Coordinating concurrent transmissions : a constant-factor approximation of maximum-weight independent set in local conflict graphs

Kaski, Petteri

Coordinating Concurrent Transmissions: A Constant-Factor Approximation of Maximum-Weight Independent Set in Local Conflict Graphs

PETTERI KASKI[1], ALEKSI PENTTINEN[2] AND JUKKA SUOMELA[1]

[1]*Helsinki Institute for Information Technology HIIT*
Helsinki University of Technology and University of Helsinki
P.O. Box 68, FI-00014 University of Helsinki, Finland
petteri.kaski@cs.helsinki.fi, jukka.suomela@cs.helsinki.fi
[2]*Department of Communications and Networking*
Helsinki University of Technology
P.O. Box 3000, FI-02015 TKK, Finland
aleksi.penttinen@tkk.fi

We study the algorithmic problem of coordinating transmissions in a wireless network where radio interference constrains concurrent transmissions on wireless links. We focus on pairwise conflicts between the links; these can be described as a conflict graph. Associated with the conflict graph are two fundamental network coordination tasks: (a) finding a nonconflicting set of links with the maximum total weight, and (b) finding a link schedule with the minimum total length. Our work shows that two assumptions on the geometric structure of conflict graphs suffice to achieve polynomial-time constant-factor approximations: (i) bounded density of devices, and (ii) bounded range of interference. We also show that these assumptions are not sufficient to obtain a polynomial-time approximation scheme (PTAS) for either coordination task. There exists a PTAS if we make an additional assumption: (iii) bounded range of radio transmissions.
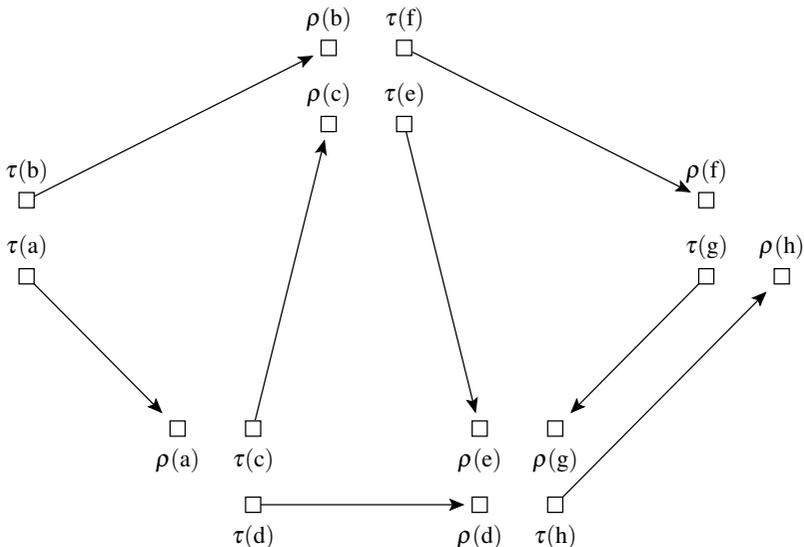
1

Figure 1

A wireless network. Wireless communication links are marked with arrows. Devices are marked with boxes, τ denotes a transmitter and ρ denotes a receiver. For clarity, each device in this illustration takes part in only one transmission.

1. INTRODUCTION

A fundamental challenge in wireless networking is the shared transmission medium, which in many cases prevents concurrent transmissions due to radio interference. This brings forth the algorithmic problem of coordinating the transmissions so that performance loss due to interference does not occur. In this work, we investigate the polynomial-time approximability of network coordination within the following framework.

Interference in Wireless Networks

A wireless network consists of devices which communicate with each other by radio transmissions. We study unicast networks where each radio transmission has one designated recipient, see Figure 1 for an illustration.

A radio transmission may interfere with other transmissions. We focus on systems where the radio interference is dominated by the near-far effect: radio reception from a distant transmitter may be blocked by other transmitters which are much closer to the receiver.
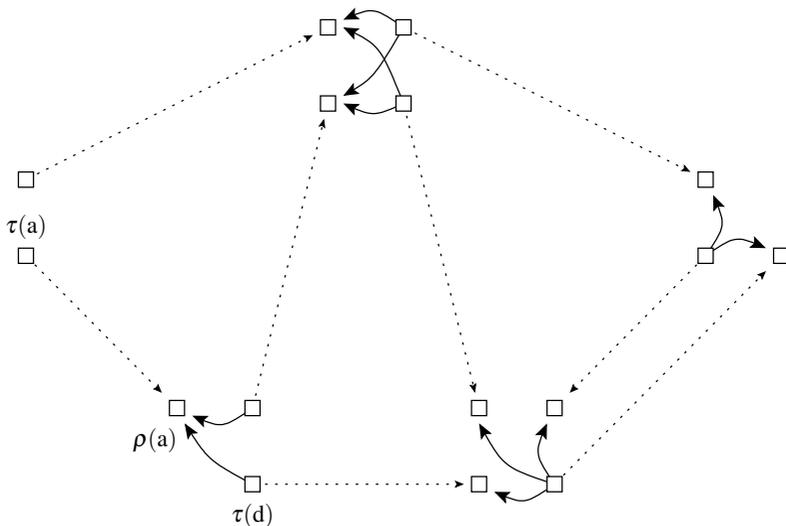
Figure 2

The near-far effect in a wireless network. Solid arrows point from an interfering transmitter to the interfered receiver. For example, if the transmitter $\tau(d)$ is active, the device $\rho(a)$ cannot receive the transmission from $\tau(a)$. The signal power received from $\tau(a)$ is too low in comparison with the interfering power received from $\tau(d)$.

