequence of this assignment, and subsequently make the next assignment. It terminates either when a full, feasible assignment has been found, or when a partial assignment has been generated that cannot be extended without violating constraints.

Given a fractional solution x:
while any x_{lf} is not rounded
  $(l^*, f^*) := \underset{l \in \mathscr{L}^0,\ f \in \mathscr{F}_l^-}{\operatorname{argmax}} \ x_{lf}.$
  $f^* \Rightarrow l^*.$
endwhile

3.3.4.3. Rounding schemes IIIa and IIIb. These rounding schemes assign a frequency as often as possible, before moving on to assigning other frequencies, terminating when a full, feasible assignment has been found, or when a partial assignment has been found that cannot be extended without violating constraints. The difference between the rounding schemes is the manner in which the frequency to assign is selected. IIIa selects a frequency according to the values of the x-variables, IIIb evaluates the values of the z-variables. Rounding scheme IIIa is formalized.

Given a fractional solution x:
while any x_{lf} not rounded
  $f^* := \underset{l \in \mathscr{L}^0,\ f \in \mathscr{F}_l^-}{\operatorname{argmax}} \ x_{lf}.$
  $\Theta := \{l \in \mathscr{L} \mid x_{lf^*} \text{ unrounded}\}.$
  while $\Theta \neq \emptyset$
    $l^* := \underset{l \in \Theta}{\operatorname{argmax}}\, x_{lf^*}.$
    $f^* \Rightarrow l^*.$
    Update Θ.
  endwhile
endwhile

3.3.4.4. Rounding schemes IVa and IVb. In these rounding schemes the link that has the least number of frequencies available is selected and assigned the frequency corresponding to its largest x_{lf} value. The rounding schemes terminate when either a full, feasible assignment has been found, or when a link has no available frequencies left. Rounding scheme IVa uses initially the full set of available frequencies for each link. In rounding scheme IVb we first construct a subset Φ of $\mathscr{F}$ containing $F - M$ frequencies, with $F - M$ the number of frequencies to be used, which can be selected according to either x or z values. In the rounding procedure only frequencies in Φ are used. We summarize rounding scheme IVa:

Given a fractional solution x:
while any x_{lf} not rounded and $\mathscr{F}_l^- \neq \emptyset,\ \forall l \in \mathscr{L}^0$
$\quad l^* := \operatorname{argmin}_{l \in \mathscr{L}} |\mathscr{F}_l^-|.$
$\quad f^* := \operatorname{argmax}_{f \in \mathscr{F}_{l^*}^-} x_{l^* f}.$
$\quad f^* \Rightarrow l^*.$
endwhile

3.3.4.5. Rounding schemes Va and Vb. These rounding schemes have been developed for the infeasible instances of the *FAP*; given a partial assignment, the cost of all possibilities of extending the assignment are determined. Subsequently, for each link the assignment corresponding to the lowest cost is determined, and of these the assignment corresponding to the highest cost is selected, until a full assignment has been generated. This is a maximin-strategy; rounding scheme Vb uses a minimin-strategy. To obtain the partial assignment the algorithm requires to start with, slightly modified versions of rounding schemes II or IIIa may be used. For instance, consider rounding scheme II. We use a slightly different notation:

$f \Rightarrow l : f \rightarrow l$ and the exclusion of all interfering assignments
with a penalty higher than $\mathscr{K}$,

where $\mathscr{K}$ is some maximum cost that is allowed for the next assignment. Va can be summarized as follows:

Given a partial assignment x (i.e. $x_i \in \{0, 1\}$):
$\Lambda := \{l \in \mathscr{L} \mid$ no frequency is assigned to $l\}$.
while $\Lambda \neq \emptyset$
$\quad \kappa_{lf} :=$ cost of $f \rightarrow l,\ \forall\, l \in \Lambda,\ \forall f \in \mathscr{F}_l.$
$\quad f_l^* := \operatorname{argmin}_{f \in \mathscr{F}_l} \kappa_{lf},\ \forall l \in \Lambda.$
$\quad l^* := \operatorname{argmax}_{l \in \Lambda} \kappa_{l f_l^*}.$
$\quad f_{l^*}^* \rightarrow l^*.$
$\quad$ Update Λ.
endwhile

