FREQUENCY ALLOCATION AND LINEAR PROGRAMMING

Martin Hellebrandt [1], Frank Lambrecht [2], Rudolf Mathar [1],
Thomas Niessen [1], Rainer Starke [2]

[1]Aachen University of Technology, D-52056 Aachen, Germany
[2]Mannesmann Mobilfunk GmbH, D-40543 Düsseldorf, Germany

Abstract The present paper deals with optimal fixed channel assignment for large real-world cellular radio networks. Examples are taken from data of the D2-network, operated by Mannesmann Mobilfunk (MMO) in Germany. Because of the huge size of the problems an exact optimal solution is presently out of reach. We present a heuristic iterative approach which performs extremely well, and significantly outperforms channel designs presently used by network operators. The basic ingredients of our approach are 1. fast and well established simple heuristics as initial assignments, 2. splitting the whole problem into smaller subproblems which can be optimized efficiently by solving a binary linear program (BLP), and repeating this process iteratively, 3. past-processing the resulting near-optimal design to avoid undesirable properties. A lot of detailed problems must be solved, such as a powerful preprocessing of constraints for the BLPs, and a carefull selection of the subproblems in 2. In summary, a very flexible tool is derived, also capable of taking into account external constraints from practical requirements.

I. INTRODUCTION

In GSM-systems frequency allocation is performed offline. Usually, the allocation is based on estimates of the traffic density and the propagation conditions [3, 8]. Based on the Erlang per cell values of the *busy hour* traffic, for every cell the smallest number of frequencies is computed such that the blocking probability is below a certain threshold, usually 1-2%. From certain propagation models, the interference probability between each pair of cells is obtained. These values can be used to derive the compatibility matrix. Once this is done, the problem of allocating frequencies to cells is a purely combinatorial, for practical instances large scale optimization problem.

In [4] a restricted model is considered, where interference depends on a certain distance condition and only co-channel interferences can occur. It is shown that under these hypotheses the frequency allocation problem can be formulated as an integer linear program. However, this description cannot be carried over to the general case. Moreover, in most cases the algorithm presented in [4] is of no practical use.

The aim of this paper is to describe methods that have been deployed recently for frequency allocation within the D2-network in Germany, operated by Mannesmann Mobilfunk (MMO). We start with introducing the frequency allocation problem as a binary linear program (BLP).

Although integer and binary linear programs are well investigated, formulating frequency allocation as a BLP is only a minor step towards a solution, since there is no algorithm performing globally well on different types of BLPs. Usually, a problem adapted preprocessing is inevitable. The BLP used in this paper contains a set of constraints typical for the so-called *node-packing* problem (a classical combinatorial optimization problem from graph theory). There are several suggestions for its preprocessing in the literature. Amazingly, it turns out that a relatively simple procedure, namely *maximum clique substitution*, yields excellent results (see section 3).

In section 4, additional restrictions arising in practice are described. Moreover, we show how each kind of restriction can be formulated by adding certain constraints to the binary linear programs. However, because of their enormous size, an exact solution of these programs is completely out of reach. This problem is tackled in section 5 by approximating the optimal solution stepwise. At each step of our procedure a small binary linear sub-program is solved. Each of these small programs has the same structure as the whole problem, and hence can be treated by the methods described in sections 3 and 4.

Finally, in section 6 we report on numerical applications and compare our results with frequency allocation designs formerly used by Mannesmann Mobilfunk.

II. FREQUENCY ALLOCATION

Let $\mathcal{Z}$ denote a cellular network consisting of n cells $Z_1, \dots, Z_n$. The aim is to allocate N frequencies $1, 2, \dots, N$ to the cells of the network such that certain constraints are satisfied. These constraints are described by a symmetric *compatibility matrix* $\mathbf{C} =$

[1]This work was supported by Deutsche Forschungsgemeinschaft grant no. Ma1184/4-3.

0-7803-5565-2/99/$10.00 © 1999 IEEE

$(c_{ij})_{i,j=1\dots,n}$ and a *requirement vector* $\mathbf{r} = (r_i)_{i=1\dots n}$, both consisting of non-negative integer entries. The meaning of $\mathbf{C}$ and $\mathbf{r}$ are as follows. Frequencies k and l can be allocated to cells Z_i and Z_j simultaneously, only if $|k-l| \geq c_{ij}$, or $i = j$ and $k = l$. The values r_i describe the demand of cell Z_i, i.e., the number of frequencies to be assigned to cell Z_i.