Figure 2 illustrates transmitter–receiver pairs where the near-far effect might occur in our example. In the illustration, the link from $\tau(a)$ to $\rho(a)$ and the link from $\tau(d)$ to $\rho(d)$ cannot be active simultaneously: the transmitter of the latter blocks the receiver of the former.

We focus on pairwise conflicts between the links. The pairwise conflicts can be described as a *conflict graph* [1]. A conflict graph $G = (V, E)$ is an undirected graph where each vertex $v \in V$ corresponds to a communication link and an edge $\{u, v\} \in E$ describes that the links u and v are mutually conflicting. To clarify the difference between the conflict graph and the underlying communication network, we use the words *device* and *link* in the context of the communication network and the words *vertex* and *edge* in the context of the conflict graph. There is a one-to-one correspondence between the links of the communication graph and the vertices of the conflict graph.

Figure 3 illustrates a conflict graph; the vertex $a \in V$ in the conflict graph corresponds to the communication link from $\tau(a)$ to $\rho(a)$ and the vertex $d \in V$ in the conflict graph corresponds to the communication link from $\tau(d)$ to $\rho(d)$. As these two links cannot transmit simultaneously, there is an edge $\{a, d\} \in E$ in the conflict graph.
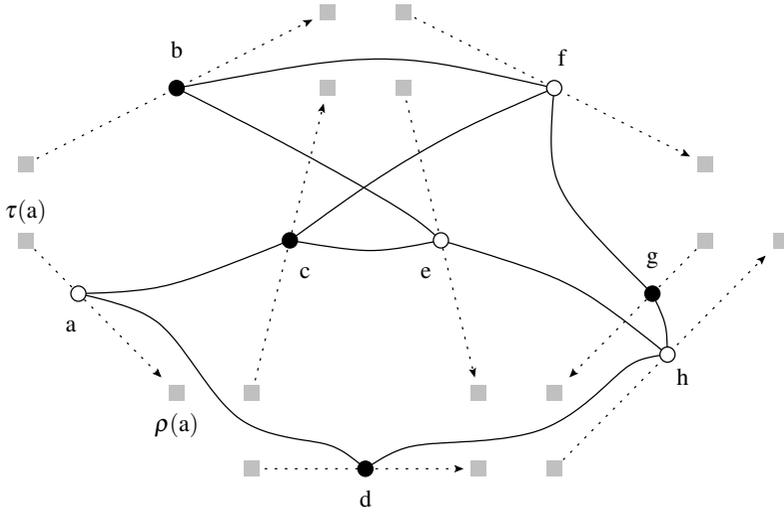
Figure 3

The conflict graph for the example in Figure 2. There is one vertex for each communication link and one edge for each pair of conflicting links; vertices are marked with circles and edges are marked with solid lines. Note that even though interference in Figure 2 was highly localised, such locality is no longer immediately visible in the conflict graph. An independent set {b, c, d, g} is highlighted; these communication links can be active simultaneously.

Algorithmic Problems and Earlier Work

Associated with a conflict graph are two network coordination tasks:

(1) Given some weights (such as priorities or utilities) on each vertex, find an *independent set* of the maximum total weight; in other words, find a nonconflicting set of links of the maximum total weight. See Figure 3 for an example.

(2) Given some data transmission needs on each link, find a *link schedule* of the minimum length such that at each point in time, the set of active links is nonconflicting, and each link is active for a time that suffices to cover its data transmission needs. See Section 6 for a precise linear programming formulation.

An approximation algorithm for maximum-weight independent set also implies an approximation algorithm for the link scheduling problem in the same class of graphs [2, 3]. Unfortunately, both problems are prohibitively hard to approximate in general graphs [4, 5, 6]. Jain et al. [1] have put forth the question of whether there is a family of conflict graphs which arises in

realistic network deployments and which makes the problem of finding an independent set easier.

Contribution

Our work shows that two assumptions on the structure of conflict graphs suffice in order to achieve a polynomial-time constant-factor approximation of maximum-weight independent set and link scheduling:

(i) Bounded density of the devices. Radios are points in a low-dimensional space and they are not located in an arbitrarily dense manner.

(ii) Bounded range of interference. Conflicts are caused by the near-far effect; if there is a conflict, the interfering transmitter is close to the interfered receiver.

These assumptions can be physically justified: in addition to the obvious physical limits on transmitter density, the received signal power typically decays in inverse proportion to more than the square of the distance from the transmitter [7]. Thus, compared with the background noise, the received power from a distant interfering transmitter is negligible.

Note that we do not need to assume that there has to be interference in certain situations, say, between devices close to each other; indeed, such assumptions are often not valid in practice [7, 8]. We can make measurements in the deployed physical system to determine whether a pair of links is mutually conflicting; there is no need to use a simplifying model of radio interference.

We also show that the assumptions (i) and (ii) alone are not sufficient in order to achieve an arbitrarily small approximation ratio. Further assumptions are required; we shall see that it suffices to assume a bounded range of radio transmissions.