Algorithm 3.2. The main algorithm

Procedure solve_FAP (Q, A, b)

$k := 0$; $\mu := \mu_0$; $w := w_0$; $\underline{\lambda} := \underline{\lambda}_0$; $\overline{\lambda} := \overline{\lambda}_0$; $K := 0$;
$(Q, A, b) := preprocess\ (Q, A, b)$ (*)
$x^k :=$ *get_start_point* (A, b);
$\tilde{x}^k :=$ *round_off* (x^k); (*)
while not STOP do
  $(\Delta x^*, \mu, \underline{\lambda}, \overline{\lambda}) :=$ *descent_direction* $(\mu, x^k, \underline{\lambda}, \overline{\lambda})$;
  $\alpha_{opt} :=$ *golden_section_ step* $(A, b, x^k, \Delta x^*)$;
  $x^{k+1} := x^k + \alpha_{opt}\Delta x^*$;
  if $\psi(x^{k+1}) < \psi(x^k) - \varepsilon_1$ **then**
    $\tilde{x}^{k+1} :=$ *round_off* (x^{k+1}); (*)
    $w := \beta w$;
    $k := k + 1$;
  else
    $(A, b, x^0, w) :=$ *local_min* (A, b, x^{k+1}, x^0, w);
    $k := 0$; $\mu := \mu_0$; $\underline{\lambda} := \underline{\lambda}_0$; $\overline{\lambda} := \overline{\lambda}_0$; $K := K + 1$;
    $\tilde{x}^k :=$ *round_off* (x^k); (*)
  endif
  STOP := *evaluate_stopping_criterium*;
endwhile

3.4. Summary of the algorithm

Now we are ready to summarize the complete algorithm; see Algorithm 3.2. We give some explanation below.

The procedure needs initial values for the multiplier, the weight vector and the acceptable length region. k is a counter for the number of minor iterations, K counts the number of *major* iterations. The marked steps (*) are optional. The boolean variable STOP represents some stopping criterium, which is evaluated in each (minor) iteration. Possible stopping criteria are, for example:

- A feasible solution has been found.
- A feasible solution that uses no more than $F - M$ frequencies has been found.
- A global minimum of x^TQx has been found (without using rounding schemes).
- A solution with 'acceptable' cost has been found.
- The number of major iterations exceeds some maximum K_{max}.

The procedures `preprocess`, `descent_direction`, `get_start_point`, `local_min` and `round_off` are discussed in the Sections 3.3.1, 3.2.2, 3.3.2, 3.3.3 and 3.3.4. Each time an improved value for the potential function has been found, the weights are decreased by a constant factor $\beta < 1$, to ensure the minimum of the potential function will be equal to zero.

4. Computational results

In this section we report on computational results on a number of test problems, both real-life and randomly generated. The CELAR problems were made available as part of the international CALMA project, which is part of the EUCLID program of the departments of defense of the United Kingdom, France and the Netherlands. The goal of this project was to develop and test algorithms for a variant of the *FAP*: the Radio Link Frequency Assignment Problem (*RLFAP*). Several algorithmic approaches have been taken to solve the *RLFAP*; see for an overview [18].[3] Both the CELAR and the randomly generated test instances have a specific structure which is explained in the next subsection. Further on in this section some implementational issues will be discussed.

4.1. A special structure of the CELAR problems

In the model constructed in Section 2 it is assumed that no equality interference constraints exist. In the CELAR data sets however, equality constraints do occur. By using a special structure of the CELAR problems, the equality constraints can be eliminated. Let us introduce some extra notation:

$\mathscr{D}^{\text{eq}}$: set of pairs of links for which an equality constraint must be satisfied;

d_{lk}^{eq}: required frequency distance between the links $(l,k)\in\mathscr{D}^{\text{eq}}$.