A frequency allocation design is described by a 0-1 matrix $\mathbf{X} = (x_{ik})_{i,k=1}^{n,N}$, where $x_{ik} = 1$ if and only if frequency k is assigned to cell Z_i. Therefore, a solution of a frequency allocation problem (FAP) consisting of $\mathcal{Z}$, $\mathbf{C}$ and $\mathbf{r}$ is a matrix $\mathbf{X} = (x_{ik})_{i,k=1}^{n,N}$ such that

$$x_{ik} = x_{jl} = 1 \Rightarrow |k-l| \geq c_{ij} \text{ or } (i,k) = (j,l), \quad (1)$$

$$\sum_{k=1}^{N} x_{ik} = r_i \text{ for every } i = 1,\dots,N. \quad (2)$$

III. LINEAR PROGRAMMING

A binary linear program consists of a linear objective function which is to be maximized with respect to a set of linear inequalities and equalities, where the values of all variables are restricted to 0 and 1.

Let $\mathbf{X} = (x_{ik})_{i,k=1}^{n,N}$ be a 0-1 matrix. By (1), $\mathbf{X}$ is an admissible frequency allocation if and only if

$$x_{ik} + x_{jl} \leq 1 \text{ for all } (i,k) \neq (j,l),\ |k-l| < c_{ij}. \quad (3)$$

If in addition $\mathbf{X}$ satisfies (2), then $\mathbf{X}$ is a solution of the FAP. However, (2) makes the problem complicated, since it is not trivially feasible. Hence we will replace this restriction by related inequalities and add an objective function as follows.

BLP: Maximize the objective function

$$f(\mathbf{X}) = \sum_{i=1}^{n}\sum_{k=1}^{N} x_{ik}$$

such that

$$x_{ik} \in \{0,1\} \text{ for all } i = 1,\dots,n,\ k = 1,\dots,N, \quad (4)$$

$$x_{ik} + x_{jl} \leq 1 \text{ for all } (i,k) \neq (j,l),\ |k-l| < c_{ij}, \quad (5)$$

$$\sum_{k=1}^{N} x_{ik} \leq r_i \text{ for all } i = 1,\dots,n. \quad (6)$$

A solution $\mathbf{X} = (x_{ik})_{i,k=1}^{n,N}$ of the FAP corresponds to a solution of the BLP with $f(\mathbf{X}) = \sum_{i=1}^{n} r_i$, and vice versa. In this respect, the BLP is equivalent to the FAP. If the FAP has no solution, the BLP can still be solved and the value of the objective function satisfies $f < \sum_{i=1}^{n} r_i$. This provides useful information, since $\sum_{i=1}^{n} r_i - f$ is the minimum number of unsatisfied requirements over all admissible frequency allocations. A key advantage of the BLP is that it is always trivially feasible, that is, $x_{ik} = 0$ for all $i = 1,\dots,n$, $k = 1,\dots,N$ is admissible with respect to (4), (5), and (6).

The above BLP is still an unfavorable starting point for codes solving integer linear programs. Such codes usually apply branch-and-bound, where in each step a (relaxed) linear program is solved. This can be briefly described as follows. First, the integral constraints are *relaxed*, which means that (4) is replaced by

$$0 \leq x_{ik} \leq 1 \text{ for all } i = 1,\dots,n,\ k = 1,\dots,N, \quad (7)$$

and the resulting linear program is solved. If an integral solution is found, then we are done. Otherwise, certain variables are fixed to admissible integral values, and the (partially) relaxed linear program is solved. Once an integral solution is obtained, the value of the objective function can be used to cut off parts of the branching tree in forthcoming steps.

To improve the upper bounds in the branching tree, a preprocessing step is ususally applied to the constraints of the integer linear program (see, e.g., [7]). In the following we present a technique which yields a significant speed-up of computation time for the FAP related BLP. First, some graph theoretical notation is introduced.

A graph $G = (V,E)$ consists of a finite set of nodes V and a set of edges $E \subseteq \{(u,v)|u,v \in V, u \neq v\}$. Every set $U \subseteq V$ defines a subgraph $H = (U,E')$ of G, where $E' = E \cap \{(u,v)|u,v \in U, u \neq v\}$. H is called the *induced subgraph* of U in G. An induced subgraph is called a *clique* of G, if every pair of distinct nodes is joined by an edge. A clique is called *maximal*, if it is not properly contained in another clique.

