Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

Ralf Borndörfer
Andreas Eisenblätter
Martin Grötschel
Alexander Martin

Frequency Assignment in Cellular Phone Networks

Preprint SC 97-35 (July 1997)

Frequency Assignment in Cellular Phone Networks*

R. Borndörfer A. Eisenblätter M. Grötschel A. Martin**

July 25, 1997

Abstract

We present a graph-theoretic model for the *frequency assignment problem* in Cellular Phone Networks: Obeying several technical and legal restrictions, frequencies have to be assigned to transceivers so that interference is as small as possible. This optimization problem is $\mathcal{NP}$-hard. Good approximation cannot be guaranteed, unless $\mathcal{P} = \mathcal{NP}$.

We describe several assignment heuristics. These heuristics are simple and not too hard to implement. We give an assessment of the heuristics' efficiency and practical usefulness. For this purpose, typical instances of frequency assignment problems with up to 4240 transceivers and 75 frequencies of a German cellular phone network operator are used. The results are satisfying from a practitioner's point of view. The best performing heuristics were integrated into a network planning system used in practice.

Keywords: Frequency Assignment Problem, Cellular Phone Network, Heuristics, Graph Coloring

Mathematics Subject Classification (1991): 90B12 05C15 90C60 68Q25

1 Introduction

High quality frequency assignments are crucial for the successful operation of todays heavily loaded cellular phone networks. Computing such assignments is difficult, whatever (reasonable) interpretation of high quality one has in mind. Our version of high quality focusses on minimizing interference. The mathematical formulation of this frequency assignment problem shows that it is a challenging generalization of several coloring problems in graph theory.

A variety of problems have been studied so far under the name of "frequency assignment" (the alternative term "channel assignment" is also in use). Hale [19] stated several frequency assignment problems as (generalized) graph coloring problems. Interference information is employed to derive a graph, sometimes called conflict graph, which has to be colored with as few channels or with channels from an as narrow interval as possible. Additional restrictions sometimes apply. Much work has been done in this direction [1, 6, 10, 11, 13, 14, 19, 21, 22, 30]. However, these approaches do generally not lead to satisfactory frequency assignments for cellular phone networks where the interval of available channels is given.

*This work is done in cooperation with `E-Plus` Mobilfunk GmbH, Germany. `E-Plus` operates a GSM1800 network. GSM1800 is a sibling of the GSM standard, the main difference between the two being the frequency band used.

**Konrad-Zuse Zentrum für Informationstechnik Berlin (ZIB), Takustr. 7, D-14195 Berlin, Germany. Email: {borndoerfer, eisenblaetter, groetschel, martin}@zib.de.

Interference minimization in mobile systems networks with a fixed spectrum of available channels is a more recent development [1, 7, 14, 15, 21, 22, 28, 30]. In this paper, we focus on fast and simple assignment heuristics. The heuristics developed are intended to be routinely used by practitioners to plan frequency assignments for cellular phone networks. All heuristics proposed have been implemented using `C++` and publicly available software libraries such as the Library of Efficient Data structures and Algorithms (LEDA) [26]. Five real-world networks of different size and structure are used to evaluate the performance. Huge interference reductions are achieved in comparison to assignments practically used, while, at the same time, speeding up the planning process considerably.

Several of the heuristics have been integrated into a network planning software system used at `E-Plus`.

2 Problem Description

The connection between a cellular phone user and his or her party is maintained by radio signals of some frequency. The radio signals of the cellular phone are received and propagated into a cable-based network by a nearby base transceiver station (BTS). This BTS is also used for the communication in the reverse direction. A BTS operates one or more elementary transceivers. Elementary transceivers are called TRXs in GSM-terminology [25] and will be represented by *carriers* in the mathematical model below.

Like a radio station, every TRX is assigned an operating frequency, whereas cellular phones may tune to various frequencies, just like radio sets. Similar to other radio based systems, the TRXs do not use arbitrary frequencies. The available radio spectrum is segmented into uniformly sized frequency slots which are called *channels* in this article. Each TRX operates on some channel. Between two TRXs using the same or adjacent channels significant interference may occur. This interference is called *co-channel* and *adjacent-channel interference*, respectively. The stronger the interference is the worse is the link quality. Interference exceeding some threshold is considered intolerable. To avoid intolerable interference, a minimum channel spacing between potentially interfering TRXs is introduced. A parameter, called *separation*, is set to one if the same channel must not be used for both TRXs. In case neither the same nor adjacent channels may be used, the separation parameter is set to two. For TRXs associated to one BTS an even larger separation may be necessary. The co- and adjacent channel interference predictions as well as the separation parameters are given between pairs of TRXs. We assume that these parameters are specified in three square matrices (the *co-channel interference matrix*, the *adjacent-channel interference matrix*, and the *separation matrix*) with rows and columns indexed by the TRXs.

Cellular phone network operators have a relatively small radio spectrum of 50 or 75 channels, say, at their disposal to operate some thousands TRXs. Some channels may even be *locally blocked*, i.e., they may not be used at any TRX of some BTS.

Our version of the *frequency assignment problem* is as follows:

> *Given are a list of TRXs, a range of channels, for each TRX a list of locally blocked channels, and the separation, the co-channel interference, and the adjacent-channel interference matrix.*
>
> *An* assignment *of channels to the TRXs has to be computed such that each TRX receives a locally not blocked channel, all separation requirements are met, and the sum over all interferences occurring between pairs of TRXs is minimal.*

Frequency assignments have to be computed on several occasions: the network is expanded or modified, a BTS is replaced by a different one with significantly different transmission power, or the interference predictions are corrected.

We give a **mathematical formulation** of the frequency assignment problem: Let (V, E) be an undirected graph. The nodes of the graph are the *carriers* representing the TRXs. The *spectrum* C is an interval of non-negative integers representing the range of channels. For every carrier v, a set $B_v \subseteq C$ of *blocked channels* is specified. The channels in $C \setminus B_v$ are called *available* at carrier v. B_v may be empty. Three functions, $d : E \to \mathbb{Z}_+$, $c^{co} : E \to [0, 1]$, and $c^{ad} : E \to [0, 1]$ with $c^{ad} \leq c^{co}$, are specified on the edge set. For an edge $vw \in E$, $d(vw)$ gives the *separation* necessary between channels assigned to v and w. $c^{co}(vw)$ and $c^{ad}(vw)$ denote the *co-channel* and *adjacent-channel interference*, respectively, which may occur between v and w. We will refer to the 7-tuple $N = (V, E, C, \{B_v\}_{v \in V}, d, c^{co}, c^{ad})$ as *carrier network*.