On examination of the CELAR data sets the following structure concerning the equality constraints becomes clear:

$$\mathscr{D}^{\text{eq}} = \{(1,2),(3,4),\ldots,(L-1,L)\};$$
$$d_{12}^{\text{eq}} = d_{34}^{\text{eq}} = \cdots = d_{L-1,L}^{\text{eq}} = 238;$$
$$p_{12} = p_{34} = \cdots = p_{L-1,L} = 0.$$

Or, in words, for each link there exists *exactly one* equality constraint, which has to be satisfied. Furthermore, the frequency domains have the following structure:

$$\forall f\in\mathscr{F} : \exists! f^*\in\mathscr{F} : |f^* - f| = 238;$$
$$\mathscr{F}_l = \mathscr{F}_{l+1},\ \forall l\in\mathscr{L},\ l \text{ odd}.$$

So, each frequency has exactly one *complementary* frequency. Therefore we can define

$$\mathscr{L}^* = \{l\in\mathscr{L} \mid l \text{ is odd}\}.$$

We define the following binary decision variables:

$$x_{lf} = \begin{cases} 1 & \text{if } f\rightarrow l, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall l\in\mathscr{L}^*,\ \forall f\in\mathscr{F}_l.$$

Obviously, it is sufficient to define x_{lf} only for all $l\in\mathscr{L}^*$, because assigning f to l automatically implies assigning the complementary frequency of f to $l+1$.

[3] For more information on the CALMA project, the interested reader is referred to the World Wide Web page http://www.win.tue.nl/win/mat/bs/comb_opt/hurkens/calma.html. Most test problems appearing in this section (and technical reports) are obtainable from ftp://ftp.win.tue.nl/pub/techreports/CALMA.

This yields a problem size reduction by a factor two. Consequently, we may assume that no equality constraints occur in the *RLFAP*; the equality constraints in $\mathscr{D}^{eq}$ will be modelled implicitly. The *GRAPH* test problems [2] are patterned after the CELAR problems and exhibit the same structure.

Finally, we mention that all test instances make use of a frequency domain that consists of 48 distinct frequencies, and for each link a subset of these 48 frequencies is available. Therefore, an upper bound on the number of variables required to model an L-link *FAP* is $\frac{1}{2}LF = 24L$.

4.2. Implementation

The algorithm is implemented in MATLAB™. A number of FORTRAN routines, provided by the linear programming interior point solver LIPSOL [22], are incorporated. These use sparse matrix techniques to do the minimum degree ordering, symbolic factorization, Cholesky factorization and back substitution to solve the linear system (18). The tests were run on a HP9000/720 workstation, 144 Mb memory, 50 mHz.

In the implementation, the following parameter settings are used:

- The initial acceptable length region $(\underline{\lambda}, \bar{\lambda}) = (0.5, 1.0)$.
- The reduction factor of the acceptable length region $\gamma = 0.25$.
- The tolerance $\varepsilon_1 = 10^{-3}$.
- The weights w_i are initially set to $100/n$; they are multiplied by $\beta = \frac{1}{2}$ in each iteration. After a restart, the weights are reset to their initial value, and a number of weights are increased by a factor 8, according to the rule described in Section 3.3.3, with $\varepsilon_2 = 10^{-2}$.
- The initial multiplier μ^0 is set to $\zeta(n/100)\max_i(w_i)\max_{ij}(Q_{ij})$, where ζ is a constant that is taken equal to $\frac{1}{2}$, 2, 1/10 for solving the problems (R_Q), $(R_{\bar{Q}})$, $(R_{\tilde{Q}})$.
- The maximal number of major iterations $K_{\max}$ is set to 5 for the smaller (<1200 variables) problems, and to 2 for the larger problems.

4.3. Results on feasible FAPs

In this section the results of applying the algorithm to a number of feasible test problems are given. All problems except the CELAR problems were generated using *GRAPH* [2].