A *node-packing problem* is a binary linear program such that the constraints are all of the form $x_i + x_j \leq 1$, $i \neq j$. These constraints can be described by a graph $G = (V,E)$, where V is the set of variables, say $V = \{x_1,\dots,x_p\}$, and $(x_i,x_j) \in E$ if and only if $x_i + x_j \leq 1$ is a constraint. In the objective function $f = \sum_{i=1}^{p} c_i x_i$ of a node-packing problem each c_i may be seen as weight of node x_i. Hence, a solution of the node-packing problem corresponds to a set of pairwise non-adjacent nodes with maximum cumulative weight. It is well-known (see, e.g., [2]) that node-packing is NP-hard.

The cliques of the graph G are of great importance for the node-packing problem, since at most one node of each clique can belong to a set of pairwise non-adjacent nodes. In other words, if $U \subseteq V$ induces a clique in G, then $\sum_{u \in U} u \leq 1$ is a valid inequality of the node-packing problem. If $|U| \geq 3$, then adding this inequality to the constraints of the node-packing problem cuts off non-integral, extremal points of the polyhedron of the relaxed linear problem. Obviously, this method is most efficient if U is a maximal clique. Hence, it is desirable to know the node sets of all maximal cliques of G. However, it is easy to see that the number of these sets can grow exponentially in $|V|$ and therefore cannot be computed efficiently. Consequently, a suitable selection of maximal cliques is of great importance.

Let us return to the above BLP. The constraints (5) describe a node-packing problem on a graph G with node set $V = \{x_{ik} \mid i = 1, \dots, n, \ k = 1, \dots, N\}$. An equivalent set of constraints is obtained as follows. By means of a greedy algorithm sets $U_1, \dots, U_q \subseteq V$ are determined such that each U_i induces a maximal clique in G, and each edge of G is contained in at least one of these cliques. Thereafter, (5) is replaced by

$$\sum_{x_{ik} \in U_l} x_{ik} \leq 1 \quad \text{for all} \quad l = 1, \dots, q. \tag{8}$$

This maximum clique substitution yields an equivalent set of constraints. (8) contains only valid inequalities for the BLP, and hence we can add these inequalities to (5) without changing the set of feasible points. Moreover, all inequalities from (5) can be removed, since each of them is implied by an inequality from (8) containing the corresponding edge of G.

IV. ADDITIONAL RESTRICTIONS

Certain additional constraints must be taken into account for real-world instances. First of all, if network providers use the same spectrum, frequencies must be negotiated in advance for interfering cells. Hence, cells may have lists of *prescribed* and *forbidden* frequencies.

If frequency k is prescribed, or banned, in cell Z_i, the additional constraint $x_{ik} = 1$, or $x_{ik} = 0$, respectively, is used in the BLP. The version of the BLP that includes all constraints resulting from prescribed and forbidden frequencies will be call *extended* BLP.

Obviously, at least one frequency must be assigned to each cell of the network. Since the requirements r_i are strictly positive, no additional condition is necessary whenever the FAP is solvable. Otherwise, a solution of the BLP will not necessarily assign at least one frequency to every cell. To achieve this, the following constraints are near at hand.

$$\sum_{k=1}^{N} x_{ik} \geq 1 \quad \text{for all} \quad i = 1, \dots, n. \tag{9}$$

However, the above constraints prevent the BLP from being trivially feasible. It may even become infeasible. To avoid this problem, new binary variables $y_1, \dots, y_n$ are introduced and the constraints

$$y_i \in \{0, 1\} \quad \text{for all} \quad j = 1, \dots, n, \quad \text{and} \tag{10}$$

$$y_i \leq \sum_{j=1}^{N} x_{ij} \quad \text{for all} \quad i = 1, \dots, n \tag{11}$$

are added. The meaning of the new variables is as follows. In a solution of the binary linear program, $y_i = 0$ will indicate that no frequency is allocated to cell Z_i. We aim at maximizing the objective function f^*, given by

$$f^* = f + p \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \Big(p \cdot y_i + \sum_{j=1}^{N} x_{ij} \Big),$$

where the penalty value p is a positive integer. By using the objective function f^* we accept a degradation of at most p allocated frequencies for every additionally covered cell.

V. HANDLING OF LARGE INSTANCES

In spite of the fact that modern integer programming codes are very powerful tools for solving combinatorial optimization problems, instances with several thousands of variables often exceed the scope of such codes. The size of the binary linear programs resulting from data of real-world networks provided as examples by Mannesmann Mobilfunk (30.000-70.000 variables) is out of reach of the code we use. Therefore, we replaced the solution of the extended BLP by an iterated procedure, where at each stage an embedded much smaller binary linear program is solved.