2. STATEMENT OF RESULTS

Let N be a constant which controls the relative density of the devices.

Definition 1. An *N-local conflict graph* is a tuple (G, τ, ρ) where $G = (V, E)$ is an undirected graph and τ, ρ are functions $V \to \mathbb{R}^2$ such that

(i) the function $v \mapsto (\tau(v), \rho(v))$ is an injection, and no unit disk in $\mathbb{R}^2$ contains more than N points in $\tau(V) \cup \rho(V)$

(ii) for all $\{v, u\} \in E$ it holds that $d(\tau(v), \rho(u)) < 1$ or $d(\tau(u), \rho(v)) < 1$ where $d(\cdot, \cdot)$ is the Euclidean distance.

We call $\tau(v)$ the *transmitter* and $\rho(v)$ the *receiver*, the intuition being that a pair $(\tau(v), \rho(v))$ corresponds to a data transmission link. Note that $d(\tau(v), \rho(v))$ is unrestricted and that some receivers and transmitters may coincide; however, the pair $(\tau(v), \rho(v))$ must be unique for every vertex.

If we required $\tau(v) = \rho(v)$ for each $v \in V$, we would obtain what we call $(2, N)$-*local graphs* [9]; these are similar to *civilised graphs* [10, §8.5]. Thus, N-local conflict graphs can be interpreted as a natural generalisation of the families of local graphs and civilised graphs.

In Section 3, we derive some basic properties of N-local conflict graphs. On one hand, we shall see that N-local conflict graphs is a large family of graphs; in addition to $(2, N)$-local graphs, it contains, for example, all bipartite graphs. On the other hand, we show that N-local conflict graphs are not contained in families such as bounded-degree, bounded-density, bipartite, planar, or disk graphs — prior work on these widely studied families does not directly answer our research questions.

In Section 4, we prove our main result; here *MWIS* refers to the problem of finding a maximum-weight independent set:

Theorem 2. *MWIS for N-local conflict graphs admits a polynomial-time* $(5 + \varepsilon)$-*approximation algorithm for any constants* $\varepsilon > 0$ *and N.*

While the time complexity of the algorithm depends on the parameter N, we emphasise that the approximation ratio does not depend on N. This is unlike families such as bounded-degree graphs where achievable approximation ratios typically depend on the parameters of the family [11]; for example, MWIS in graphs of maximum degree Δ can be approximated within a factor of $O(\Delta \log \log \Delta / \log \Delta)$ [12].

In Section 5, we show that approximating beyond a certain constant factor remains hard:

Theorem 3. *MWIS for N-local conflict graphs admits no polynomial-time approximation scheme (PTAS) for any N unless* $P = NP$.

This is unlike families such as disk graphs; for example, MWIS in disk graphs admits a PTAS [13].

In Section 6, we consider the problem of fractional covering by independent sets in local conflict graphs. We obtain analogous approximability and inapproximability results for the covering LP which captures the link scheduling problem. In Section 7, we generalise from the link scheduling problem to a joint scheduling and routing problem, and in Section 8, we generalise beyond pairwise conflicts.

In Section 9, we consider an additional assumption on the underlying wireless network: bounded range of radio transmissions. With this assumption, we are able to obtain a PTAS for both MWIS and link scheduling. Furthermore, there exists a local, distributed approximation scheme for link scheduling if the communication network is a geometric spanner.

3. REPRESENTABILITY

It is not immediate from Definition 1 which graphs admit representation as a local conflict graph. The purpose of this section is to shed some light on this question. We begin by showing that local conflict graphs are not contained in families of graphs such as planar graphs, bounded-degree graphs, or disk graphs.

A first observation is that the family of N-local conflict graphs is closed under deletion of edges and vertices. Furthermore, an N_1-local conflict graph is an N_2-local conflict graph for any $N_2 \geq N_1$.

Theorem 4. *Any bipartite graph can be represented as a* 1-*local conflict graph.*

Proof. Consider a bipartite graph $G = (V, E)$. The set V can be partitioned into

$$A = \{a_1, a_2, \ldots, a_m\},$$
$$B = \{b_1, b_2, \ldots, b_n\}$$

for some m, n such that all edges are between A and B. Let $\tau(a_i) = (3i, -3)$ and $\rho(a_i) = (0, 0)$ for all i; let $\tau(b_j) = (0, 0)$ and $\rho(b_j) = (3j, 3)$ for all j. □

It follows that the maximum degree of a vertex in a 1-local conflict graph can be as high as $|V| - 1$ (consider the complete bipartite graph $K_{1,n}$), the average degree and the minimum degree can be as high as $|V|/2$ (consider $K_{n,n}$), and there are 1-local conflict graphs that are not planar and not disk graphs (consider $K_{3,3}$).

A local conflict graph need not be bipartite. To illustrate the rich substructure that can occur in a local conflict graph, we show that a local conflict graph may contain relatively large but not arbitrarily large cliques. First we prove the positive result.

Theorem 5. *A complete graph on N^2 vertices can be represented as an N-local conflict graph.*

Proof. Consider a complete graph with vertices

$$V = \{v_{i,j} \mid i, j \in \{1, 2, \ldots, N\}\}.$$

Let $\tau(v_{i,j}) = (0, i/N)$ and $\rho(v_{i,j}) = (0, j/N)$ for all i and j.   □

Next we proceed to obtain the negative result.