A *frequency assignment* or simply an *assignment* for N is a mapping $y : V \to C$. An assignment is *feasible* if every carrier $v \in V$ is assigned an available channel (from $C \setminus B_v$) and all separation requirements are met, i.e., $|y(v) - y(w)| \geq d(vw)$ for all $vw \in E$.

Definition 1 *Given a carrier network N, we call the optimization problem*

$$\min_{y \text{ feasible}} \sum_{\substack{vw \in E: \\ y(v) = y(w)}} c^{co}(vw) \quad + \sum_{\substack{vw \in E: \\ |y(v) - y(w)| = 1}} c^{ad}(vw) \tag{FAP}$$

frequency assignment problem.

The objective is to determine a feasible assignment that minimizes the total of co- and adjacent-channel interferences. Feasible assignments are a generalization of list colorings and are related T-colorings of graphs in the following way.

For a *list coloring* problem, a graph and lists of colors for every vertex are given. The task is to find a proper coloring of the graph using a color from its list for every vertex. Since an available channel has to be picked for every carrier, feasible assignments are list colorings.

T-colorings were introduced in [10]. Given an undirected graph G and a finite set T of non-negative integers containing 0. A *T-coloring* of G is a labeling f of the vertices of G with non-negative integers so that $|f(v) - f(w)| \notin T$ for all edges vw in G. In our case, there is a minimal distance required between adjacent carriers, expressed by the separation parameter. Every edge may thus have a different "T-set", but all those sets are restricted to be either empty or of the form $\{0, \dots, k\}$, for some non-negative integer k.

3 Computational Complexity

For every $q \in \mathbb{Q}_+$, we associate a decision problem q–FAP with the frequency assignment problem FAP:

> Given a carrier network N, decide whether N has a feasible assignment of cost no more than q. (q–FAP)

To discuss complexity issues we make the standard assumption that all numbers appearing as input data for FAP and q–FAP are rational and that they are encoded in binary form. It is easily observed that q–FAP is in $\mathcal{NP}$. This together with the fact that GRAPH K-COLORABILITY (see [17], GT4) can be reduced to q–FAP yields the following result.

Theorem 2 *For every $q \in \mathbb{Q}_+$, the decision problem* q–FAP *is $\mathcal{NP}$-complete.*

The standard notion of polynomial time approximation, see [4, 12, 27], for example, requires that a feasible solution can be produced in time polynomially bounded in the input size. FAP does not lend itself to approximation in this sense, since the proof of the preceding theorem reveals that finding a feasible assignment is already $\mathcal{NP}$-complete.

Corollary 3 *The problem of deciding whether an instance of* FAP *has a feasible solution is $\mathcal{NP}$-complete.*

Furthermore, it is also hard to find good approximate solutions for instances of FAP where obtaining a feasible solution is easy. More precisely, this can be stated in the following way.

Theorem 4 *Let N be an instance of* FAP *for which feasible solutions can be obtained in time polynomial in the input size. Then, unless $\mathcal{P} = \mathcal{NP}$, there exists an $0 < \varepsilon < 1$ such that the cost of an optimal assignment cannot be approximated within a factor of $|V|^{\varepsilon}$, where V is the set of carriers in N.*

This statement can be proved using a reduction of the MINIMUM GRAPH COLORING problem to FAPand thereby extending a result on the hardness of approximating MINIMUM GRAPH COLORING [3] to FAP.

4 Heuristics

As stated in the previous section, the frequency assignment problem belongs to the class of hard combinatorial problems. That is, one should not expect that a polynomial-time algorithm will always produce a feasible assignment. Even if a feasible assignment is produced, it is not guaranteed that its cost is close to optimal, e.g., within a small constant factor.

In this section, we describe some heuristics which can be used in practice to compute frequency assignments. Recall that our focus is on fast algorithms. We distinguish starting and improvement heuristics.

Starting heuristics compute a frequency assignment from scratch, step-wise extending an initially empty assignment to a complete assignment. Thus, as we go along, we are dealing with partial frequency assignments. A *partial* frequency assignment is a mapping $y : A \to C$ that is defined on a subset A of the carrier set V. In case $A = V$, a partial assignment is just an ordinary frequency assignment.

Improvement heuristics take a (feasible) assignment as input and try to improve it. Neither the assignment to be improved nor the assignments obtained in the course of computation are required to be feasible.

4.1 T-coloring

This starting heuristic [18] is a modification of a procedure used by Costa [9] in the context of T-colorings (see [10, 19]). The underlying algorithmic idea was first used in Brélaz's `DSATUR` [5] to compute ordinary graph colorings with few colors. This heuristic is the only one that does not try to minimize the cost of an assignment, but focusses on computing some feasible solution (which will tend to use few different channels).

```
Input:    G = (V,E), C, B_v for all v ∈ V, minimal distances d
Output: a feasible assignment y or a message that none was found

// Initialization
for all  v ∈ V  do
        // "saturation degree" = # of forbidden channels at carrier v
        satdeg[v] := |B_v|

        // "spacing degree" = sum over all d(vw), vw ∈ E with w unassigned
        spadeg[v] := sum over all d(vw) with vw ∈ E
// end for

// Assigning
U := V
while  U ≠ ∅  do
        let v ∈ U be a carrier whose satdeg[v] is maximal and among
          those one with maximal spadeg[v] (ties are broken arbitrarily)
        U := U \ {v}
        let y(v) be the available channel of least index at v
        if  no such available channels exists  then
                return "no feasible assignment found"
        for all  w ∈ U with vw ∈ E  do
                update satdeg[w], spadeg[w]
// end while

return y
```

Figure 1: Pseudo code for the T-coloring heuristic

Figure 1 gives a sketch of the algorithm. For each carrier not yet assigned, the *saturation degree* keeps track of how many channels are no longer available. The *spacing degree* is intended to represent how much impact assigning all the still unassigned neighbors of a carrier would have on its assignability. If the impact is very large, it should rather become assigned before most of its neighbors are. For similar reasons, carriers with high saturation degree should be assigned as soon as possible. The first `forall`-loop does the initialization. Assigning channels to carriers is done in the `while`-loop. Which carrier to assign next is determined by the saturation and spacing degrees.

