Optimal channel allocation for several types of cellular radio networks

Thomas Niessen*

Institut für Statistik, RWTH Aachen, 52056 Aachen, Federal Republic of Germany

Received 11 September 1995; received in revised form 10 August 1996; accepted 5 March 1997

Abstract

The channel allocation problem in cellular radio networks is formulated as an optimization problem on set-valued graph colorings. Thereby a common model is found for some optimization criteria that appeared formerly to be distinct. The optimization problem is then transformed to a weighted graph coloring problem. Several efficient algorithms for the weighted coloring on special classes of graphs are known. They are investigated, when they are applied to an instance resulting from a transformation of a channel allocation problem.

1. Introduction

The wireless part of cellular radio network can be described as follows. The service area is divided into cells each having a base station. Mobile subscribers communicate only with the base station of their cell by transmitting radio signals at certain channels. In order to decide which channels a base station may use, the network is often described by the so-called interference graph. The vertices of this graph correspond to the base stations and two vertices are adjacent if interference can occur, when the same channel is used in the corresponding cells simultaneously. Interferences between different channels are thereby neglected. Most existing cellular radio networks operate with fixed channel allocation (FCA), that is, to each base station a set of channels is assigned during a planning process. So, the channel allocations are naturally described by generalized colorings of the interference graph. To compare the utilization of different channel allocations several criteria are suggested in the literature, and thus, there is a whole class of channel assignment problems. In Section 2 we present a general channel assignment problem (CAP) including some of the most common

---

* This work was supported by the Deutsche Forschungsgemeinschaft under grant Ma 1184/4-2.
* E-mail: niessen@stochastik.rwth-aachen.de.
optimality criteria. Moreover, we will see that it contains some well-known optimization problems as well. In the third section, we show that the CAP can be transformed to a known weighted coloring problem restricted to so-called gapless colorings. Thereafter, we present a condition depending on the optimization criterion that allows to neglect this additional restriction. Since most of the examples satisfy the condition, the CAP can be solved for many instances by means of efficient algorithms for the weighted coloring problem. Such algorithms are known for special classes of graphs (interval graphs, cocomparability graphs, and comparability graphs, respectively). It is thereby possible to compare the result of heuristical strategies to the optimal solutions for a wide range of graphs. In Section 4 we study the performance of these algorithms, when they are applied to instances resulting from the transformation of Section 3. The graphs to which these algorithms are applied are compositions of the interference graph and a complete graph. Due to their special structure, we are able to modify the algorithm for cocomparability graphs and obtain thereby a better time performance.

The terminology used in this paper is as follows. Let G be an undirected graph without loops and multiple edges. The vertex set and the edge set of G are \( V(G) \) and \( E(G) \), respectively. An edge joining the vertices \( u, v \in V(G) \) is denoted by \( uv \). For a positive integer \( N \), an \( N \)-set coloring \( \varphi \) of \( G \) assigns to each vertex \( x \in V(G) \) a set \( \varphi(x) \subseteq \{1, 2, \ldots, N\} \) such that \( \varphi(x) \cap \varphi(y) = \emptyset \) for all \( xy \in E(G) \). The number of colors assigned to a vertex \( x \) by \( \varphi \) is \( a_x(\varphi) \). If no ambiguity may occur, we write \( a_x \) instead of \( a_x(\varphi) \).

In our context, the graph \( G \) is the interference graph of a cellular radio network. The channels are the colors \( 1, 2, \ldots, N \), and an \( N \)-set coloring \( \varphi \) assigns to each vertex \( x \) the set \( \varphi(x) \) of channels that can be used by the corresponding base station without interference.

If \( G \) is a graph and \( p \) is a positive integer, then \( \overline{G} \) is the complement of \( G \) and \( pG \) is a graph consisting of \( p \) disjoint copies of \( G \).

An orientation of \( G \) is a directed graph \( D \) with vertex set \( V(D) = V(G) \), where each edge \( uv \in E(G) \) is replaced either by the arc \( (u, v) \) or by the arc \( (v, u) \). The arc set of a directed graph is denoted by \( A(D) \).

For two graphs \( G \) and \( H \) we denote by \( G[H] \) the composition (or lexicographic product) defined by \( V(G[H]) = V(G) \times V(H) \) and two vertices \( (g, h) \) and \( (g', h') \) are adjacent in \( G[H] \) if and only if either \( g \) is adjacent to \( g' \) in \( G \) or \( g = g' \) and \( h \) is adjacent to \( h' \) in \( H \). Similarly we denote by \( D[E] \) the composition of two directed graphs \( D \) and \( E \) defined by \( V(D[E]) = V(D) \times V(E) \) and there is an arc from \( (d, e) \) to \( (d', e') \) in \( D[E] \) if and only if either \( (d, d') \in A(D) \) or \( d = d' \) and \( (e, e') \in A(E) \).

A (flow) network \( \mathcal{N} = (V, A, s, t, \text{cap}) \) consists of a set of vertices \( V \), a collection of arcs \( A \), two distinguished vertices \( s \) (source) and \( t \) (sink), and a capacity function \( \text{cap} \) that assigns to each vertex and to each arc a nonnegative number. A flow \( f \) in the network \( \mathcal{N} \) assigns to every arc \( (u, v) \in A \) a value \( f(u, v) \) such that the following conditions are satisfied:

\[
0 \leq f(u, v) \leq \text{cap}(u, v) \quad \text{for all } (u, v) \in A,
\]
and
\[ \sum_{v : (u,v) \in A} f(u,v) = \sum_{v : (v,u) \in A} f(v,u) \leq \text{cap}(u) \text{ for all } u \in V - \{s,t\}. \]

The value of a flow $f$ is $|f| = \sum_{u,v : (u,v) \in A} f(u,v)$.

Note that in the standard model the capacity function assigns capacities only to arcs. Vertex capacities are here permitted to achieve simpler descriptions of some algorithms. They can be avoided by a well-known technique (see [6, p. 23]).

Occasionally, we will consider a network $\mathcal{N} = (V,A,s,t,\text{cap})$ with a cost function $\text{cost}$ that assigns to every arc $(u,v) \in A$ nonnegative costs $\text{cost}(u,v)$. The costs of a flow $f$ in $\mathcal{N}$ are as usual defined by $\text{COST}(f) = \sum_{(u,v) \in A} \text{cost}(u,v)f(u,v)$.