Lemma 6. *For every tournament (complete oriented graph) $G = (V, A)$ on n vertices, there are $s \in V$, $t \in V$ and $X \subseteq V$ with $\{s\} \times X \subseteq A$, $X \times \{t\} \subseteq A$, and $|X| \geq (n-2)/6$.*

Proof. Let $v, u \in V$, $v \neq u$. Let

$$\begin{aligned}
Q &= \{x \mid (v, x) \in A, (x, u) \in A\}, \\
R &= \{x \mid (u, x) \in A, (x, v) \in A\}, \\
S &= \{x \mid (x, v) \in A, (x, u) \in A\}, \\
T &= \{x \mid (v, x) \in A, (u, x) \in A\}.
\end{aligned}$$

If $|Q| \geq (n-2)/6$ or $|R| \geq (n-2)/6$, we are done. Otherwise, $|S| + |T| \geq 2(n+1)/3$. If $|S| \geq (n+1)/3$, the subgraph induced by S contains a vertex a with outdegree at least $(n-2)/6$; let $s = a$, $t = u$, and let $X \subseteq S$ consist of the successors of a. The case $|T| \geq (n+1)/3$ is analogous.   □

Theorem 7. *A complete graph on $6N^2 + 8$ vertices cannot be represented as an N-local conflict graph.*

Proof. To reach a contradiction, assume that there is a complete graph on $6N^2 + 8$ vertices that is an N-local conflict graph. We say that $P(v, u)$ holds if $d(\tau(v), \rho(u)) < 1$. Orient the graph as follows: if $P(v, u)$ and not $P(u, v)$, assign the direction (v, u) on $\{v, u\} \in E$; if $P(u, v)$ and not $P(v, u)$, assign the opposite direction; otherwise both $P(v, u)$ and $P(u, v)$ hold, in which case assign an arbitrary direction.

Choose s, t, and X as in Lemma 6; $|X| \geq N^2 + 1$. Now, $P(s, x)$ and $P(x, t)$ hold for all $x \in X$. A unit disk centred at $\tau(s)$ contains all points $\rho(X)$. There can be at most N distinct points; thus, there is a set $X' \subseteq X$ and a point $\rho' \in \mathbb{R}^2$ such that $\rho(X') = \{\rho'\}$ and $|X'| \geq N + 1$.

A unit disk centred at $\rho(t)$ contains all points $\tau(X')$. Again, there can be at most N distinct points; thus, there are two distinct vertices $v, u \in X'$ and a point $\tau' \in \mathbb{R}^2$ with $\tau(v) = \tau(u) = \tau'$. This is a contradiction because $v \mapsto (\tau(v), \rho(v))$ is an injection by Definition 1.   □

It follows immediately that the family of N-local conflict graphs is not closed under taking minors (form a bipartite graph by splitting each edge of a large complete graph in two).

4. APPROXIMABILITY

In this section, we prove Theorem 2: we show that there is a polynomial-time $(5 + \varepsilon)$-approximation algorithm for MWIS in N-local conflict graphs for any constants $\varepsilon > 0$ and N. The parameters ε and N are fixed to some constants throughout this section; the input of the algorithm consists of a graph $G = (V, E)$, points $\tau(v)$ and $\rho(v)$ for each $v \in V$, and a weight $w(v)$ for each $v \in V$.

Generalisation of the Problem

We present the algorithm in a somewhat more general setting than required by the MWIS problem. We write $C(v, I)$ for the *contribution* of the vertex $v \in V$ in the proposed solution $I \subseteq V$; we shall shortly define what the contribution means in the case of MWIS. Let

$$W(v, I) = w(v)C(v, I), \qquad \text{for each } I \subseteq V, v \in V,$$
$$W(A, I) = \sum_{v \in A} W(v, I), \qquad \text{for each } A \subseteq V.$$

The objective is to find a solution $I \subseteq V$ that maximises $W(V, I)$, that is, *maximises weighted contributions*.

Next we define $C(v, I)$ so that maximising weighted contributions is equivalent to finding a maximum-weight independent set in G. In Section 8, we shall see a different definition of $C(v, I)$ which generalises the algorithm beyond pairwise conflicts.

The Special Case of MWIS

In the case of MWIS, we define the contribution $C(v, I)$ as follows. Let $C(v, I) = 1$ if $v \in I$ and there is no $u \in I$ with $\{v, u\} \in E$ and $d(\tau(u), \rho(v)) < 1$. Otherwise let $C(v, I) = 0$.

With these definitions, the set I that maximises $W(V, I)$ is — after removing vertices with zero contribution — a maximum-weight independent set in G. Therefore if we are able to find the solution $I \subseteq V$ which maximises weighted contributions, we are also able to solve MWIS.

Assumptions on the Contributions

Naturally we cannot expect to maximise the weighted contributions for an arbitrarily complicated definition of $C(v, I)$. We make some assumptions on the structure of $C(v, I)$. The assumptions are immediate in the case of MWIS for local conflict graphs.