The T-coloring heuristic is implemented using binary heaps for book-keeping which carrier to assign next. The running time obtained this way is $O(|C||E| + |E|\log|V|)$. The space requirement of the heuristic is $O(|C||V| + |E|)$.

4.2 Dual Greedy

The dual greedy is a starting heuristic that tries to avoid major decisions [20, 23]. Instead of going ahead assigning a channel to some carrier right away, it tries to identify what would be a particularly bad combination of a carrier and a channel. We will call a carrier-channel combination (v, f) an *available combination* if channel f is available at carrier v. Starting

Input: (V, E, C, $\{B_v\}_{v \in V}$, d, c^{co}, c^{ad}), control parameters M and W
Output: assignment y

// Initialization
U := set of all available combinations (v, f)
for all (v, f) in U do
    `penalty`[v,f] := 0

while $U \neq \emptyset$ do
    // Assigning
    for all (v, f) in U that are the only combination in U involving v do
        for all
            combinations (w, g) in U where
            vw $\in$ E and ((f = g and $c^{co}(vw) > 0$) or
                ($|g - f| = 1$ and $c^{ad}(vw) > 0$) or
                ($|g - f| < d(vw)$))
        do
            `penalty`[w,g] := `penalty`[w,g] + W

        set y(v) := f and remove (v, f) from U
    // end for

    // Eliminating combinations
    if $U \neq \emptyset$ then
        delete a combination (v, f) of highest weight

$$\sum_{\substack{(w,f)\in \underline{V}: \\ w\in\delta(v)\wedge d(vw)=0}} c^{co}(vw) + \sum_{\substack{(w,f\pm 1)\in \underline{V}: \\ w\in\delta(v)\wedge d(vw)\leq 1}} c^{ad}(vw) + \sum_{\substack{(w,g)\in \underline{V}: \\ w\in\delta(v)\wedge |f-g|<d(vw)}} M + \texttt{penalty}[v,f]$$

        from U
// end while

return y

Figure 2: Pseudo code for the Dual Greedy heuristic

from all available combinations of carriers and channels, the algorithm works its way through all of those, eliminating one “worst looking” combination at a time. For each carrier the last remaining carrier-channel combination is used to make an assignment.

Figure 2 shows a formulation of the algorithm in pseudo-code. One way to determine a weight of a carrier-channel combination is displayed together with the pseudo-code. Such a weighting is used as a measure for “badness” of a combination. The displayed measure is not the best performing weighting procedure investigated. We chose it for the sake of an easy exposition.

This approach hinges on identifying bad carrier-channel combinations. The successful application of the dual greedy requires extensive analysis of appropriate strategies to find bad combinations. Good strategies are problem dependent [20].

Fibonacci Heaps (see [8], for example) are used to keep track of bad carrier-channel combinations. Using those heaps, the dual greedy heuristic runs in $O(|C|^2|V|\log(|C||V|)+|C|^2|E|)$ time and uses $O(|C||V|+|E|)$ space.

4.3 DSATUR With Costs

This starting heuristic is another modification of Brélaz's `DSATUR` [5] incorporating ideas of Costa [9]. While in the setting of Brélaz and Costa the objective is to obtain an ordinary coloring using few colors or a T-coloring using a small interval of channels, respectively, our goal is to compute a feasible assignment using a given interval of channels incurring little cost.

A matrix `cost`, with rows indexed by the carriers in V and columns indexed by the channels in C, is used to record the cost of the different available combinations. First, we invalidate all entries corresponding to unavailable combinations of channels by an appropriately chosen entry `BLOCKED`.

Input: (V, E, C, $\{B_v\}_{v\in V}$, d, c^{co}, c^{ad})
Output: an assignment y, possibly infeasible

```
// Initialization
for all  v ∈ V  do
        initialize cost[v][f] to 0 if f is in C \ B_v, and to BLOCKED otherwise
        insert v into the heap with key |B_v|

// Assigning
while  the heap is not empty  do
        extract a carrier v with maximum key from the heap
        let y(v) be a non-blocked channel f of least value from row cost[v]
        update cost-matrix by adding Δ(v, f)
        update the keys of all carriers still in the heap

return y
```

Figure 3: DSATUR With Costs

A non-blocked channel is *bad* for a carrier if its matrix entry is at least as large as `BAD`, which is another suitably chosen constant. For every still unassigned carrier, a heap-entry is maintained. As the key for the heap serves the number of blocked or bad channels times BAD plus the sum over all non-blocked, non-bad row entries of the matrix `cost`. That is

$$key(v) = |B_v| \cdot \texttt{BAD} + \sum_{f \in C \setminus B_v} h(cost_{v,f}) \quad \text{with} \quad h(c) := \begin{cases} \texttt{BAD} & \text{if } c \geq \texttt{BAD}, \\ c & \text{otherwise.} \end{cases}$$

While the heap is not empty, a carrier v with maximum key is extracted and assigned its least costly available channel f. Such a channel may induce separation violations. But in that case (and if BAD was chosen big enough) all other available channels do, too. Next, all rows indexed by carriers adjacent to v are updated as well as the carriers' heap keys. The latter only happens in case they are still unassigned. Formally, a matrix $\Delta(v, f)$ is added to `cost`, where

$$\Delta_{w,g}(v, f) := \begin{cases} \texttt{BAD} & \text{if } vw \in E, g \in C \setminus B_w, |f - g| < d(vw), \\ c^{co}(vw) & \text{if } vw \in E, d(vw) = 0, f = g \in C \setminus B_w, \\ c^{ad}(vw) & \text{if } vw \in E, d(vw) \leq 1, f \pm 1 = g \in C \setminus B_w, \\ 0 & \text{otherwise.} \end{cases}$$

This heuristic is implemented using a Fibonacci heap for determining the carrier to assign next. The minimum-cost channel for a carrier is searched for in the corresponding row of the matrix `cost`. The running time obtained is $O(|C||E| + |V| \log |V|)$, assuming $|V| = O(|E|)$. The space requirement is $O(|V||C| + |E|)$.