We will assume some familiarity with several classes of graphs, in particular interval graphs, comparability graphs and cocomparability graphs. All unexplained terms and properties can be found in [12].

2. Model assumptions

We first introduce our model of the channel allocation problem. The formulation is derived from a stochastic model (see Example 2 below).

Let $G$ be a graph and assume that to each vertex $x \in V(G)$ belongs a monotone decreasing function $f_x : \{0,1,\ldots,N\} \rightarrow \mathbb{Q}^+$, where $\mathbb{Q}^+$ denotes the set of nonnegative rational numbers. Using these functions we assign to every $N$-set coloring $\phi$ of $G$ the value
\[ \mathcal{F}(\phi) = \sum_{x \in V(G)} f_x(a_x(\phi)). \]

The channel allocation problem is the following.

CAP $(N)$ (Channel Allocation Problem)

**Instance:** A pair $(G,\{f_x | x \in V(G)\})$.

**Problem:** Determine an $N$-set coloring minimizing $\mathcal{F} = \sum_{x \in V(G)} f_x$.

**Examples.** (1) (Requirement Problem) Let $G$ be the interference graph of a cellular radio network and suppose that we have to allocate at least $n_x$ channels to each cell $x \in V(G)$.

This problem appears frequently in the literature, if fixed channel allocation is performed in the following way. The traffic intensities in all cells are estimated, and thereafter a decision is made on the number of channels each base station needs. Another setting in which the problem occurs is a dynamical channel allocation strategy called Maximum Packing [5, 16].

Let for each $x \in V(G)$ and for every nonnegative integer $j$
\[ f_x(j) = \max\{n_x - j, 0\}. \]
Since an $N$-set coloring $\varphi$ satisfies the requirement of each cell if and only if $F(\varphi) = 0$, we see that the requirement problem is covered by $\text{CAP}(N)$.

(2) (Stochastic model) If a mobile subscriber aims to communicate with a base station, but cannot be served, then it is called blocked. Channel allocations can be ranked by their overall blocking probability allowing a very careful analysis of their performance over a long period. Therefore the traffic in all cells is described stochastically and the probability that a mobile subscriber is blocked is computed for each cell. Let $B_x(j)$ denote the blocking probability in cell $x$, if $j$ channels are available. The overall blocking probability of an $N$-set coloring $\varphi$ assigning $a_x$ channels to cell $x$ is calculated by

$$B(\varphi) = \sum_{x \in V(G)} w_x B_x(a_x)$$

with positive weights $w_x$ (expressing the relative importance to satisfying the demand in $x$) such that $\sum_{x \in V(G)} w_x = 1$.

Letting $f_x(j) = w_x B_x(j)$ for every $x \in V(G)$ and every nonnegative integer $j$ we obtain immediately an instance of the $\text{CAP}(N)$.

The most important stochastic model is the following [16, 19]. Arriving calls of mobile subscribers in each cell are described by independent Poisson processes with arrival rate $v_x$, and the corresponding service times by independent random variables having a distribution with finite expectation $\alpha_x$. Then $\lambda_x = v_x \alpha_x$ is the traffic load in $x$, and in the stationary state the blocking probability $B_x(a_x)$ in cell $x$ is obtained by the Erlang-B formula [25]

$$B_x(a_x) = E_x(\lambda_x), \text{ where } E_x(\lambda) = \frac{\lambda^j/j!}{\sum_{i=0}^{\infty} \lambda^i/i!} \text{ for } j \in \mathbb{N}_0 \text{ and } \lambda \geq 0.$$ 

The weights $w_x$ can be chosen proportional to $\lambda_x$ [19, 28].

(3) (Maximizing the number of allocated channels) If the traffic load is high for all base stations, then one can expect that the traffic that can be carried is directly proportional to the number of allocated channels $\sum_{x \in V(G)} a_x$. Maximizing this sum is equivalent to minimizing $F$ where $f_x$ is defined for every $x \in V(G)$ by $f_x(j) = N - j$ for all nonnegative integers $j$.

(4) (Models with local performance guarantee) A set coloring that minimizes $F = \sum_{x \in V(G)} f_x$ (for instance, in Example 2 or 3) may have the drawback that certain base stations receive only a few channels, and so the local performance is bad. This can be overcome by restricting the attention to $N$-set colorings allocating at least $n_x$ channels to each cell $x$, where $n_x$ is chosen such that good local performance is guaranteed. To see how this can be formulated in our model, we first choose a rational $\varepsilon$ with $0 < \varepsilon < 1$ satisfying

$$\varepsilon \sum_{u \in V(G)} f_u(n_u) < \min_{n_u > 0} \{ \varepsilon f_x(n_x) + (1 - \varepsilon) \}$$

(1)
and
\[ \varepsilon(f_x(n_x) - f_x(n_x + 1)) \leq 1 - \varepsilon \] (2)
for every \( x \) with \( n_x > 0 \). This choice is possible, since, if \( \varepsilon \to 0 \), the left hand sides of (1) and (2) tend to 0, whereas the right hand sides tend to 1.

Let now for each \( x \in V(G) \)
\[ f^*_x(j) = \begin{cases} \varepsilon f_x(n_x) + (1 - \varepsilon)(n_x - j) & \text{for } j < n_x, \\ \varepsilon f_x(j) & \text{for } j \geq n_x. \end{cases} \]
It is easy to see that \( f^*_x \) is monotone decreasing for every \( x \in V(G) \). Thus \( (G, \{ f^*_x \mid x \in V(G) \}) \) is also an instance of \( \text{CAP}(N) \), where \( \mathcal{F}^* = \sum_{x \in V(G)} f^*_x \) is to minimize. Suppose now that \( \varphi \) allocates at least \( n_x \) channels for every \( u \in V(G) \). Then \( \mathcal{F}^*(\varphi) \leq \varepsilon \sum_{u \in V(G)} f_u(n_u) \). On the other hand, if \( \varphi \) assigns to some \( v \in V(G) \) less than \( n_x \) channels, then \( \mathcal{F}^*(\varphi) \geq \varepsilon f_x(n_{x} - 1) = \varepsilon f_x(n_x) + (1 - \varepsilon) \). Therefore it follows from (1) that \( \mathcal{F}^* \) will be minimized by an \( N \)-set coloring \( \varphi \) that allocates \( n_x \) channels to each \( x \in V(G) \), if such an \( N \)-set coloring exists. Inequality (2) will be used later.