First, define the set of possibly interfering vertices

$$U(v) = \{v\} \cup \{u \in V \mid d(\tau(u), \rho(v)) < 1\}.$$

The algorithm makes use of the following assumptions on C:

- $C(v, I)$ can be evaluated in polynomial time

- $C(v, I) = 0$ for all $v \notin I$

- $C(v, I_1) \geq C(v, I_2)$ for all $I_1 \subseteq I_2$ with $v \in I_1$ (contributions are nonincreasing)

- $C(v, I) = C(v, I \cap U(v))$ for all I and $v \in V$ (locality).

Subproblems

In the algorithm, the full problem is divided into *subproblems*. Each subproblem is defined by a subset $A \subseteq V$, and the associated task is to find a set $I \subseteq A$ that maximises $W(A, I)$.

Let $\widehat{W}(v) = W(v, \{v\})$ for each $v \in V$, and let $\widehat{W}(A) = \sum_{v \in A} \widehat{W}(v)$. We make use of the following properties. As $C(v, I) = 0$ for $v \notin I$, we have

$$W(A, I) = W(A \cap I, I), \qquad\qquad \text{for each } A \subseteq V.$$

By nonincreasingness, $W(v, I_1) \geq W(v, I_2)$ for all $I_1 \subseteq I_2 \subseteq V$ and $v \in I_1$; in particular, $\widehat{W}(v) \geq W(v, I)$ for all $v \in I$ and $I \subseteq V$. As $C(v, I) = 0$ for $v \notin I$, we have $\widehat{W}(v) \geq W(v, I)$ for all $v \in V \setminus I$ and $I \subseteq V$. In summary,

$$\widehat{W}(A) \geq W(A, I), \qquad\qquad \text{for each } A, I \subseteq V,$$
$$\widehat{W}(I) \geq W(I, I) = W(V, I), \qquad\qquad \text{for each } I \subseteq V.$$

The Approximation Algorithm

To create the subproblems, we apply a shifting strategy [14, 15, 16]. We make the following initial assignments. Choose an integer $k \geq 3$ such that

$$\frac{5k^2}{(k-2)^2} < 5 + \varepsilon.$$

Let

$$
\begin{aligned}
A'_i &= \{(x,y) \in \mathbb{R}^2 \mid i \le x < i+1\}, & i \in \mathbb{Z}, \\
B'_j &= \{(x,y) \in \mathbb{R}^2 \mid j \le y < j+1\}, & j \in \mathbb{Z}, \\
A_i &= \bigcup\{A'_t \mid t \in \mathbb{Z}, t \equiv i \pmod{k}\}, & i = 0,1,\ldots,k-1, \\
B_j &= \bigcup\{B'_t \mid t \in \mathbb{Z}, t \equiv j \pmod{k}\}, & j = 0,1,\ldots,k-1, \\
D_{ij} &= \mathbb{R}^2 \setminus A_i \setminus B_j, & i,j = 0,1,\ldots,k-1.
\end{aligned}
$$

Each D_{ij} consists of squares $k-1$ units wide and high. We write $D_{ij1}, D_{ij2}, \ldots$ for the nonempty squares of D_{ij}. Let $Z_{ij} \subseteq V$ be the set of vertices v with both $\tau(v)$ and $\rho(v)$ in D_{ij}, and let $X_{ij\beta} \subseteq Z_{ij}$ be the set of vertices v with both $\tau(v)$ and $\rho(v)$ in $D_{ij\beta}$. Form the set of "short links" $X_{ij} = \bigcup_\beta X_{ij\beta}$ and the set of "long links" $Y_{ij} = Z_{ij} \setminus X_{ij}$.

Now we can use the procedure in Figure 4 to find an approximately optimal solution. In the first part we solve small subproblems by exhaustive search; in the second part we apply the standard greedy algorithm for finding a large cut. See Figure 5 for an illustration.

Proof of Correctness

The time complexity of the algorithm is polynomial in the size of the input since the number of nonempty squares $D_{ij\beta}$ is bounded by $2|V|$, the number of distinct transmitters or receivers in each square is bounded by a constant, and a pair $(\tau(v), \rho(v))$ uniquely determines v for all $v \in V$. The following three lemmata establish the correctness of the algorithm. We denote by $I^*(A)$ an optimal solution to the subproblem A.

Lemma 8. *Each S_{ij} is an optimal solution of the subproblem X_{ij}.*

Proof. To reach a contradiction, assume that S_{ij} is not optimal, i.e., there exists an $I \subseteq X_{ij}$ with $W(X_{ij}, I) > W(X_{ij}, S_{ij})$. Then there exists a β with $W(X_{ij\beta}, I) > W(X_{ij\beta}, S_{ij})$. As the squares $D_{ij\delta}$ are separated by stripes of width one, $U(v) \cap X_{ij} \subseteq X_{ij\beta}$ for all $v \in X_{ij\beta}$. Thus,

$$
\begin{aligned}
W(X_{ij\beta}, I) &= W(X_{ij\beta}, I \cap X_{ij\beta}), \\
W(X_{ij\beta}, S_{ij}) &= W(X_{ij\beta}, S_{ij} \cap X_{ij\beta}) = W(X_{ij\beta}, S_{ij\beta}).
\end{aligned}
$$

Thus, $W(X_{ij\beta}, I \cap X_{ij\beta}) > W(X_{ij\beta}, S_{ij\beta})$, contradicting the choice of $S_{ij\beta}$. $\square$

Lemma 9. *Each R_{ij} is a 4-approximate solution of the subproblem Y_{ij}.*

1. Initialise $\mathcal{N}$ to an empty set.

2. For all $i = 0, 1, \ldots, k-1$ and $j = 0, 1, \ldots, k-1$:

   (a) For each nonempty square $D_{ij\beta}$, find a subset $I \subseteq X_{ij\beta}$ which maximises $W(X_{ij\beta}, I)$. Call this set $S_{ij\beta}$.