It turns out that the choice of the first carrier to assign has considerable impact on the quality of the assignment obtained. No generally good rule could be identified as to which carrier to start with. One might start with each carrier in turn, and pick the best assignment obtained. A running time reducing option is to choose some set of start-carriers at random and then pick the best assignment computed this way.

4.4 Iterated 1-OPT

This improvement heuristic uses a neighborhood structure defined on the set of all assignments. Two assignments are considered *adjacent* if one can be obtained from the other by changing the channel of a single carrier. Given this neighborhood structure, an assignment y, and a carrier v, a *1-opt step* determines a least costly neighbor y' of y. If y' is at most as costly as y, y' becomes the current assignment. Otherwise, the assignment remains unchanged. An assignment y is considered less costly than an assignment y' if y implies fewer constraint violations, or, if both assignments violate equally many (or no) constraints, causes less interference than y'. To be more precise, we introduce some notation concerning the cost and the infeasibility of (partial) frequency assignments. This notation simplifies the formulation of the heuristic. We define the *cost* of a carrier-channel combination (v, f), $v \in V$, $f \in C$, with respect to the partial assignment y on A, denoted by y_A, as

$$c(y_A; (v, f)) := \frac{1}{2} \sum_{\substack{w \in \delta(v) \cap A: \\ f = y_A(w)}} c^{co}(vw) \quad + \quad \frac{1}{2} \sum_{\substack{w \in \delta(v) \cap A: \\ |f - y_A(w)| = 1}} c^{ad}(vw),$$

Input: (V, E, C, $\{B_v\}$, d, c^{co}, c^{ad}), partial assignment y_A
Output: assignment y'

```
// Initialization
y' := y
A' := A
order all carriers in A decreasingly according to
  infeas(y_A; (v, y(v))) and c(y_A; (v, y(v)))
put all unassigned carriers to the front

// perform a pass
for  every carrier v in V in the above order  do
       if  v ∉ A'  then
              add v to A'
       set y'(v) to a channel f so that (v, f) is minimal among all available
         combinations with respect to infeas(y'_A'; (v, f)) and c(y'_A'; (v, f))

return y'
```

Figure 4: Pseudo code for a pass of the Iterated 1-OPT heuristic

where $\delta(v)$ denotes the set of nodes incident to v in (V, E). The *infeasibility* of a carrier-channel combination (v, f), $v \in V$, $f \in C$, with respect to y_A is defined as

$$\text{infeas}(y_A; (v, f)) := \sum_{\substack{w \in \delta(v) \cap A: \\ |f - y_A(w)| < d(vw)}} 1 \quad + \quad \begin{cases} 1 & f \in B_v, \\ 0 & \text{otherwise.} \end{cases}$$

Iterated 1-OPT is an improvement heuristic that repeatedly selects a carrier and performs a 1-opt step. A sequence of 1-opt steps where every carrier is selected once is called a *pass*. Clearly, there is some freedom in selecting which carrier of the not yet examined ones to consider next. Experiments have shown that the following approach produces reasonably good results: The carriers are ordered decreasingly according to the infeasibility and the cost that the current carrier-channel combination incurs. Figure 4 gives a formulation of one pass of the algorithm.

Fibonacci Heaps are used to determine which carrier to consider next and what channel to assign to that carrier. The running time of a pass is $O(|C||E| \log |C| + |V| \log |V|)$ and the space required is $O(|C||V| + |E|)$.

Conceivably, several consecutive passes are capable of improving an assignment. The following mechanism aims at this phenomenon. A percentage is specified by which the next assignment has to outperform the previous one. If the latest assignment fails to achieve this goal, no further pass is performed. This variant is called *(multi-pass) Iterated 1-OPT heuristic*.

The repeated application of this heuristic will lead to an assignment that cannot be further improved by 1-opt steps. Such assignments are not necessarily optimal. The algorithm might be trapped in a local minimum.

We have also experimented with more complex exchange techniques such as “k-opt” and tested randomized exchange and search methods that also allow a cost or infeasibility increase.

These are often capable of producing better solutions, however, in general after very long running times, that are not acceptable for our industry partner.

4.5 Min-Cost Flow

The improvement heuristic called MCF tries to modify an assignment y yielding an assignment y' while obeying the following condition.

For a fixed linear order on the set V:

$$\forall vw \in E \text{ with } v < w : \quad \begin{array}{ccc} (y(v) \leq y(w) & \Rightarrow & y'(v) \leq y'(w)) \\ & \wedge & \\ (y(v) > y(w) & \Rightarrow & y'(v) \geq y'(w)) \end{array}$$

If a frequency assignment does not use all of the available channels, the Min-Cost Flow heuristic is likely to produce an assignment of lower cost by utilizing more of the channels available. (One might think in terms of making the histogram of the number of times a channel is used look more evenly.) A min-cost flow problem—giving the heuristic its name—is solved on a directed graph derived from the graph (V, E). The functions d, c^{co}, and c^{ad} are used to compute cost coefficients and arc capacities, respectively. The dual variables associated to a min-cost flow are integral and correspond to a frequency assignment. Assuming $B_v = \emptyset$ for all $v \in V$ and $2\,c^{ad}(vw) \leq c^{co}(vw)$ for all $vw \in E$ with $d(vw) = 0$, it can be proved that MCF computes an optimal assignment y' among all assignments satisfying the above condition. Details about this procedure will appear elsewhere.

The auxiliary directed graph is easily constructed in $O(|E|)$ time. The min-cost flow problem is solved using a Network Simplex Method implementation [24]. This algorithm has space requirement $O(|E|)$ but its worst-case running time is exponential in the input size. Although there are strongly polynomial min-cost flow algorithms (see [2]), we have chosen this implementation of the Network Simplex Algorithm since it turned out to be very fast in practice.