Assume that an admissible frequency allocation $\mathbf{X} = (x_{ik})_{i,k=1}^{n,N}$ is found. Then $(x_{ik})_{i,k=1}^{n,N}$ is a feasible point of the constraints of the extended BLP. If all requirements are satisfied, then we are done. Otherwise, we select a subset of cells $\mathcal{Z}'$ of moderate size and try to find a better frequency allocation within this group. For this purpose the extended BLP is first formulated for the sub-network given by $\mathcal{Z}'$. To ensure that a frequency allocation for $\mathcal{Z}'$ will be compatible with the frequency allocation at the remaining cells, the list of forbidden frequencies for each cell in $\mathcal{Z}'$ is changed accordingly. Thereafter, the resulting binary linear program is solved. Let $(x'_{ik})_{i:Z_i \in \mathcal{Z}', k=1}^{N}$ denote the solution. Then $(x^*_{ik})_{i,k=1}^{n,N}$ defined by

$$x^*_{ik} = \begin{cases} x_{ik}, & \text{if } Z_i \notin \mathcal{Z}' \\ x'_{ik}, & \text{if } Z_i \in \mathcal{Z}' \end{cases}$$

is also an admissible frequency allocation for $\mathcal{Z}$. Moreover, the new value of the objective function f^* obviously satisfies $f^* \geq f$. If now all requirements are fulfilled, we are done. Otherwise, the same procedure is repeated with a new set of cells $\mathcal{Z}'$, until certain stopping criteria apply.

There are some interesting approaches to speed up computation time. First of all, not every binary linear subprogram must be solved. If for instance it turns out that the current BLP is intractable, then a different subnetwork $\mathcal{Z}'$ can be chosen. We used time limits to declare a subproblem intractable and switch to the next. Moreover, the values of the objective functions of the (relaxed) linear programs that are solved during the branch-and-bound algorithm are also of great value. If one of these is less than $\sum_{i:Z_i \in \mathcal{Z}'} \sum_{k=1}^{N} x_{ik} + 1$, then no improvement can be expected in the remaining part of the branching tree. In particular, if this happens with the first linear program, the actual procedure is stopped and a new set $\mathcal{Z}'$ will be chosen.

Much care must be taken with the selection of the set $\mathcal{Z}'$. The basic idea resembles multidimensional scaling [6]. Given pairwise dissimilarities between n objects, the aim of multidimensional scaling is to determine a configuration of n points in a Euclidean space such that the interpoint distances fit the given dissimilarities best.

The positive entries of the compatibility matrix $C = (c_{ij})_{i,j=1,\dots,n}$ are transformed to quantities which are roughly related to the geometrical distance between individual base stations by setting $\delta_{ij} = a/c_{ij}$. Closely neighboring cells will have a small δ_{ij}, while large δ_{ij} indicate that the cells are well separated. a is an appropriate scaling factor.

In practice, only integer values 0,1,2 and 3 occur in the compatibility matrix C, where $c_{ij} = 0$ is interpreted as missing distance information. Finding an approximate location of base stations in the plane reads as to

$$\min \; H(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{c_{ij}>0} \Big(||\boldsymbol{x}_i - \boldsymbol{x}_j|| - \delta_{ij} \Big)^2 \quad (12)$$

over all $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^2$. For real-world problem sizes this is an extremely difficult problem. Because of the huge number of local minima in (12) it is very unlikely that gradient based methods will find the global minimum.

We apply a special type of two-phase simulated annealing to the above problem. (For general information on simulated annealing, see [1].) The results are amazingly accurate. For an example with $n = 1181$ cells and 17342 positive entries in $\mathbf{C}$ the algorithm was stopped at $H = 12.369$. The average error per data point δ_{ij} is $7.13 \cdot 10^{-4}$.

The final coordinates $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ represent the interference pattern quite accurately in a geometrical form. These points are used for further processing in the algorithm which generates new candidates for the sub-BLP. In the following description we identify x_i and the corresponding cell Z_i.

1. Select a random cell $\boldsymbol{x}_i$.
2. Choose a radius r and include in $\mathcal{Z}'$ all cells x_j satisfying $||x_i - x_j|| \leq r$. Select r such that a certain number L of included cells is not exceeded. L is determined by the maximum acceptable size of the sub-BLP.

The resulting cells and the updated lists of forbidden frequencies are now passed to the corresponding BLP. Each request for a new candidate $\mathcal{Z}'$ from the main routine initiates the above steps 1. and 2. to be executed. Observe that the geometric configuration $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ needs to be calculated only once in advance.