   (b) Let $S_{ij} = \bigcup_\beta S_{ij\beta}$. Insert S_{ij} into $\mathcal{N}$.

3. For all $i = 0, 1, \ldots, k-1$ and $j = 0, 1, \ldots, k-1$:

   (a) Initialise Γ and Λ to empty sets.

   (b) For each nonempty square $D_{ij\beta}$, in an arbitrary order: Write $[\beta, \Gamma]$ for the set of vertices $v \in Y_{ij}$ such that one of the points $\tau(v), \rho(v)$ is located in $D_{ij\beta}$ and the other point is located in

   $$D_{ij\Gamma} = \bigcup_{\gamma \in \Gamma} D_{ij\gamma}.$$

   Define the set $[\beta, \Lambda]$ similarly. Add β to the set Λ if $\widehat{W}([\beta, \Gamma]) > \widehat{W}([\beta, \Lambda])$; otherwise add β to Γ.

   (c) Define

   $$T_1 = \{v \in Y_{ij} \mid \tau(v) \in D_{ij\Gamma}, \rho(v) \in D_{ij\Lambda}\},$$
   $$T_2 = \{v \in Y_{ij} \mid \tau(v) \in D_{ij\Lambda}, \rho(v) \in D_{ij\Gamma}\}.$$

   Let $T \in \{T_1, T_2\}$ be the set that maximises $\widehat{W}(T)$. Call this set R_{ij}, and insert R_{ij} into $\mathcal{N}$.

4. Return $\tilde{I} = \arg\max_{I \in \mathcal{N}} W(V, I)$.

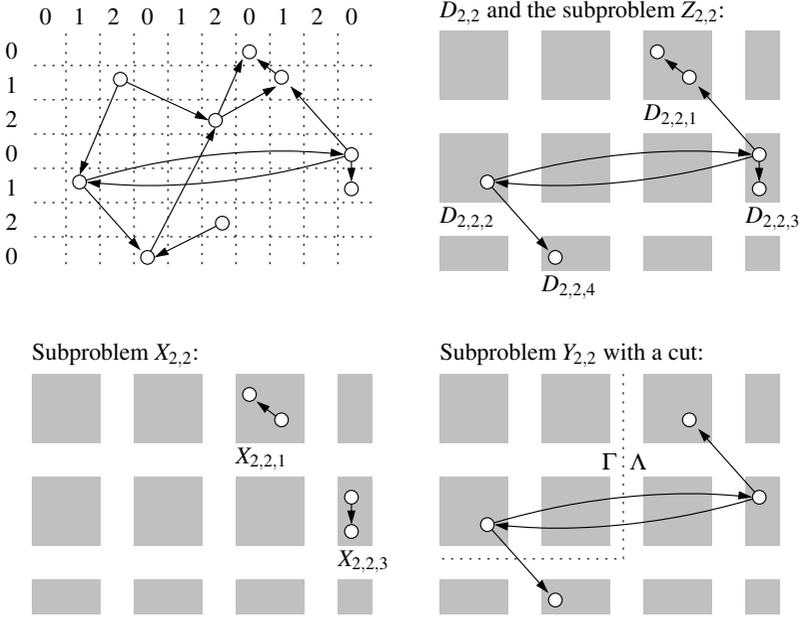Figure 4
The approximation algorithm.

Figure 5
An illustration of the approximation algorithm. In this example, $k = 3$.

Proof. The greedy algorithm in parts (3a) and (3b) finds a cut (in the directed graph with an arc of weight $\widehat{W}(v)$ from $\tau(v)$ to $\rho(v)$ for each $v \in Y_{ij}$) of a total weight at least $\widehat{W}(Y_{ij})/2$, implying by (3c) that $\widehat{W}(Y_{ij})/4 \leq \widehat{W}(R_{ij})$. All transmitters of R_{ij} are in $D_{ij\Gamma}$ and all receivers of R_{ij} are in $D_{ij\Lambda}$ or vice versa. The distance between any receiver and transmitter is larger than 1. Thus, $U(v) \cap R_{ij} = \{v\}$ for each $v \in R_{ij}$, implying that $W(v, R_{ij}) = W(v, \{v\}) = \widehat{W}(v)$. Therefore, $\widehat{W}(R_{ij}) = W(Y_{ij}, R_{ij})$. In summary, it holds that

$$W(Y_{ij}, I^*(Y_{ij})) \leq \widehat{W}(Y_{ij}) \leq 4W(Y_{ij}, R_{ij}). \qquad \square$$

Lemma 10. *The set $\tilde{I}$ is a $\left(5k^2/(k-2)^2\right)$-approximate solution.*

Proof. There is exactly one set A_i and exactly one set B_j that contains any given point in $\tau(V) \cup \rho(V)$. For each vertex $v \in V$, there are at most two sets A_i and at most two sets B_j that contain $\tau(v)$ or $\rho(v)$. Thus, there are at least $(k-2)^2$ pairs (i, j) such that D_{ij} contains both $\tau(v)$ and $\rho(v)$. In notation, $|\{(i, j) \mid v \in Z_{ij}\}| \geq (k-2)^2$ for all $v \in V$.