4.6 Tightening a separation

As before, let $N = (V, E, C, \{B_v\}_{v \in V}, d, c^{co}, c^{ad})$ denote a carrier network. Let v and w be adjacent carriers. The value $d(vw)$ is the minimal separation necessary between the channels assigned to v and w. So, if $d(vw) \geq 1$, the same channel must not be given to both carriers. Hence $d(vw) \geq 1$ debars co-channel interference between v and w. Similarly, if $d(vw) \geq 2$, no adjacent-channel interference can occur between v and w in a feasible assignment. An approach to control interference is to exclude assignments causing large interference between pairs of carriers. To achieve this goal, a *threshold* t is introduced. The threshold is used to produce a problem which prescribes a sufficiently large separation between carriers that may cause interference exceeding t:

$$d^t(vw) := \begin{cases} \max\{1, d(vw)\} & \text{if } c^{co}(vw) > t \quad \wedge \quad c^{ad}(vw) < t, \\ \max\{2, d(vw)\} & \text{if } c^{ad}(vw) > t, \\ d(vw) & \text{otherwise.} \end{cases}$$

Instance	$\|V\|$	$\|E\|$	density [%]	minimum degree	average degree	maximum degree	# components	component sizes	diameter of components	maximum clique in components	$\|\{e \in E : d(e) \neq 0\}\|$	$\|\{e \in E : d(e) = 1\}\|$	$\|\{e \in E : d(e) = 2\}\|$	$\|\{e \in E : d(e) = 3\}\|$	$\|\{e \in E : c^{co}(e) \neq 0\}\|$	$\sum_{e \in E} c^{co}(e)$	$\|\{e \in E : c^{ad}(e) \neq 0\}\|$	$\sum_{e \in E} c^{ad}(e)$
k	267	20164	57	2	151	238	1	267	3	69	1053	4	1046	3	19111	2857.44	996	28.87
a	353	11746	19	0	66	174	6	1×348 5×1	1×5 5×0	1×27 5×1	1265	8	1252	5	10481	1516.56	233	8.07
f	2877	187753	5	0	130	453	58	1×2786 34×2 23×1	1×12 34×1 23×0	1×69 34×2 23×1	15210	4049	9911	1250	172543	29146.7	24952	983.17
l	2918	186787	4	1	128	335	36	1×2832 1×18 34×2	1×16 1×2 34×1	1×96 1×14 34×2	24283	8730	14094	1459	162504	27852.2	38004	1185.33
h	4240	529000	6	11	249	561	1	4240	10	130	29524	9470	17934	2120	499476	79092.4	103290	2354.65

Table 1: Parameters of the problem instances supplied by `E-Plus`. In the column labeled "component sizes" entries such as "1×348, 5×1" express that there is one component of size 348 and 5 singletons. The entries in the columns "diameter of components" and "maximum clique in components" are to be read likewise.

The carrier network $N^t = (V, E, C, \{B_v\}_{v \in V}, d^t, c^{co}, c^{ad})$ is obtained from N by *tightening the separation with* t. A feasible assignment for N^t may incur interference, but none exceeding the threshold t. Thus, feasible assignments for the original problem may be infeasible for the modified problem. Since an assignment causing high interference between some pair of carriers might save considerably between others, it may be the case that no optimal assignment for the original problem is feasible for the modified one. Despite this fact, tightening the separation often works well in conjunction with the heuristics. By applying the heuristics described above to N^t for different threshold values, solutions of varying quality are usually obtained. Depending on the heuristic and the problem instance at hand, a suitable threshold value may be determined by some search routine.

Assignment	total interference	co-channel interference	adjacent-channel interference	separation violations	invalid assignments	unassigned carriers	time [secs]
Original	23.7416	23.2958	0.4458	8	6	21	—
+ (MCF 1-OPT)*	2.3729	2.1756	0.1974	0	0	0	2.84
RANDOM	54.1935	53.3958	0.7977	52	0	0	0.01
+ (MCF 1-OPT)*	2.6981	2.3785	0.3197	0	0	0	5.21
T-coloring	5.0286	4.6600	0.3686	0	0	0	0.27
+ (MCF 1-OPT)*	1.7982	1.7007	0.0976	0	0	0	2.25
DSATUR 0%	1.2755	1.2440	0.0315	0	0	0	0.31
+ (MCF 1-OPT)*	1.2232	1.1944	0.0288	0	0	0	2.28
DSATUR 1%	1.0761	1.0377	0.0384	0	0	0	0.62
+ (MCF 1-OPT)*	1.0430	1.0056	0.0374	0	0	0	2.20
DSATUR 5%	1.1059	1.0549	0.0510	0	0	0	4.83
+ (MCF 1-OPT)*	1.0547	1.0116	0.0431	0	0	0	2.36
DSATUR 10%	0.9799	0.9433	0.0366	0	0	0	7.65
+ (MCF 1-OPT)*	0.9701	0.9347	0.0354	0	0	0	2.16
DSATUR 25%	0.9799	0.9433	0.0366	0	0	0	22.02
+ (MCF 1-OPT)*	0.9701	0.9347	0.0354	0	0	0	2.21

Table 2: Assignments computed for Problem **k** with 50 channels. The separation is tightened with a threshold of 0.035.