Before we discuss the relations of \( \text{CAP}(N) \) to other problems, we note that the number of colors does not belong to the instance of the problem. We prefer this formulation for two reasons. First, only this version can be related to a lot of other problems, since the input length is independent of \( N \) if and only if \( N \) is fixed. Second, in real world applications the number of channels is in fact a constant. Nevertheless, this constant is usually large and we will see in Section 4 that the running time of some algorithms heavily depends on \( N \).

It is well known that even some special cases of \( \text{CAP}(1) \) are more general than the independent set problem and thus \( \text{CAP}(1) \) is NP-hard in general. Moreover, \( \text{CAP}(1) \) is even hard to approximate, unless \( P = NP \) [19]. In fact, the natural one-to-one correspondence of the 1-set colorings and the independent sets of \( G \) shows that \( \text{CAP}(1) \) is equivalent to the maximum weighted independent set problem. To see this let \( (G, \{ f_x \mid x \in V(G) \}) \) be any instance of \( \text{CAP}(1) \). Assign to each vertex \( x \) of \( G \) the weight \( f_x(0) - f_x(1) \). Now solving the \( \text{CAP}(1) \) for this instance is equivalent to finding an independent set with maximum weight. Conversely, if \( G \) is a graph with nonnegative rational weights \( W(x) \), then we let \( f_x(0) = W(x) \) and \( f_x(1) = 0 \) and hence \( (G, \{ f_x \mid x \in V(G) \}) \) is an instance of the \( \text{CAP}(1) \).

It is also well-known that the requirement version of \( \text{CAP}(N) \) contains the \( N \)-coloring problem. Therefore let \( f_x(0) = 1 \) and \( f_x(j) = 0 \) for \( j = 1, \ldots, N \) for all \( x \in V(G) \). Then \( G \) is \( N \)-colorable if and only if \( \mathcal{F}(\varphi) = 0 \) for some \( N \)-set coloring \( \varphi \). Hence \( \text{CAP}(N) \) is NP-hard for \( N \geq 3 \). The \( \text{CAP}(N) \) for these instances is just the maximum \( N \)-colorable induced subgraph problem, which is for every \( N \geq 1 \) as hard to approximate as the maximum independent set problem [22].

The requirement version of the \( \text{CAP}(N) \) is closely related to one of the problems studied by Grötschel et al. [14]. Given a graph \( G \) and a nonnegative integer \( n_x \) for
every \( x \in V(G) \) the problem is the following: find independent sets \( S_1, \ldots, S_t \) and positive integers \( s_1, \ldots, s_t \) such that for all \( x \in V(G) \), \( \sum_{i : x \in S_i} s_i \geq n_x \) holds, and such that \( \sum_{i=1}^{t} s_i \) is as small as possible. This problem is equivalent to finding the smallest integer \( N^* \) and an \( N^* \)-set coloring \( \varphi \) solving the requirement problem for \( G \) and all \( n_x \).

This can be seen as follows. If \( S_1, \ldots, S_t \) and \( s_1, \ldots, s_t \) are like above, then we obtain an \((\sum_{i=1}^{t} s_i)\)-set coloring by assigning the colors \( 1, \ldots, s_1 \) to the vertices in \( S_1 \), the colors \( s_1 + 1, \ldots, s_1 + s_2 \) to the vertices in \( S_2 \), and so on. Then it holds \( \sum_{i : x \in S_i} s_i \geq n_x \) for every \( x \in V(G) \), and so the set coloring fulfills the requirement problem. Therefore we have \( \sum_{i=1}^{t} s_i \geq N^* \). Conversely, if an \( N^* \)-set coloring \( \varphi \) solves the requirement problem, consider the sequence \((T_i)_{i=1,\ldots,N^*}\) where \( T_i = \{ x \in V(G) | i \in \varphi(x) \} \). Let \( S_1, \ldots, S_t \) be the distinct, non-empty sets in this sequence, and denote by \( s_i \) the number of times that \( S_i \) appears. Clearly, each \( S_i \) is an independent set of \( G \) and for every \( x \in V(G) \) it holds \( \sum_{i : x \in S_i} s_i = |\varphi(x)| \geq n_x \). So \( N^* \geq \sum_{i=1}^{t} s_i \geq N^* \), and hence equality follows, as desired. An algorithm solving the above problem for perfect graphs is presented in [14]. But, though the algorithm is polynomial, it cannot be recommended for practical purposes [14].

3. A relation to a weighted coloring problem

In this section we show how the \( \text{CAP}(N) \) is related to a weighted coloring problem.

Let \( G \) be a graph and let \( N \) be a positive integer. A function \( \psi : U \rightarrow \{1, \ldots, N\} \), where \( U \subseteq V(G) \), is a partial \( N \)-coloring of \( G \), if \( \psi(u) \neq \psi(v) \) for all vertices \( u, v \in U \) with \( uv \in E(G) \). Let \( K_N \) denote the complete graph of order \( N \) with vertex set \( \{1, \ldots, N\} \). A partial \( N \)-coloring \( \psi \) of \( G[K_N] \) that colors \( U \subseteq V(G[K_N]) \) is called gapless, if \( (x,i) \in U \) implies \( (x,j) \in U \) for all \( j < i \). In the following, a pair \((G, W)\) consists of a graph \( G \) and a weight function \( W \) that assigns to each vertex of \( G \) a nonnegative rational weight \( W(x) \).

Now we define the weighted coloring problem mentioned above.

**MWCP \( (N) \) (Maximum Weighted Coloring Problem):**

**Instance:** A pair \((G, W)\).

**Problem:** Find a partial \( N \)-coloring \( \psi \) coloring \( U \subseteq V(G) \) such that \( \sum_{u \in U} W(u) \) is maximum.

The following three lemmas establish a relation of \( \text{CAP}(N) \) and \( \text{MWCP}(N) \).