4.3.1. Preprocessing the feasible FAPs

Most problems can be reduced by applying the preprocessing technique of Section 3.3.1.1. Table 2 shows the results obtained using this preprocessing method. It appears that this preprocessing method is a powerful tool to reduce the size of the problems; three problems are even completely solved, as it appears that only *one* feasible assignment exists for these problems. Obviously, this method is effective only if there exist some reasonably large interference constraints. Unfortunately, not in all problems such constraints exist; these problems do not reduce in size.

Table 2
Preprocessing results using preprocessing method 1

Problem description			min.	Reduction # variables		
Name	L	$\|\mathscr{D}\|$	# frq.	Before	After	Perc.
GR10.4	10	8	4	240	144	41%
GR10.6	10	16	6	240	158	35%
GR10.8	10	32	8	178	75	60%
GR16.6	16	28	6	384	208	47%
GR20.6	20	36	6	480	387	20%
GR20.10	20	80	10	480	272	44%
GR24.6	24	74	6	494	494	0%
GR26.8	26	72	8	624	363	43%
GR30.8	30	84	8	720	538	26%
GR36.12	36	204	12	526	506	4%
GR40.8	40	170	8	670	627	7%
GR40.10	40	124	10	786	20	**100%**
GR40.18	40	212	18	782	239	71%
GR50.12	50	240	12	1200	568	58%
GR60.12	60	296	12	948	30	**100%**
GR76.10	76	392	10	1544	512	69%
GR100.12	100	484	12	2400	1156	53%
TUD200.1	200	1071	12	4064	3991	2%
TUD200.2	200	1043	46	3840	100	**100%**
TUD200.3	200	1060	$\leqslant 14$	3620	3425	6%
TUD200.4	200	1042	20	3456	302	94%
CELAR04	680	3627	46	13 428	740	96%

Note: Given are, for each test instance, the numbers of links and interference constraints, the minimal number of frequencies required and the number of variables before and after preprocessing. The reduction percentage can be computed by noting that for a feasible L-link problem, at least $\frac{1}{2}L$ variables must remain.

The method described in Section 3.3.1.2 was developed to reduce the size of the larger ($\geqslant 200$ links) problems. Table 3 shows the results.

4.3.2. Computational results on feasible FAPs

The algorithm was applied to all the preprocessed problems of the previous section, using model (R_Q). After each iteration, the rounding schemes I, II, IIIa and IVa all were applied, because earlier experiences with the algorithm indicated that, generally speaking, it is hard to predict beforehand which of these rounding schemes will give the best results. In our implementation, the amount of time needed to round the solutions and solve the linear systems is comparable. In an efficient low level language implementation, solving the linear systems will dominate the computation times. A number of problems were solved using both the full and preprocessed version. In Table 4 the best solutions are shown. For all problems feasible assignments were found, and in most cases the assignments found are optimal or near-optimal. During the experiments, we observed that for the majority of problems feasible (but not necessarily optimal) assignments were generated in the first few iterations (this is not indicated in the table). Globally speaking, rounding scheme IIIa was the most successful

Table 3
Preprocessing results using preprocessing method 2

Problem description			min.	Preprocessing		Reduction # variables		
Name	L	$\|\mathscr{D}\|$	# frq.	# frq.	Time	Before	After	Perc.
TUD200.1	200	1071	12	20	6.2	3991	1740	55%
TUD200.3	200	1060	≤14	26	6.4	3425	1483	55%
GRAPH01	200	1034	18	22	6.7	3460	1702	51%
GRAPH02	400	2045	14	18	24.2	7312	3078	58%
GRAPH14	916	4180	8	12	127	18 358	5062	73%
CELAR01	916	5090	16	20	118	18 100	7734	59%
CELAR02	200	1135	14	16	9.2	4002	1506	63%
CELAR03	400	2560	14	18	26.7	7946	3180	61%

Note: Given are, for each test instance, the numbers of links and interference constraints, the minimal number of frequencies required, the quality of the solution and the time required for preprocessing, the number of variables before and after preprocessing, and the reduction in percentages.

in finding optimal assignments, rounding scheme II generated feasible assignments quickly, rounding scheme IVa worked when the others failed (especially for the largest problems) and rounding scheme I found feasible assignments when the iterates were close to a global minimum.