5 Computational Experiments

In the following, computational results on 5 problem instances, named **k**, **a**, **f**, **l**, and **h**, are shown. These instances stem from real-world cellular phone networks. The chosen instances differ in size as well as in structure. Table 1 lists several parameters of the instances. Following the name of the problem instance, the next 10 columns display properties of the underlying graph $G = (V, E)$. No edge $vw \in E$ satisfies $d(vw) = c^{co}(vw) = c^{ad}(vw) = 0$. The remaining 8 columns show features of d, c^{co}, and c^{ad}, in particular, the size of the supports. Almost all sets B_v of locally blocked channels are empty. Therefore, no detailed information on the

Assignment	total interference	co-channel interference	adjacent-channel interference	separation violations	invalid assignments	unassigned carriers	time [secs]
Original	1.1564	1.0973	0.0591	0	3	0	—
+ (MCF 1-OPT)*	0.6066	0.5464	0.0602	0	0	0	1.99
RANDOM	54.9708	54.4163	0.5545	118	0	0	0.02
+ (MCF 1-OPT)*	0.8791	0.8092	0.0699	0	0	0	1.79
T-coloring	0.9135	0.8845	0.0290	0	0	0	0.16
+ (MCF 1-OPT)*	0.1623	0.1427	0.0196	0	0	0	2.01
DSATUR 0%	0.0292	0.0226	0.0066	0	0	0	1.48
+ (MCF 1-OPT)*	0.0223	0.0223	0.0000	0	0	0	1.42
DSATUR 1%	0.0318	0.0304	0.0015	0	0	0	0.34
+ (MCF 1-OPT)*	0.0258	0.0244	0.0015	0	0	0	1.48
DSATUR 5%	0.0209	0.0209	0.0000	0	0	0	2.82
+ (MCF 1-OPT)*	0.0189	0.0189	0.0000	0	0	0	1.43
DSATUR 10%	0.0261	0.0235	0.0026	0	0	0	7.09
+ (MCF 1-OPT)*	0.0248	0.0222	0.0026	0	0	0	1.57
DSATUR 25%	0.0177	0.0177	0.0000	0	0	0	15.13
+ (MCF 1-OPT)*	0.0175	0.0175	0.0000	0	0	0	1.46

Table 3: Assignments computed for Problem **a** with 30 channels. The separation is tightened with a threshold of 0.01.

B_v's is given. The size of the spectrum for problems **k** and **f** is 50, for **a** it is 30, and 75 for problems **h** and **l**.

Every carrier network is either connected or has *one major* component. The density, the diameter of the major component, and its clique number all indicate that the graph is very far from being planar. In all problems but **a**, the maximum clique exceeds the spectrum size. This does not necessarily imply that no feasible assignment exists, but the fact can be used to derive a lower bound on the interference in feasible assignments.

Each instance was supplied by `E-Plus` together with a (partial) frequency assignment. This assignment was either manually or automatically generated using a commercial program for solving the frequency assignment problem. This program implements the algorithm described in [16].

In Tables 2, 3, 4, 5, and 6 the quality of the supplied assignment is shown for comparison. The first column in each table lists the source of the frequency assignment. In rows headed by a '+', the preceding assignment was used to improve on. In columns two, three, and four the interference incurred is listed, with the third and fourth column breaking the total up into co-channel and adjacent-channel interference. The column titled "separation violations" contains the number of violated minimal distance constraints. The next two columns show the number of invalidly assigned and unassigned carriers. A feasible assignment has to have zeros in all three columns that were mentioned last. Finally, the rightmost column lists the time consumed to run the starting or improvement heuristic, respectively. *The computations were performed on a SUN SPARCstation 20-501.*

Assignment	total interference	co-channel interference	adjacent-channel interference	separation violations	invalid assignments	unassigned carriers	time [secs]
Original	52.1004	40.8344	11.2661	0	0	3	—
+ (MCF 1-OPT)*	23.6927	19.6156	4.0771	0	0	0	51.43
RANDOM	616.9697	578.5104	38.4593	808	0	0	0.13
+ (MCF 1-OPT)*	22.1688	18.2175	3.9513	0	0	0	49.06
T-coloring	84.5410	61.5274	23.0136	0	0	0	2.83
+ (MCF 1-OPT)*	18.8387	15.2667	3.5720	0	0	0	213.75
DSATUR 0%	9.4011	8.1711	1.2299	0	0	0	19.71
+ (MCF 1-OPT)*	8.9613	7.8807	1.0806	0	0	0	134.50
DSATUR 1%	8.8580	7.6198	1.2382	0	0	0	96.37
+ (MCF 1-OPT)*	8.5398	7.3106	1.2291	0	0	0	89.31
DSATUR 5%	8.9662	7.7877	1.1784	0	0	0	471.20
+ (MCF 1-OPT)*	8.6693	7.4998	1.1695	0	0	0	127.46
DSATUR 10%	8.8684	7.8654	1.0030	0	0	0	1011.78
+ (MCF 1-OPT)*	8.7380	7.7631	0.9749	0	0	0	134.88
DSATUR 25%	8.7733	7.6393	1.1340	0	0	0	2342.90
+ (MCF 1-OPT)*	8.5808	7.4550	1.1258	0	0	0	87.53

Table 4: Assignments computed for Problem **f** with 50 channels. The separation is tightened with a threshold of 0.05.

“RANDOM” is a trivial starting heuristic which randomly assigns an available channel to every carrier. Possible separation constraint violations are of no concern. “(MCF 1-OPT)*” stands for alternatingly applying MCF and Iterated 1-OPT until no more improvement is obtained during Iterated 1-OPT. The percentage listed following “DSATUR” tells how many of the carriers were checked out as a starting node for applying DSATUR With Costs. On calling DSATUR With Costs a `threshold` to tighten the separation is supplied. The value of this parameter is given in the annotation to every table listing computational results. A summary of the performance of the heuristics is given below.

5.1 T-coloring

The T-coloring heuristic mostly succeeds in computing a feasible frequency assignment. These assignments are typically of inferior quality, although the quality may be affected by the threshold used for tightening the separation. The assignments tend to use only frequencies from an initial segment of the interval of available frequencies. Thus, large improvements are possible when applying MCF and Iterated 1-OPT.