**Lemma 1.** Let \( \psi \) be a partial \( N \)-coloring of \( G[K_N] \) coloring the set \( U \). Define \( \varphi : V(G) \rightarrow 2^{\{1, \ldots, N\}} \) by \( \varphi(x) = \{ \psi(x,i) | (x,i) \in U \} \). Then \( \varphi \) is an \( N \)-set coloring of \( G \), and we call it the \( N \)-set coloring derived from \( \psi \).

**Proof.** Let \( x \) and \( y \) be distinct vertices of \( G \) with \( \varphi(x) \cap \varphi(y) \neq \emptyset \). Then there exists \( (x,i), (y,j) \in U \) with \( \psi(x,i) = \psi(y,j) \). Therefore, \( (x,i) \) and \( (y,j) \) are non-adjacent in \( G[K_N] \), and hence \( x \) and \( y \) are non-adjacent in \( G \). \( \square \)
Lemma 2. For every N-set coloring $\varphi$ there exists a gapless partial N-coloring $\psi$ of $G[K_N]$ such that $\varphi$ is the N-set coloring derived from $\psi$.

Proof. Let $\varphi$ be an N-set coloring of $G$ with $\varphi(x) = \{c_{x,1}, \ldots, c_{x,a_x}\}$ for every $x \in V(G)$. Define $U = \{(x,i) \mid x \in V(G), 1 \leq i \leq a_x\}$ and $\psi: U \rightarrow \{1, \ldots, N\}$ by $\psi(x,i) = c_{x,i}$. Clearly $\psi$ is a gapless partial N-coloring of $G[K_N]$ and $\varphi$ is the N-set coloring derived from $\psi$. $\square$

Lemma 3. Let $(G, \{f_x \mid x \in V(G)\})$ be an instance for the CAP(N). Assign to each vertex $(x,i)$ of $G[K_N]$ the weight $W(x,i) = f_x(i - 1) - f_x(i)$. If $\psi$ is a gapless partial N-coloring of $G[K_N]$ coloring the set $U$ and if $\varphi$ is the N-set coloring derived from $\psi$, then $F(\varphi) = \sum_{x \in V(G)} f_x(0) - \sum_{(x,i) \in U} W(x,i)$.

Proof. Since $\psi$ is a gapless partial N-coloring of $G[K_N]$, we have $\{(i \mid (x,i) \in U\} = \{1, \ldots, a_x\}$ for every $x \in V(G)$, where $a_x = |\varphi(x)|$. Thus

$$F(\varphi) = \sum_{x \in V(G)} f_x(0) - \sum_{x \in V(G)} \sum_{i=1}^{a_x} (f_x(i - 1) - f_x(i))$$

$$= \sum_{x \in V(G)} f_x(0) - \sum_{(x,i) \in U} W(x,i). \quad \square$$

Lemmas 1–3 indicate that solving the CAP(N) for $(G, \{f_x \mid x \in V(G)\})$ can be done by solving the instance $(G[K_N], W)$ of MWCP(N) restricted to gapless partial N-colorings. Though this additional restriction seems to be hard to tackle in general, we will show that it can be neglected for most of our applications. Therefore call a function $f: \{0, \ldots, N\} \rightarrow \mathbb{Q}^+$ convex if and only if the sequence $\{f(i - 1) - f(i)\}_{i=1}^{N}$ is monotone decreasing.

Theorem 4. Let $(G, \{f_x \mid x \in V(G)\})$ be an instance of the CAP(N) where $f_x$ is convex for every $x \in V(G)$. If $\psi$ is a solution of the MWCP(N) with instance $(G[K_N], W)$, where $W(x,i) = f_x(i - 1) - f_x(i)$ for all $x \in V(G)$ and $i \in \{1, \ldots, N\}$, then the N-set coloring $\varphi$ derived from $\psi$ is a solution for the CAP(N).

Proof. For every $x \in V(G)$ let $\{(i \mid (x,i) \in U)\} = \{j_{x,1}, \ldots, j_{x,a_x}\}$ with $j_{x,1} \leq j_{x,2} \leq \cdots \leq j_{x,a_x}$. Define $U^* = \{(x,i) \mid x \in V(G), 1 \leq i \leq a_x\}$ and $\psi^*: U^* \rightarrow \{1, \ldots, N\}$ by $\psi^*(x,i) = \psi(x,j_{x,i})$. It is easy to see that $\psi^*$ is a gapless partial N-coloring of $G[K_N]$ having $\varphi$ as derived N-set coloring. Since $f_x$ is convex for every $x \in V(G)$ and $i \leq j_{x,i}$, we have $W(x,i) \geq W(x,j_{x,i})$ for all $(x,i) \in U^*$, and thus

$$\sum_{(x,i) \in U^*} W(x,i) \geq \sum_{x \in V(G)} \sum_{i=1}^{a_x} W(x,j_{x,i}) = \sum_{(x,i) \in U} W(x,i).$$

Therefore $\psi^*$ is also a solution to the MWCP(N). Now it follows from the previous lemmas that $\varphi$ is a solution for the CAP(N). $\square$
Next we show convexity for the \( f_X \) in the examples presented above.

**Examples (continued).** (1) Here we have

\[
f_X(i - 1) - f_X(i) = \begin{cases} 1 & \text{for } i = 1, \ldots, n_X - 1, \\ 0 & \text{for } i = n_X, \ldots, N \end{cases}
\]

and hence \( f_X \) is convex.

(2) It is shown in [15, 20] that \( E_j(\lambda) \) considered as a function of \( j \) is convex for every \( \lambda \geq 0 \). Hence \( f_X \) with \( f_X(j) = w_X E_j(\lambda_x) \) is convex for every \( x \in G \).

(3) We have \( f_X(i - 1) - f_X(i) = 1 \) for \( i = 1, \ldots, N \), and hence \( f_X \) is convex.

(4) It holds

\[
f^*_X(i - 1) - f^*_X(i) = \begin{cases} 1 - \varepsilon & \text{for } i = 1, \ldots, n_x, \\ \varepsilon(f_X(i - 1) - f_X(i)) & i = n_x + 1, \ldots, N. \end{cases}
\]

Suppose now that \( f_X \) is convex. If \( n_x = 0 \), we have \( f^*_X = f_X \) and hence \( f^*_X \) is convex. If \( n_x > 0 \), then \( f^*_X \) is convex if and only if

\[
f^*_X(n_x - 1) - f^*_X(n_x) \geq f^*_X(n_x) - f^*_X(n_x + 1).
\]

Since this is equivalent to (2), \( f^*_X \) is convex in all cases, if \( f_X \) is convex.