Furthermore, for most problems a global minimum which yields in all cases more than one and for the larger problems an enormous number of feasible assignments was found. Unfortunately, though not surprisingly, in most cases the best assignments that can be constructed from a fractional global minimizer are nonoptimal.

Therefore, we also solved most of the above problems using model ($R_{\overline{Q}}$) (Table 5). The number of frequencies to be used $F - M$ was set equal to the optimal or best known solution. We used the rounding schemes I–IV. Using this model, for a number of problems a better solution was found (than previously). However, it is much more rare that a global minimum was found. This only happened for a few (relatively small) instances. In some cases a local minimum was found in which only the constraint on the number of frequencies to be used was violated, i.e. for the final solution $y = (x, z)^{\mathrm{T}}$ we found that $x^{\mathrm{T}}Qx = 0$ while $z^{\mathrm{T}}Rx > 0$. Still, the assignments induced by x were close to optimal, and are therefore included in the table.

Generally, if no global minimum was found, the local minima that were found had a value close to zero, usually in the range of 1–8. So, the corresponding assignments violate only a very small number of interference constraints. By applying some local search technique to these infeasible assignments, feasible assignments can be generated, probably in only a few steps.

4.4. *Results on infeasible FAPs*

Finally, a number of infeasible instances were solved. For these instances $p^{\min} = mob^{\max} = 4$ and the associated penalties τ_i resp. θ_i are equal to 10^{4-i}; except for CELAR10 where $\theta_i = 10^{6-i}$. A priority of 0 occurs only for equality constraints.

Table 4
Computational results using model (R_Q)

	Best assignment			Final feasible assignments			
Problem	# frq.	Time	# iter.	# ass.	Best	Time	# iter.
GR10.4	**4**	0.2	0	36 000	**4**	22	25
GR10.6	**6**	0.2	0	1170	**6**	37	29
GR10.8	**8**	0.2	0	15	**8**	6	16
GR16.6	**6**	4	2	7650	**6**	29	22
GR20.6	**6**	0.4	0	7.1×10^{6}	10	49	18
GR20.10	**10**	2.8	1	1200	16	36	19
GR24.6	**6**	0.5	0	9.9×10^{6}	8	69	19
GR26.8	**8**	22	9	9.0×10^{6}	10	33	17
GR30.8	10	35	7	2.1×10^{5}	12	90	22
GR36.12	14	49	10	82944	14	75	18
GR40.8	10	0.8	0	7.8×10^{9}	10	80	19
GR40.18	**18**	8	5	1.8×10^{8}	20	13	15
GR50.12	**12**	325	80	Local minima		—	—
GR76.10	16	36	12	5.4×10^{10}	22	37	19
GR100.12	14	69	7	1.3×10^{10}	20	336	43
TUD200.1	18	3602	18	1.8×10^{35}	32	5215	26
TUD200.1**pp**	14	348	16	3.6×10^{21}	20	342	22
TUD200.3	32	2926	29	2.9×10^{34}	32	2575	29
TUD200.3**pp**	20	2395	32	Local minima		—	—
TUD200.4	**20**	8	4	8.8×10^{12}	**20**	14	14
GRAPH01	22	568	4	2.3×10^{40}	32	2776	23
GRAPH01**pp**	**18**	397	19	1.8×10^{28}	22	1426	101
GRAPH02**pp**	**14**	286	4	9.0×10^{55}	18	1199	29
GRAPH14**pp**	**10**	3806	11	Local minima		—	—
CELAR01**pp**	18	2863	8	Local minima		—	—
CELAR02	16	164	3	9.6×10^{45}	28	2238	54
CELAR02**pp**	**14**	102	7	1.5×10^{32}	16	350	50
CELAR03	20	248	1	9.1×10^{91}	32	7973	59
CELAR03**pp**	16	183	5	Local minima		—	—
CELAR04**pp**	**46**	5	**0**	1.7×10^{66}	**46**	83	12

Note: Given are times and numbers of minor iterations required to find the best (**optimal**) feasible assignment and a global minimum (if any). The times are respectively with and without time for rounding the fractional solutions incorporated. Also, the number of feasible assignments and the best assignment that can be constructed from the (fractional) global minimizer are given.