5.2 Dual Greedy

The dual greedy heuristic turned out to be an overall failure. Extensive experiments did not show any regularity as to how the parameters of the heuristic could be tuned to achieve feasible assignments of competitive quality. In order to increase the performance, a special

Assignment	total interference	co-channel interference	adjacent-channel interference	separation violations	invalid assignments	unassigned carriers	time [secs]
Original	89.1725	74.8875	14.2849	0	0	10	—
+ (MCF 1-OPT)*	32.5201	24.4670	8.0531	0	0	0	25.06
RANDOM	506.5964	467.6431	38.9533	1040	0	0	1.26
+ (MCF 1-OPT)*	32.5016	23.3914	9.1102	0	0	0	107.14
T-coloring	175.0639	123.5290	51.5349	0	0	0	1.48
+ (MCF 1-OPT)*	25.0412	17.6132	7.4280	0	0	0	133.14
DSATUR 0%	17.0741	12.2901	4.7841	1	0	0	31.51
+ (MCF 1-OPT)*	16.8456	12.1753	4.6702	0	0	0	194.40
DSATUR 1%	15.0838	10.9065	4.1773	0	0	0	175.15
+ (MCF 1-OPT)*	14.9117	10.7065	4.2052	0	0	0	167.80
DSATUR 5%	14.8636	10.6555	4.2081	0	0	0	685.24
+ (MCF 1-OPT)*	14.6246	10.4734	4.1512	0	0	0	165.14
DSATUR 10%	14.7445	10.3496	4.3949	0	0	0	1237.07
+ (MCF 1-OPT)*	14.5592	10.1898	4.3694	0	0	0	164.40
DSATUR 25%	14.1321	10.2416	3.8905	0	0	0	3284.12
+ (MCF 1-OPT)*	13.9559	10.0677	3.8882	0	0	0	168.83

Table 5: Assignment computed for Problem **l** with 75 channels. The separation is tightened with a threshold of 0.1.

implementation of a heap operation, namely of *change_key*, is used. The amortized running time of this operation is still $O(\log n)$, but time savings of roughly 25% are achieved [20]. Still, the running time is prohibitive. Further performance monitoring did not reveal pathological behavior of individual routines which would recommend them for fine tuning. No computational results for the dual greedy heuristic are included here.

5.3 DSATUR With Costs

This is the best starting heuristic considered. It produces assignments of comparatively excellent quality in little running time. Running this heuristic for some random starting node usually irons out the lack of a good deterministic choice for the carrier to start assigning with. Selecting 3 to 5% randomly as starting nodes will do most of the time. Quite often, the obtained frequency assignments can be further improved by MCF and Iterated 1-OPT. However, it does not seem to pay to perform an Iterated 1-OPT run for every starting node. Finally, experiments support that an aggressive choice of the separation threshold is advisable. That is, the threshold should be chosen as low as possible while maintaining feasibility for some (randomly chosen) starting points.

5.4 Iterated 1-OPT

In several cases, Iterated 1-OPT does succeed in improving over results obtained by any of the starting heuristics. Depending on the quality of the initial assignment, the improvement

Assignment	total interference	co-channel interference	adjacent-channel interference	separation violations	invalid assignments	unassigned carriers	time [secs]
Original	167.1547	137.7719	29.3828	0	0	0	—
+ (MCF 1-OPT)*	83.4942	69.2971	14.1971	0	0	0	141.92
RANDOM	1216.0486	1146.1457	69.9030	1117	0	0	1.73
+ (MCF 1-OPT)*	86.5573	70.90 31	15.6542	0	0	0	715.81
T-coloring	79.4230	65.8120	13.6109	0	0	0	3.93
+ (MCF 1-OPT)*	79.1824	65.4874	13.6950	0	0	0	661.68
DSATUR 0%	47.6638	39.9521	7.7116	0	0	0	85.65
+ (MCF 1-OPT)*	44.8170	37.1074	7.7096	0	0	0	995.12
DSATUR 1%	45.4530	38.0736	7.3794	0	0	0	399.09
+ (MCF 1-OPT)*	43.8773	36.5236	7.3538	0	0	0	663.49
DSATUR 5%	46.0968	38.4557	7.6410	0	0	0	2276.10
+ (MCF 1-OPT)*	44.9377	37.3064	7.6313	0	0	0	991.11
DSATUR 10%	45.7894	38.2648	7.5246	0	0	0	5271.92
+ (MCF 1-OPT)*	44.6267	37.1624	7.4643	0	0	0	936.36
DSATUR 25%	45.8451	38.2032	7.6419	0	0	0	10961.89
+ (MCF 1-OPT)*	44.8259	37.2771	7.5489	0	0	0	658.62

Table 6: Assignments computed for Problem **h** with 75 channels. The separation is tightened with a threshold of 0.1.

ranges from minor to huge. The running time observed is slightly inferior to a single run of the DSATUR With Costs. This may be explained by a more detailed analysis of the operations performed by either heuristic in the implementation used.

5.5 Min-Cost Flow

Considering the nature of changes MCF is capable to perform on an assignment, it does not come by surprise that improvements are typically small. The main purpose of MCF is to escape from local minima of the neighborhood structure underlying the Iterated 1-OPT heuristic. This goal is achieved often enough to recommend MCF in combination with Iterated 1-OPT. Taking into account the huge min-cost flow problems that have to be solved, the MCF-implementation shows good performance.

6 Conclusions

Interference minimization of some sort is present in several of the approaches to frequency assignment problems published so far. To our knowledge, this paper is one of the first to make overall interference minimization the objective and to report detailed computational results.

We investigated several primal heuristics. Due to their modest space requirements and acceptable to very good running times, these heuristics are suitable for industrial application. Our results show that DSATUR With Costs applied to a small percentage (3–5% is a good choice) of randomly selected carriers as starting points is a powerful starting heuristic. Iterated 1-OPT proved capable to still improve on those assignments in reasonable time. Finally, by using MCF we are able to bring in a global optimization aspect that is helpful for escaping local minima of the neighborhood structure underlying the Iterated 1-OPT heuristic.

Assignment	**k**‡	**a**‡	**f**‡	**l**‡	**h**
`DSATUR 5% + (MCF 1-OPT)*`	95.61%	98.37%	83.36%	83.60%	73.12%

Table 7: Improvement of assignment quality relative to the original interference. A '‡' appended to the instance name expresses that the original assignment is not feasible.

We were able to *drastically improve on the original assignment*. Table 7 shows the quality achieved by DSATUR 5% followed by an alternating sequence of MCF and Iterated 1-OPT. This combination of heuristics was chosen since it produces competitive results in reasonable running times which suits it well for practice.