Obviously, MWCP(\( N \)) is an NP-hard problem, but it is well investigated and tractable for some classical subfamilies of perfect graphs. In view of Theorem 4 we are interested in graphs \( G \) for which \( G[K_n] \) belongs to one of these families.

The following two lemmas are well-known. We include their proofs for later reference.

**Lemma 5.** Let \( \mathcal{C} \) denote the family of comparability graphs or cocomparability graphs, respectively. Then \( G[H] \) belongs to \( \mathcal{C} \) if and only if \( G \) and \( H \) belong to \( \mathcal{C} \).

**Proof.** Since \( G \) and \( H \) are induced subgraphs of \( G[H] \) and since being a comparability or cocomparability is a hereditary property, it follows immediately that \( G \) and \( H \) belong to \( \mathcal{C} \), if \( G[H] \) belongs to \( \mathcal{C} \).

Let now \( G \) and \( H \) belong to \( \mathcal{C} \). If \( \mathcal{C} \) is the family of comparability graphs, then \( G \) and \( H \) have transitive orientations \( D_G \) and \( D_H \), respectively. It is easy to see that \( D_G[D_H] \) is a transitive orientation of \( G[H] \), and so \( G[H] \) is also in \( \mathcal{C} \). Similarly, if \( \mathcal{C} \) is the family of cocomparability graphs, then \( \overline{G} \) and \( \overline{H} \) have transitive orientations \( D_{\overline{G}} \) and \( D_{\overline{H}} \), respectively. Observing that \( D_{\overline{G}}[D_{\overline{H}}] \) is a transitive orientation of \( G[H] = \overline{G[H]} \) we find that \( G[H] \) belongs to \( \mathcal{C} \). \( \square \)

The corresponding statement for perfect graphs is known to be true [18], but it does not hold for chordal graphs and interval graphs (to see this let \( P_3 \) denote a path with three vertices and consider \( P_3[P_3] \)).
Lemma 6. Let $\mathcal{C}$ denote the family of chordal graphs or interval graphs, respectively. Then $G[K_N]$ belongs to $\mathcal{C}$ if and only if $G$ belongs to $\mathcal{C}$.

Proof. As in the proof of Lemma 5 we obtain that $G$ belongs to $\mathcal{C}$, if $G[K_N]$ belongs to $\mathcal{C}$.

Let now $G$ be a chordal graph and let $((v_0, i_0), (v_1, i_1), \ldots, (v_l, i_l))$ denote a cycle of length $l + 1 \geq 4$ in $G[K_N]$. If $v_0 = v_1 = \cdots = v_l$, then there is a chord joining $(v_0, i_0)$ and $(v_2, i_2)$. If $v_i \neq v_j$ for all $0 \leq i < j \leq l$, then there is a chord corresponding to a chord in the cycle $(v_0, v_1, \ldots, v_l)$ of $G$. In all other cases we can find an index $j$ such that $v_j \neq v_{j+1}$ and $v_{j+1} = v_{j+2}$ (where indices are taken modulo $l + 1$). Hence there is a chord joining $(v_j, i_j)$ and $(v_{j+2}, i_{j+2})$ in $G[K_N]$.

An alternative proof for chordal graphs giving additional information can be done using perfect elimination schemes. Therefore it is only necessary to observe that $(v_1, 1), (v_1, 2), \ldots, (v_1, N), (v_2, 1), \ldots, (v_n, N)$ is a perfect elimination scheme of $G[K_N]$ if $v_1, v_2, \ldots, v_n$ is one of $G$.

If $\mathcal{C}$ is the family of interval graphs, the statement of the lemma follows from the results already shown for cocomparability graphs and chordal graphs, since $\mathcal{C}$ is just the intersection of these families. A direct proof which yields in addition an interval representation of $G[K_N]$ is also quite easy. Therefore verify that $\{I(v, i) \mid (v, i) \in V(G[K_N])\}$ with $I(v, i) = I_v$ for every $v \in V(G)$ and $1 \leq i \leq N$ is an interval representation of $G[K_N]$, if $\{I_v \mid v \in V(G)\}$ is one of $G$. □

Note that the statement of Lemma 6 is also valid for circular-arc graphs, but it does not hold for cocircular-arc graphs or bipartite graphs.

4. Some algorithms for the MWCP(N)

In this section we will discuss some known algorithms solving the MWCP(N) on some families of graphs. More precisely, we will investigate these algorithms, when they are applied to an instance $(G[K_N], W)$ resulting from an instance $(G, \{f_x \mid x \in V(G)\})$ of CAP(N) by an application of Theorem 4. Therefore we assume throughout the whole section that every $f_x$ is convex. Furthermore, $n$ and $e$ denote order and size of the input graph.

In [23] Sarrafzadeh and Lou present an algorithm for maximum weighted $N$-covering on comparability graphs. This covering problem is the MWCP(N) applied to the complement of the graph, i.e., the problem is to find in a weighted graph a collection of $N$ vertex disjoint cliques such that the sum of the weights of the vertices belonging to the cliques is maximum. The running time of their algorithm is $O(n^2)$. Applying Theorem 4 and Lemma 5, we find that on cocomparability graphs CAP(N) can be solved in $O(n^3 n^2)$ time. During the implementation of this algorithm it turned out that the performance is rather bad. This is mainly due to the size of the network flow problem to which the instance of CAP(N) is transformed. Introducing some
refinements, which are possible by the special structure of the composition \( G[K_N] \), we were able to improve the performance significantly. In order to explain these refinements, we present first a short description of the algorithm.