Table 6 gives the preprocessing results for a number of problems. The maximum cost for any assignment ($C_{\max}$) was set to 50 (see Section 3.3.1.1). In Table 7 the computational results are shown. The rounding schemes I and Va and Vb were applied. The partial assignment needed by the rounding schemes V was generated with rounding scheme II, with $\mathscr{K}=50$ (see Section 3.3.4). We observe that only for the smallest instance an optimal solution was found, by rounding scheme Va. In most other cases the best found solution was generated by rounding scheme I and equal to the value of the final solution, except for CELAR10; this solution was obtained by rounding scheme Va. Again, large numbers of equivalent assignments were found simultaneously.

Table 5
Computational results using model $(R_{\overline{Q}})$

Name	Best assignment # frq.	Time	# iter.	Final feasible assignments # ass.	# frq.	Time	# iter.
GR10.4	**4**	0.2	0	8	**4**	23	27
GR10.6	**6**	0.2	0	8	**6**	36	31
GR10.8	**8**	0.2	0	2	**8**	57	130
GR16.6	**6**	13	8	4	**6**	35	28
GR20.6	**6**	0.4	0	Local minima		—	—
GR20.10	**10**	2.9	1	Local minima		—	—
GR24.6	**6**	0.5	0	4096	**6**	133	28
GR26.8	**8**	18	6	Local minima		—	—
GR30.8	**8**	6	1	Local minima		—	—
GR36.12	**12**	78	15	64	**12**	193	39
GR40.8	**8**	850	145	4	**8**	729	151
GR40.18	**18**	61	22	Local minima		—	—
GR76.10	14	180	46	Local minima		—	—
GR100.12	18	684	59	Local minima		—	—
TUD200.1**pp**	16	2611	114	Local minima		—	—
TUD200.3	26	6632	60	Local minima		—	—
TUD200.3**pp**	18	2434	155	Local minima		—	—
GRAPH01	20	237	1	4.7×10^{39}	20	3212	29
GRAPH01**pp**	**18**	1397	73	Local minima		—	—
GRAPH02**pp**	**14**	2355	39	Local minima		—	—
CELAR01**pp**	**16**	544	12	Local minima		—	—
CELAR02	**14**	2355	39	4.0×10^{39}	16	2545	60
CELAR02**pp**	**14**	13	1	1.2×10^{28}	16	380	52
CELAR03	18	6618	31	8.1×10^{61}	20	11 225	75
CELAR03**pp**	16	3374	79	5.7×10^{62}	18	1524	110

Note: Given are times and numbers of minor iterations required to find the best (**optimal**) feasible assignment and a global minimum (if any). The times are respectively with and without time for rounding the fractional solutions incorporated. Also, the number of feasible assignments (if any) that can be constructed from the final (fractional) minimizer and the number of frequencies they use are given.

Table 6
Preprocessing results on infeasible *FAP*s

Problem description Name	L	$\vert\mathscr{D}\vert$	min. cost	Reduction # variables Before	After	Perc.
GR12.6	12	32	100	214	—	—
GR40.6	40	124	2130	704	636	10%
CELAR06	200	1222	3437	4010	2164	47%
CELAR09	680	3763	15 665	13 428	5550	60%
CELAR10	680	3763	32 456	13 428	3450	76%

Note: Given are, for each test instance, the problem size, the optimal cost and the reduction of the number of variables both absolute and in percentages. CELAR09(10) was preprocessed by fixing assignments for links with a mobility lower than 2(3).