From experiments with various other parameter settings and other rather time consuming methods such as randomized local search procedures not documented here (see [29]), we know that the best values displayed in our tables are not optimal. Improvements are not easily obtained, though.

All of our computational experiments were performed on carrier networks that stem from a `E-Plus`' cellular phone network. `E-Plus` has integrated the well-performing heuristics presented here into their software system, thereby enhancing its network-management system with respect to frequency assignment considerably.

7 Acknowledgements

This work is done in cooperation with `E-Plus` Mobilfunk GmbH, Germany. `E-Plus` most kindly supplied us with the data sets of carrier networks. In particular, we want to express our thanks to Axel Kaudewitz and Dr. Thomas Kürner for their excellent collaboration.

References

[1] Aardal, K., Hipolito, A., van Hoesel, C., Jansen, B., Roos, C., and Terlaky, T. EUCLID CALMA Radio Link Frequency Assignment Project: A branch-and-cut algorithm for the frequency assignment problem. Technical Annex T-2.2.1 A, T.U. Eindhoven RLFAP Group and T.U. Delft RLFAP Group, Feb 1995.

[2] Ahuja, R. K., Magnanti, T., and Orlin, J. B. *Network Flows*. Prentice Hall, 1992.

[3] Bellare, M., Goldreich, O., and Sudan, M. Free bits, PCPs and non–approximability—towards tight results. In *Proc. of 36th Ann. IEEE Symp. on Foundations of Comput. Sci.* (1995), IEEE Computer Society, pp. 422–431.

[4] Bovet, D. P., and Crescenzi, P. *Introduction to the Theory of Complexity*. Series in Computer Science. Prentice Hall, 1994.

[5] Brélaz, D. New methods to color the vertices of a graph. *Communication of the ACM 22*, 4 (April 1979), 169–174.

[6] Carlsson, M., and Grindal, M. Automatic frequency assignment for cellular telephones using constraint satisfaction techniques. In *Proceedings of the Tenth International Conference on Logic Programming* (1993).

[7] Castelino, D., Hurley, S., and Stephens, N. A Tabu Search Algorithm for Frequency Assignment. *Annals of Operations Research 63* (1996), 301–319.

[8] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. *Introduction to Algorithms*. The MIT Press, 1990.

[9] Costa, D. On the use of some known methods for T–colorings of graphs. *Annals of Operations Research 41* (1993), 343–358.

[10] Cozzens, M., and Roberts, F. S. T–colorings of graphs and the channel assignment problem. *Congressum Numerantium*, 35 (1982), 191–208.

[11] Cozzens, M. B., and Wang, D. The general channel assignment problem. *Congr. Numerantium 41* (1984), 115–129.

[12] Crescenzi, P., and Kann, V. A compendium of NP optimization problems. URL: http://www.nada.kth.se/nada/theory/problemlist.html, Oct. 1995.

[13] de Werra, D., and Gay, Y. Chromatic scheduling and frequency assignment. *Discrete Appl. Math. 49*, 1–3 (1994), 165–174.

[14] Duque-Antón, M., Kunz, D., and Rüber, B. Channel Assignment for Cellular Radio Using Simulated Annealing. *IEEE Transactions on Vehicular Technology 42*, 1 (Feb 1993).

[15] Fischetti, M., Lepschy, C., Minerva, G., Jacur, G. R., and Toto, E. Frequency assignment in mobil radio systems using branch-and-cut techniques. Tech. rep., Dipartimento di Matematica e Informatica, Università di Udine, Italy, 1997.

[16] Gamst, A. Some Lower Bounds for a Class of Frequency Assignment Problems. *IEEE Transactions on Vehicular Technology VT-35*, 1 (1986), 8–14.

[17] Garey, M. R., and Johnson, D. S. *COMPUTERS AND INTRACTABILITY – A Guide to the Theory of NP–Completeness*. W.H. Freeman and Company, 1979.

[18] Haberland, D. Azyklische Subdigraphenprobleme und Frequenzzuweisung im Mobilfunk. Master's thesis, Technische Universität Berlin, Fachbereich Mathematik, 1996.

[19] Hale, W. K. Frequency Assignment: Theory and Applications. In *Proceedings of the IEEE* (Dec 1980), vol. 68, IEEE, pp. 1497–1514.

[20] Hebermehl, M. Heuristische Algorithmen zur Lösung von Färbungsproblemen im Mobilfunk. Master's thesis, Technische Universität Berlin, Fachbereich Mathematik, 1996.

[21] Hurkens, C., and Tiourine, S. Upper and lower bounding techniques for frequency assignment problems. Tech. rep., Eindhoven University of Technology, Netherlands, July 1995.

[22] Hurley, S., Smith, D., and Thiel, S. FASoft: A Software System for Frequency Assignment. URL: http://www.cs.cf.ac.uk/User/Steve.Hurley/freq_pub.htm, 1997. Submitted to *Radio Science*.

[23] Jünger, M., Martin, A., Reinelt, G., and Weismantel, R. Quadratic 0/1 optimization and a decomposition approach for the placement of electronic circuits. *Mathematical Programming 63* (1994), 257–279.

[24] Löbel, A. *MCF Version 1.0 – A network simplex implementation*, 1997. Available for academic use free of charge via WWW at URL: `http://www.zib.de/Optimization`.

[25] Mouly, M., and Pautet, M.-B. *The GSM System for Mobile Communications*. Cell & Sys, France, 1992.

[26] Näher, S., and Uhrig, C. *The LEDA User Manual Version R 3.3*. Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany, 1996.

[27] Papadimitriou, C. H. *Computational Complexity*. Addison-Wesley, 1994.

[28] Plehn, J. Applied Frequency Assignment. In *Proceedings of the IEEE Vehicular Technology Conference* (1994), IEEE.

[29] Schneider, K. Frequenzzuweisung im Mobilfunk mittels lokaler Suche. Master's thesis, Technische Universität Berlin, Fachbereich Mathematik, 1997.

[30] Tiourine, S., Hurkens, C., and Lenstra, J. K. An overview of algorithmic approaches to frequency assignment problems. Tech. rep., Eindhoven University of Technology, Netherlands, Aug 1995. CALMA Project.