Let \( G \) be a comparability graph with \( n \) vertices and \( e \) edges. Starting from \((G,W)\) a network \( \mathcal{N}_D \) is constructed as follows. First obtain a transitive orientation \( D \) of \( G \). This can be done in \( O(n^2) \) time [24]. Thereafter add to \( D \) two distinguished vertices \( s \) and \( t \), the arcs \((s,v)\) and \((v,t)\) for every \( v \in V(G) \), and the arc \((s,t)\). The resulting directed graph is also transitive. Let \( V \) and \( A \) denote its vertex set and arc set, respectively. The network \( \mathcal{N}_D \) is \((V,A,s,t,\text{cap})\), where the capacity function \( \text{cap} \) assigns the value 1 to every vertex and every arc. Next a cost function \( \text{cost} \) is introduced. Therefore begin-weights \( \lambda_b(v) \) and end-weights \( \lambda_e(v) \) are defined for every \( v \in V \). Let first \( \lambda_b(s) = \lambda_e(s) = 0 \). Then sort the vertices of \( D \) topologically in \( O(n + e) \) time [1]. According to the topological sorting, the begin- and end-weights are computed by \( \lambda_b(v) = \max \{ \lambda_e(x) \mid (x, v) \in A \} \) and \( \lambda_e(v) = \lambda_b(v) + W(v) \) for all \( v \in V(G) \). Finally, let \( \lambda_{\max} = \max \{ \lambda_b(v) \mid v \in V(G) \} \) and \( \lambda_b(t) = \lambda_e(t) = \lambda_{\max} \). These computations can be done in \( O(n + e) \) time [23]. The cost function \( \text{cost} : A \to \mathbb{Q}^+ \) is now defined by \( \text{cost}(u,v) = \lambda_b(v) - \lambda_e(u) \) for every arc \((u,v)\in A\). Thus starting from \((G,W)\), the network \( \mathcal{N}_D \) and the cost function \( \text{cost} \) can be obtained in \( O(n^2) \) time [23].

The algorithm proceeds with the determination of an integer-valued minimum cost flow \( f \) in \( \mathcal{N}_D \) with \( |f| = N \). The correctness of the algorithm follows immediately from the next lemma.

**Lemma 7** (Sarrafzadeh and Lou [23]). There is a one-to-one correspondence between the integer-valued flows \( f \) in \( \mathcal{N}_D \) with \( |f| = N \) and the sets \( \{C_1, \ldots, C_N\} \) of \( N \) vertex disjoint cliques of \( G \). Moreover, for every flow \( f \) and its corresponding set \( \{C_1, \ldots, C_N\} \) it holds \( \sum_{i=1}^{N} \sum_{u \in V(C_i)} W(u) = N\lambda_{\max} - \text{COST}(f) \).

Since \( N\lambda_{\max} \) is a constant, based on Lemma 7, minimizing \( \text{COST}(f) \) in \( \mathcal{N}_D \) is equivalent to maximizing \( \sum_{i=1}^{N} \sum_{u \in V(C_i)} W(u) \) in \( G \). Thus, a solution to a maximum weighted \( N \)-covering problem on a weighted comparability graph can be obtained by finding a solution to a minimum cost network flow problem. Solving this network flow problem can be done in \( O(Ne \log_2 \frac{n+e}{N} n) \) time [26] (Theorem 8.13). Hence the running time of the algorithm is \( O(n^2 + Ne \log_2 \frac{n+e}{n} n) \) (or \( O(Nn^2) \) as claimed in [23]). Since it takes \( O(n^2) \) time to compute the complement of a graph, MWCP(N) on cocomparability graphs can also be solved in \( O(n^2 + Ne \log_2 \frac{n+e}{n} n) \) time.

Suppose now that we apply Theorem 4 and the algorithm of Sarrafzadeh and Lou to an instance \((G, \{f_x \mid x \in V(G)\})\) of CAP(N), where \( G \) is a cocomparability graph. We first compute \( G \) in \( O(n^2) \) time, and then we find a transitive orientation \( D \) of \( G \) in \( O(n^2) \) time. A transitive orientation of \( G[K_N] \) is \( D[NK_1] \) (see the proof of Lemma 6), and hence completely described by \( D \) and \( N \). Though we do not need to compute and store \( D[NK_1] \), the running time of the network flow algorithm will depend on the order \( Nn \) and size \( N^2 \tilde{e} \) of this directed graph, where \( \tilde{e} \) is the size of \( G \). Thus it is \( O(N^2 n^2 + N^3 \tilde{e} \log_2 \frac{n+e}{n} (Nn)) \).
Let \((V,A)\) denote the directed graph of \(\mathcal{N}_{D[\mathbf{NK}_1]}\). In order to decrease the size of
the network we show that \((V,A)\) can be replaced by its transitive reduction \((V,A_\text{tr})\).
Like above we determine the transitive orientation \(D\) of \(\overline{G}\). Thereafter, we compute
the transitive reduction \(D_\text{tr}\) of \(D\). The transitive reduction of \(D[\mathbf{NK}_1]\) is \(D_\text{tr}[\mathbf{NK}_1]\). We
add to \(D_\text{tr}[\mathbf{NK}_1]\) again the vertices \(s\) and \(t\), but then only the arcs from \(s\) to vertices
with indegree 0 and the arcs from vertices with outdegree 0 to \(t\) are added. It is easy
to see that we obtain just \((V,A_\text{tr})\) in this way. Now we proceed like in the algorithm
of Sarrafzadeh and Lou with \((V,A)\) replaced by \((V,A_\text{tr})\). The resulting network will
be denoted by \(\mathcal{N}_{D[\mathbf{NK}_1]_\text{tr}}\). Note that the begin- and end-weights are the same as above,
since all weights are nonnegative. Therefore, the cost function obtained here is the
restriction from \(A\) to \(A_\text{tr}\) of the cost function obtained above.

**Lemma 8.** There is a one-to-one correspondence between the integral flows \(f\) in
\(\mathcal{N}_{D[\mathbf{NK}_1]_\text{tr}}\) with \(|f| = N\) and the sets \(\{C_1, \ldots, C_N\}\) of \(N\) vertex disjoint, maximum in-
dependent sets of \(G[\mathbf{K}_N]\). Moreover, for every flow \(f\) and its corresponding set
\(\{C_1, \ldots, C_N\}\) it holds
\[
\sum_{i=1}^{N} \sum_{(u,j) \in V(C_i)} W(u,j) = N \lambda_{\text{max}} - \text{COST}(f).
\]

**Proof.** Since the capacity function assigns the value 1 to all vertices, every flow \(f\)
in \(\mathcal{N}_{D[\mathbf{NK}_1]_\text{tr}}\) with \(|f| = N\) corresponds to a unique collection of \(N\) directed paths from
\(s\) to \(t\) in \((V,A_\text{tr})\) such that (except of \(s\) and \(t\), of course) each vertex appears in at
most one of these paths. Since \(\overline{G}[\mathbf{NK}_1]\) is a comparability graph, the inner vertices
of these paths induce vertex disjoint cliques in \(\overline{G}[\mathbf{NK}_1]\). Moreover, every path from
\(s\) to \(t\) in \((V,A_\text{tr})\) corresponds to a unique maximum clique in \(\overline{G}[\mathbf{NK}_1]\) or to a unique
maximum independent set of \(G[\mathbf{K}_N]\). In this sense, every integral flow \(f\) in \(\mathcal{N}_{D[\mathbf{NK}_1]_\text{tr}}\)
with \(|f| = N\) corresponds to a collection of \(N\) vertex disjoint, maximum independent
sets in \(G[\mathbf{K}_N]\).