Table 7
Computational results using model $(R_{\tilde{Q}})$

	Best assignment			Final assignments			
Problem	Cost	Time	# iter.	# ass.	Cost	Time	# iter.
GR12.6	**100**	8	3	288	103	50	25
GR40.6	3604	648	69	663 552	3604	637	78
GR40.6**pp**	2776	263	29	84 672	2776	293	38
CELAR06**pp**	4539	2620	90	$7.3 * 10^{23}$	4539	2176	95
CELAR09**pp**	15 775	1756	30	$8.1 * 10^{15}$	15 775	1905	43
CELAR10**pp**	32 460	50	1	$3.4 * 10^{15}$	32 474	582	27

Note: Given are times and numbers of minor iterations required to find the best (**optimal**) assignment and a local minimum. The times are respectively with and without time for rounding the fractional solutions incorporated. Also, the number of assignments that can be constructed from the (fractional) local minimizer, and their cost, are given.

Remark. *The matrices $\tilde{Q}$ are generally more dense than the matrices Q and $\bar{Q}$; this is due to the fact that the required frequency distances in most infeasible instances of the (RL)FAP are larger than for the feasible instances. Therefore, solving the linear systems requires more time.*

5. Concluding remarks

In this paper an interior point potential reduction algorithm for the Frequency Assignment Problem has been developed.

- A quadratic formulation of the *FAP* has been developed, which results in a compact and computationally attractive problem formulation. A further advantage of the quadratic formulation is, that it provides a uniform model for the various objectives of the *FAP* and that by optimizing it multiple solutions may be found.
- By applying an interior point algorithm to the potential function for solving the quadratic model, problems up to a size of 8000 variables have been solved within reasonable time. The assignments obtained are generally speaking fairly good; for most feasible instances optimal or near optimal assignments have been found, while for the infeasible instances the obtained assignments are within reasonable distance of the best known or optimal assignments. For the larger problems vast amounts of assignments have been found; though this requires substantial computation times (since the algorithm must run until it converges to a global minimum), the numbers of assignments found are so large that it is worth the effort if the user is interested in obtaining multiple solutions.
- A drawback of this method is that it does require substantial computational effort as in each iteration a square symmetric matrix has to be factorized at least once; for large problems this matrix is considerably large, and therefore computation times

will increase. Reducing the problem size as much as possible is therefore important. For the *FAP* several preprocessing methods, both exact and heuristic, have been developed. These methods work quite well and subsequently good solutions are found using potential reduction.

- The results described in this paper were obtained using a MATLAB™/FORTRAN implementation. Computation times can substantially be improved when using an efficient low level language implementation.

As the results indicate, potential reduction methods can be quite effective in solving difficult combinatorial optimization problems. An overview paper by Tiourine et al. [18] evaluates the assets and drawbacks of the various approaches which have been applied to the *FAP* within the CALMA project. Although it is hard to compare the various algorithms, since different programming languages and hardware were used to test them, the following tentative conclusions are drawn. With respect to effectiveness, the potential reduction method is ranked along with simulated annealing, variable depth search and certain types of genetic algorithms, while it rates slightly better than taboo search and other types of genetic algorithms. As far as efficiency is concerned, the method is comparable to the above-mentioned algorithms.

Taking into consideration the short history of the method, the outlook is promising: more experimentation with potential reduction methods on various kinds of combinatorial optimization problems will give us more insight in the behavior of the algorithm and thus lead us to apply it as successful as possible.

Acknowledgements

We thank the anonymous referees for their useful comments.

References

[1] K.I. Aardal, A. Hipolito, C.P.M. van Hoesel, B. Jansen, C. Roos and T. Terlaky, A branch-and-cut algorithm for the frequency assignment problem, CALMA project, Delft and Eindhoven Universities of Technology, The Netherlands (1995).

[2] H.P. van Benthem, GRAPH: Generating Radio link frequency Assignment Problems Heuristically, Master's Thesis, Faculty of Technical Mathematics and Informatics, Delft University of Technology, The Netherlands (1995).