Let $I_{ij}^* = I^*(V) \cap Z_{ij}$. As the contributions are nonincreasing, we have $W(v, I^*(V)) \leq W(v, I_{ij}^*)$ for all $v \in I_{ij}^*$ and $W(Z_{ij}, I^*(V)) \leq W(Z_{ij}, I_{ij}^*)$ for all

Z_{ij}, implying that

$$(k-2)^2 W(V, I^*(V)) \leq \sum_{i,j} W(Z_{ij}, I^*(V)) \leq \sum_{i,j} W(V, I_{ij}^*).$$

Thus, there is a pair (i, j) satisfying $(k-2)^2 W(V, I^*(V)) \leq k^2 W(V, I_{ij}^*)$. As the sets X_{ij} and Y_{ij} partition Z_{ij}, we have

$$\begin{aligned} W(V, I_{ij}^*) &= W(X_{ij}, I_{ij}^*) + W(Y_{ij}, I_{ij}^*) \\ &\leq W(X_{ij}, I^*(X_{ij})) + W(Y_{ij}, I^*(Y_{ij})). \end{aligned}$$

By Lemmata 8 and 9, we obtain

$$\begin{aligned} W(X_{ij}, &I^*(X_{ij})) + W(Y_{ij}, I^*(Y_{ij})) \\ &\leq W(V, S_{ij}) + 4W(V, R_{ij}) \\ &\leq (1+4) \max\{W(V, S_{ij}), W(V, R_{ij})\} \\ &\leq 5W(V, \tilde{I}). \end{aligned}$$

This completes the proof. □

5. INAPPROXIMABILITY

Above, we have seen that there is a constant-factor approximation algorithm for MWIS in N-local conflict graphs. A natural question to pose at this point is whether we can achieve *any* approximation ratio above 1 in polynomial time, that is, if there is a polynomial-time approximation scheme for MWIS. This turns out not to be the case — in this section we prove Theorem 3, showing that MWIS for N-local conflict graphs admits no PTAS for any N, assuming $P \neq NP$. We reduce from the problem of finding a maximum-weight directed cut.

Maximum-Weight Directed Cut

The maximum-weight directed cut problem is defined as follows: given a directed graph $G = (V, A)$ and a nonnegative weight $w(a)$ for each arc $a \in A$, find a subset $S \subseteq V$ such that the total weight of the resulting cut $\delta^+(S) = \{(u, v) \in A \mid u \in S, v \notin S\}$ is maximised. The problem is APX-complete already in the unweighted case [17, 18].

Reduction

We show that for any N, approximating MWIS in N-local conflict graphs within factor α implies approximating maximum-weight directed cut in an

arbitrary directed graph within factor α. The reduction is similar to the one used by Chvátal and Ebenegger [19]; applied here, the reduction actually shows that the underlying undirected graph of the directed line graph of an arbitrary directed graph is a 1-local conflict graph.

Consider an instance of maximum-weight directed cut $G = (V, A)$ with the arc weights $w(a)$. Without loss of generality, we may arbitrarily relabel the vertices so that

$$V = \{(0,3), (0,6), \ldots, (0,3|V|)\} \subseteq \mathbb{R}^2.$$

Construct an N-local conflict graph $G' = (V', E')$ with weights w' as follows. Let

$$V' = A,$$
$$E' = \{\{(t,u), (u,v)\} \mid (t,u) \in A, (u,v) \in A\}.$$

For each $(u,v) \in A$, let

$$\tau((u,v)) = u,$$
$$\rho((u,v)) = v,$$
$$w'((u,v)) = w((u,v)).$$

The construction is a valid N-local conflict graph for any $N \geq 1$.

Let $S \subseteq V$ be a directed cut of G. Now, $\delta^+(S)$ is an independent set of the same weight in G' because there is no pair of arcs $(t,u), (u,v)$ in $\delta^+(S)$. Conversely, let $I' \subseteq V'$ be an independent set in G'. Let S consist of all transmitters of the vertices in I'. Note that S contains no receiver of a vertex in I'. Thus, $I' \subseteq \delta^+(S)$, i.e., S defines a directed cut with weight at least that of I'.