Let now \(f\) be a flow like above and let \(\{C_1, \ldots, C_N\}\) be the corresponding set of \(N\)
maximum vertex disjoint, maximum independent sets in \(G[\mathbf{K}_N]\). Consider one of the \(N\)
paths from \(s\) to \(t\) given by \(f\), say \((s, (v_1, i_1), \ldots, (v_p, i_p), t)\). The costs along this path are
\[
\text{cost}(s, (v_1, i_1)) + \sum_{j=1}^{p-1} \text{cost}((v_j, i_j), (v_{j+1}, i_{j+1})) + \text{cost}((v_p, i_p), t)
\]
\[
= -\lambda_c(s) + \lambda_b(t) - \sum_{j=1}^{p} (\lambda_c(v_j, i_j) - \lambda_b(v_j, i_j))
\]
\[
= \lambda_{\text{max}} - \sum_{j=1}^{p} W(v_j, i_j).
\]

Hence we obtain
\[
\text{COST}(f) = N \lambda_{\text{max}} - \sum_{i=1}^{N} \sum_{(u,j) \in V(C_i)} W(u,j),
\]
as required.
Consider an instance \((H, W)\) of MWCP\((N)\) with \(H\) being a cocomparability graph. Solving the problem for this instance by the algorithm of Sarrafzadeh and Lou, one cannot expect that the transitive orientation of \(\overline{H}\) can be replaced by its transitive reduction. This is due to the fact that in general no optimal solution exists such that every color class, that is, a set of vertices receiving the same color, is a maximal independent set of \(H\). But, if \(H = G[K_N]\), then there exists always an optimal solution such that all color classes are maximal independent sets of \(H\). Thereby, the correctness of the modified algorithm follows immediately from Lemma 8 like the correctness of the original one from Lemma 7.

Next we are going to investigate the time complexity of the modified algorithm. For this purpose we need the following lemma.

**Lemma 9.** Let \(G\) be a cocomparability graph and let \(D\) be a transitive orientation of \(\overline{G}\). Then \(|A(D_{tr})| \leq |V(G)| - 1 + 2|E(G)|\).

**Proof.** Let \(n = |V(G)|\). Since

\[n - 1 + 2|E(G)| = n - 1 + n(n - 1) - 2|A(D)| = n^2 - 1 - 2|A(D)|,
\]

it suffices to verify

\[2|A(D)| + |A(D_{tr})| \leq n^2 - 1. \tag{3}\]

Define the height \(h_D(v)\) of a vertex \(v \in V(G)\) as the length of a longest path in \(D\) ending in \(v\), and the height of \(D\) by \(h(D) = \max\{h_D(v) \mid v \in V(G)\}\).

Suppose that there exists a transitive digraph \(D\) violating (3). Let \(D\) be chosen with maximum height among all counterexamples. Denote by \(I\) a maximum independent set of vertices of \(D\), and let it be chosen such that \(\max\{h_D(v) \mid v \in I\}\) is minimum among all maximum independent sets of \(D\).

Since \(|I| = 1\) would imply that \(G\) is complete and hence \(2|A(D)| + |A(D_{tr})| = n(n - 1) + n - 1 = n^2 - 1\), that is, equality in (3), we may assume that \(|I| \geq 2\). Next let \(x \in I\) be a vertex such that \(h(D - x) = h(D)\). This is possible, since every directed path in \(D\) can contain at most one vertex of \(I\). Now add to \(D - x\) a new vertex \(x'\) and all arcs \((v, x')\) for \(v \in V(G) - \{x\}\). It is easy to see that the resulting digraph \(D'\) is transitive. Moreover, we have

\[|A(D')| \geq |A(D)| + |I| - 1. \tag{4}\]

Let \(I^+ = \{v \in V(G) \mid (x, v) \in A(D_{tr})\}\) and \(I^- = \{v \in V(G) \mid (v, x) \in A(D_{tr})\}\). Since every arc of \(D_{tr} - x\) belongs also to \(D_{tr}'\) and since at least one arc of \(D_{tr}'\) is incident with \(x'\), we have

\[|A(D_{tr}')| \geq |A(D_{tr})| - |I^+| - |I^-| + 1. \tag{5}\]

By their definition, \(I^+\) and \(I^-\) are independent sets of \(D\) and thus \(|I^+| \leq |I|\) and \(|I^-| \leq |I|\). In the later estimate equality cannot appear, since we have max\(\{h_D(v) \mid v \in I^-\}\) =
\[ h_D(x) - 1 < \max \{ h_D(v) \mid v \in I \}, \] and so \(|I^+| = |I|\) would contradict the choice of \(I\).

Hence we obtain with (4) and (5)

\begin{align*}
2 |A(D')| + |A(D'_\tau)| & \geq 2(|A(D)| + |I| - 1) + |A(D'_\tau)| - |I^+| - |I^-| + 1 \\
& \geq 2|A(D)| + 2|I| - 2 + |A(D'_\tau)| - 2|I| + 2 \\
& = 2|A(D)| + |A(D'_\tau)|.
\end{align*}

This implies that also \(D'\) violates (3). But this is impossible by the choice of \(D\), since \(h(D') = h(D - x) + 1 = h(D) + 1\). This contradiction completes the proof of the lemma. \(\square\)

Now we are able to prove the following result.