[3] A.A.F. Bloemen, Frequency assignment in mobile telecommunication networks, Report of the post-graduate program 'Mathematics for Industry', Eindhoven University of Technology, Eindhoven, The Netherlands (1992).

[4] I.I. Dikin, Iterative solution of problems of linear and quadratic programming, Dokl. Akad. Nauk SSSR 174 (1967) 747–748. Translated into English in Soviet Math. Dokl. 8 (1967) 674–675.

[5] O.E. Flippo and B. Jansen, Duality and sensitivity in nonconvex quadratic optimization over an ellipsoid, European J. Oper. Res. 94 (1996) 167–178.

[6] A.P. Kamath, N.K. Karmarkar, K.G. Ramakrishnan and M.G.C. Resende, Computational experience with an interior point algorithm on the satisfiability problem, Ann. Oper. Res. 25 (1990) 43–58.

[7] S.E. Karisch and F. Rendl, Semidefinite programming and graph equipartition, Report 302, CDLDO 5, Institut für Mathematik, Technische Universität Graz, Austria (1995).

[8] N. Karmarkar, A new polynomial-time algorithm for linear programming, Combinatorica 4 (1984) 373–395.
[9] N. Karmarkar, An interior-point approach to NP-complete problems – part I, Contemp. Math. 114 (1990) 297–308.
[10] N. Karmarkar, M.G.C. Resende and K.G. Ramakrishnan, An interior point algorithm to solve computationally difficult set covering problems, Math. Programming 52 (1991) 597–618.
[11] L.G. Khachiyan, A polynomial algorithm in linear programming, Dokl. Akad. Nauk SSSR 244 (1979) 1093–1096. Translated into English in Soviet Math. Dokl. 20 (1979) 191–194.
[12] T.A. Lanfear, Graph theory and radio frequency assignment, Technical Report, Allied Radio Frequency Agency NATO Headquarters, B-1110 Brussels, Belgium (1989).
[13] P.M. Pardalos and J. Xue, The maximum clique problem, J. Global Optim. 4 (1994) 301–328.
[14] F. Rendl and H. Wolkowicz, A semidefinite framework for trust region subproblems with application to large scale minimization, CORR Report 94-32 (Revised), University of Waterloo, Canada (1996).
[15] C.-J. Shi, A. Vannelli and J. Vlach, An improvement on Karmarkar's algorithm for integer programming, COAL Bull. 21 (1992) 23–28.
[16] D.C. Sörensen, Newton's method with a model trust region modification, SIAM J. Numer. Anal. 19 (1982) 409–426.
[17] S. Tiourine, C. Hurkens and J.K. Lenstra, Frequency assignment by local search, CALMA project, Department of Mathematics and Computing Science, Eindhoven university of Technology, Eindhoven, The Netherlands (1994).
[18] S. Tiourine, C. Hurkens and J.K. Lenstra, An overview of algorithmic approaches to frequency assignment problems, CALMA project, Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands (1995).
[19] J.P. Warners, A potential reduction approach to the Radio Link Frequency Assignment Problem, Master's Thesis, Faculty of Technical Mathematics and Informatics, Delft University of Technology, The Netherlands (1995).
[20] J.P. Warners, T. Terlaky, C. Roos and B. Jansen, Potential reduction algorithms for structured combinatorial optimization problems, Technical Report 95-88, Faculty of Technical Mathematics and Informatics, Delft University of Technology, The Netherlands (1995). Oper. Res. Lett., to appear.
[21] Y. Ye, On affine scaling algorithms for nonconvex quadratic programming, Math. Programming 56 (1992) 285–300.
[22] Y. Zhang, LIPSOL – a MATLAB^{TM} toolkit for linear programming interior-point solvers, Department of Mathematics and Statistics, University of Maryland Baltimore Country, Baltimore, Maryland (1994). FORTRAN subroutines written by E.G. Ng and B.W. Peyton (ORNL), J.W.H. Liu (Waterloo), and Y. Zhang (UMBC).