VI. COMPUTATIONAL RESULTS

The techniques of the preceding sections were applied to 10 frequency assignment instances with data provided by Mannesmann Mobilfunk GmbH, Germany. First, an admissible initial allocation was determined by a fast graph coloring heuristic. Thereafter, successive improvements were found by iteratively solving subproblems, until no unsatisfied requirements remained or no further improvement could be found.

It turned out that some instances (namely 3,4,10) could be solved by the initial heuristic within a few seconds. All other problems demanded for a considerable amount of computation time, varying from some minutes to several hours. Subproblems of up to 1800 variables had to be solved. This corresponds to a size of 80 to 100 cells depending on the number of external constraints passed to the sub-BLP.

The results of our channel allocation strategy used on test data in comparison to the former algorithm in use at Mannesmann Mobilfunk are depicted in table 1. In all but the most trivial cases BLP could significantly improve the frequency allocation. In particular, for every instance a channel allocation could be calculated that assigns at least one frequency to every cell, whereas the former D2-algorithm left three cells uncovered in problem 2. Furthermore, the overall blocking probability $\mathcal{B}_{\mathcal{Z}}$ of this network could be decreased by more than 50%, where $\mathcal{B}_{\mathcal{Z}}$ is defined by

$$\mathcal{B}_{\mathcal{Z}} = \Big(\sum_{i=1}^{n} \rho_i \Big)^{-1} \sum_{i=1}^{n} \rho_i \mathcal{B}_{n_i}(\rho_i).$$

problem number	cells	former MMO algorithm uncovered cells	former MMO algorithm unsatisfied requirements	BLP method uncovered cells	BLP method unsatisfied requirements
1	558	0	19	0	8
2	597	3	160	0	111
3	782	0	0	0	0
4	921	0	0	0	0
5	1181	0	1	0	0
6	927	0	124	0	72
7	517	0	45	0	23
8	836	0	3	0	3
9	681	0	19	0	12
10	755	0	0	0	0

Table 1: Results of the BLP method compared to the former algorithm used by MMO.

Here $\mathcal{B}_k(x) = (\sum_{j=0}^{k} \frac{x^j}{j!})^{-1} \frac{x^k}{k!}$ denotes the first Erlang formula, ρ_i is the offered load in cell Z_i, and $n_i = \sum_{j=1}^{N} x_{ij}$ is the number of channels allocated to cell Z_i.

Further analysis [5] of certain test data showed that an average gain of capacity of 2.6% by the BLP method reduces the blocking rate by 30%.

VII. CONCLUSIONS

Fixed channel assignment (FCA) can be seen as a combinatorial optimization problem. However, because of the enormous size of present cellular radio networks an exact solution is completely out of reach. In order to provide near optimal feasible solutions for network management purposes, in this paper we suggest to formulate the channel assignment problem as a binary linear program (BLP). A feasible initial solution is determined by a fast and simple heuristic. We then iterate to choose successively subnets and optimize the corresponding sufficiently small BLP, until certain stopping criteria are satisfied. Finally, the number of cells with no channel so far is minimized by modifying the objective function. An appropriate selection procedure of subnets, and preprocessing of the corresponding BLPs are further important steps, which are satisfactorily solved.

The computational results presented in section 6 show that our approach yields a powerful and flexible software tool for automatic channel allocation. Channel designs derived by this tool significantly outperform allocations formerly in use.

References

[1] E. Aarts, J. Korst. *Simulated Annealing and Boltzmann Machines.* Wiley, Chichester, 1989.

[2] M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* H.W. Freeman, New York, 1979.

[3] F. Lambrecht and A. Baier. Methodik der Funknetzplanung für zellulare Mobilfunksysteme. *Mobile Kommunikation* (ITG-Fachbericht 135, B. Walke (ed.)), 517 – 524.

[4] M.G. Kazantzakis, P.P. Demestichas and M.E. Anagnostou. Optimum frequency reuse in mobile telephony systems. *Intern. J. Comm. Sys.* **8** (1995), 185 – 190.

[5] H. Kurth and R. Starke. Auswertung der Ergebnisse eines neuen Frequenzzuweissungsalgorithmus. *Internal Report, Mannesmann Mobilfunk GmbH* (Düsseldorf, 1997).

[6] R. Mathar. Multidimensional scaling with ℓ_p-distances, a unifying approach. *Information Systems and Data Analysis*, (H.H. Bock, W. Lenski, M.M. Richter (eds.)) Springer-Verlag, Heidelberg, 1994, 325 – 331.

[7] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization.* Wiley, 1988.

[8] J. Plehn. Applied frequency assignment. *IEEE 44th Vehicular Technology Conference* (Stockholm, 1994), 853 – 857.