**Theorem 10.** Let \((G, \{f_x \mid x \in V(G)\})\) be an instance of \(\text{CAP}(N)\), where \(G\) is a co-comparability graph and \(f_x\) is convex for every \(x \in V(G)\). Let \(D\) denote a transitive orientation of \(G\). If the transitive reduction \(D_t\) of \(D\) is known, then a solution can be found in \(O(N^3 \log(Nn))\) time, where \(n\) and \(e\) denote order and size of \(G\), respectively.

**Proof.** We have already verified the correctness of the modified algorithm.

To analyse its time performance we suppose for the moment that \(e \geq n - 1\). The directed graph of the network \(\mathcal{N}_{D[NK]}\) has \(Nn + 2\) vertices and, by Lemma 9, the number of its arcs is \(O(N^2 e)\). Starting from \(D_t\) the network can be constructed in \(O(N^2 e)\) time. Computing a topological sorting takes also \(O(N^2 e)\) time \([1]\). Begin- and end-weights are computed using the topological sorting in \(O(N^2 e)\) time, and thus the network \(\mathcal{N}_{D[NK]}\) and the cost function \(\text{cost}\) can be determined in \(O(N^2 e)\) time. Applying the algorithm of \([26]\) (Theorem 8.13) to find a minimum cost flow \(f\) with \(|f| = N\) takes \(O(N^3 e \log(Nn))\) time. Finally, the optimal channel assignment can be derived from \(f\) in time \(O(N^2 e)\).

If \(e \leq n - 2\), then \(G\) is disconnected and we apply the algorithm to the components of \(G\). For each component with \(e'\) edges and \(n'\) vertices it holds \(e' \geq n' - 1\), and so the algorithm runs in \(O(N^3 e' \log(Nn))\) time as shown above. Thus, the time needed to solve the problem for all components is again \(O(N^3 e \log(Nn))\). \(\square\)

To explain the impact of Theorem 10 consider the following situation. Let \(G\) be the interference graph of a cellular radio network. Such graphs are usually sparse, say \(e \leq cn\) for some small constant \(c\), since cellular radio networks are designed such that the number of possible interferences is small. If \(G\) is a co-comparability graph and if \(D_{ir}\) is known, then Theorem 10 indicates that the worst case performance of the last phase of the modified algorithm is approximately by a factor \(n\) faster than that of the original one, since \(\bar{G}\) and \(D\) are dense. The last phase of the algorithms is of particular importance. On the one hand it turned out that for realistic problem sizes this phase is the most time consuming one. On the other hand we can consider a semi-dynamical channel allocation strategy called Iterative FCA. Hereby, \(D_{ir}\) can be determined during
the planning process. During the operation of the network the traffic is measured and from time to time a new channel allocation is used, which is computed with respect to the present demand.

If the modified algorithm is used for FCA, then the computation of $D_{tr}$ has also been taken into account. Therefore, $\bar{G}$ and $D$ are computed in $O(n^2)$ time like before. Thereafter, $D_{tr}$ can be found in $O(n + \bar{e} + ne_{tr})$ time [13], where $e_{tr}$ is the number of arcs of $D_{tr}$. Note that, by Lemma 9, $D_{tr}$ is sparse, if $G$ is sparse.

Cocomparability graphs turned out to be a good model for large parts of real world networks. In [17] a cellular radio network for the area of Helsinki is described. The interference graph of a subnetwork obtained by removing only four base stations is a cocomparability graph. Moreover, cocomparability graphs include (unit) interval graphs, which were used as a model for cellular radio networks covering a highway (see, for instance, [8, 9, 16, 21]). Consider therefore base stations covering a line segment. The possible interferences can be described by intervals around the base stations which correspond to areas with high signal strength and intersecting intervals indicate that interference can occur. On interval graphs the CAP($N$) can be solved much faster than on cocomparability graphs by Theorem 10. Carlisle and Lloyd [3] presented recently an algorithm solving the MWCP($N$) on interval graphs in $O(e + NS(n))$ time, where $S(n)$ denotes the running time of any algorithm for finding shortest paths in graphs with $O(n)$ edges. Furthermore, they showed that the running time is even $O(NS(n))$, if the graph is represented by a set of intervals. Using a result from [7] their algorithm can be implemented to run in $O(e + Nn \log n)$ or $O(Nn \log n)$ time, respectively. Hence, by Theorem 4 and Lemma 6, CAP($N$) can be solved in $O(N^2(e + n \log(Nn)))$ time or in $O(N^2n \log(Nn))$ time (note that we saw in the proof of Lemma 6 that an interval representation of $G$ yields also one of $G[K_N]$).

Another class of graphs on which the MWCP($N$) is tractable are comparability graphs. These graphs contain in particular so-called Manhattan networks, that is, regular two-dimensional lattices, which were used as an interference model for urban areas. The MWCP($N$) on comparability graphs is equivalent to the weighted $N$-family problem for partially ordered sets. Berenguer et al. [2] gave a series of polynomial reductions ending up with a maximum flow problem. Using the push-relabel method of Goldberg and Tarjan [11], which is superior to several other methods [10], their algorithm can be implemented to run in $O(N^2ne \log(Nn^2/e))$ time. With Theorem 4 and Lemma 5 we find that CAP($N$) can be solved thereby in $O(N^5ne \log(Nn^2/e))$ time. A considerable improvement of this result can be obtained by using a recent algorithm of Engel [4]. He solved the $N$-family problem by reducing it to a minimum cost flow problem. By means of this reduction and the algorithm of [26], the MWCP($N$) can be solved in $O(Ne \log_{2+e/n} n)$ time for comparability graphs. So, by Theorem 4 and Lemma 5, the CAP($N$) can be solved in $O(N^3e \log(nN))$ time.

Finally we would like to remark that Yannakakis and Gavril [27] have shown that MWCP($N$) is solvable in polynomial time on chordal graphs. They also proved that the maximum $N$-colorable subgraph problem for split graphs (and thus, also for chordal graphs and their complements) is NP-hard, if $N$ is not fixed.
Acknowledgement

I would like to thank Jutta Mitas for her help concerning Lemma 9. The result is conjectured in [21] for complements of unit interval graphs. She pointed out that it probably holds for all comparability graphs. Moreover, after preparing this manuscript she presented a different proof. Very special thanks to Konrad Engel for pointing out his new algorithm and to the referees for their valuable comments.

References