It follows that if W^* is the maximum weight of a directed cut in G, there is an independent set with weight W^* in G'. An α-approximation algorithm for MWIS finds an independent set of weight at least W^*/α in G', which transforms to a directed cut with weight at least W^*/α in G. This completes the proof of Theorem 3. $\qquad\square$

6. LINK SCHEDULING

We proceed to consider fractional covering by independent sets in local conflict graphs. For an independent set I, we let $I(v) = 1$ if $v \in I$ and $I(v) = 0$ if $v \notin I$.
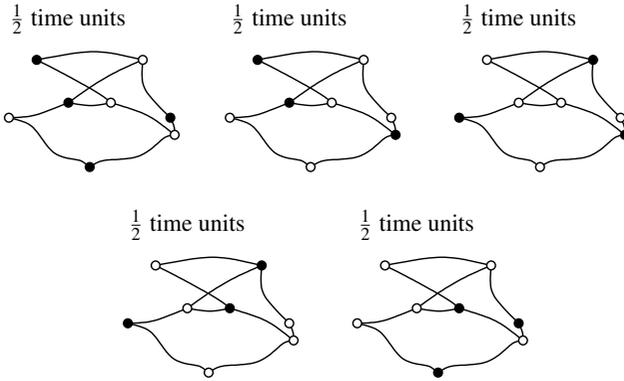
$\frac{1}{2}$ time units $\frac{1}{2}$ time units $\frac{1}{2}$ time units

$\frac{1}{2}$ time units $\frac{1}{2}$ time units

Figure 6
An link schedule of length 5/2.

Definition 11. In the *link scheduling problem*, the input consists of an N-local conflict graph (G, τ, ρ) and a *requirement* $r(v) \geq 0$ for each vertex v in G. The task is to

$$\text{minimise } \sum_I x(I)$$
$$\text{subject to } \sum_I I(v)x(I) \geq r(v) \quad \text{for all } v, \tag{1}$$
$$x(I) \geq 0 \qquad \text{for all } I,$$

where v ranges over all vertices in G and I ranges over all independent sets in G. The value $L = \sum_I x(I)$ is called the *length of the schedule*.

If G is a conflict graph and $r(v)$ is the amount of data that has to be transmitted on the link from $\tau(v)$ to $\rho(v)$, the link scheduling problem corresponds to finding an optimal schedule of data transmissions in a wireless communication network. In this setting, the vector x is interpreted as a schedule that assigns to independent set I the time slice $x(I)$.

The special case $r(v) = 1$ for all v corresponds to fractional colouring; the minimum schedule length is the fractional chromatic number of G.

Example 12. Consider the example in Figure 3, and assume that $r(v) = 1$ for each v. Let $I_1 = \{b, c, d, g\}$, $I_2 = \{b, c, h\}$, $I_3 = \{a, f, h\}$, $I_4 = \{a, e, f\}$, and $I_5 = \{d, e, g\}$; each of these sets is an independent set. Choose $x(I_1) = x(I_2) = x(I_3) = x(I_4) = x(I_5) = 1/2$. Now x is a solution to the link scheduling problem, and the length of the schedule equals 5/2. See Figure 6 for an illustration.

The number of variables in the above LP may be exponential in the size of the input. However, it is not necessary to construct the LP explicitly: the link scheduling problem can be solved $(\alpha + \varepsilon)$-approximately in polynomial time for any $\varepsilon > 0$ as long as there is a polynomial-time α-approximation algorithm for finding the maximum-weight independent set of G for arbitrary weights [2, 3]. Thus, Theorem 2 has the following corollary.

Corollary 13. *The link scheduling problem for N-local conflict graphs admits a polynomial-time $(5 + \varepsilon)$-approximation algorithm for any constants $\varepsilon > 0$ and N.*

The main result of this section shows that approximating beyond a certain constant factor remains hard.

Theorem 14. *The link scheduling problem for N-local conflict graphs admits no PTAS for any N unless $P = NP$.*

We begin with the following lemma.

Lemma 15. *There are constants k, $c > 1$, and Δ such that the following problem is NP-hard: Given a graph with maximum vertex degree at most Δ, decide whether its fractional chromatic number is at most k or at least ck.*

Proof. Khot [5] established that it is NP-hard to colour a k-colourable graph with $k^{\log(k)/25}$ colours for all sufficiently large constants k, even for graphs of bounded degree. In fact, Khot's proof shows that distinguishing between the following two cases is NP-hard for graphs of bounded degree: (i) There is a k-colouring. Thus, also the *fractional* chromatic number is at most k. (ii) The ratio of the number of vertices to the maximum size of an independent set is at least $k^{\log(k)/25}$. Thus, the *fractional* chromatic number is at least $k^{\log(k)/25}$. □

Proof of Theorem 14. We show that a PTAS for link scheduling in N-local conflict graphs can be used to solve the NP-hard problem in Lemma 15. Let $G = (V, E)$ be an arbitrary graph with maximum vertex degree at most Δ. Label the vertices so that $V \subseteq \mathbb{Z}$. We associate with each $v \in V$ three points, v_1, v_2, and v_3, by setting $v_j = (3v, 3j)$. No unit disk contains more than one point.

Construct an instance of the link scheduling problem: an N-local conflict graph $G' = (V', E')$ and the corresponding requirements $r(v')$ for each $v' \in V'$. We refer to the elements $v' \in V'$ as *links*, the intuition being that they correspond to a pair of transmitter and receiver; we reserve the word *vertex* for the elements $v \in V$.

For each vertex $v \in V$, introduce two links $(v_1, v_2) \in V'$ and $(v_2, v_3) \in V'$ with the requirements $r((v_1, v_2)) = k - 1$ and $r((v_2, v_3)) = 1$. For each edge $\{u, v\} \in E$, introduce two links $(u_2, v_2) \in V'$ and $(v_2, u_2) \in V'$ with the requirements $r((u_2, v_2)) = r((v_2, u_2)) = 1$. Finally, let

$$\tau((x, y)) = x,$$
$$\rho((x, y)) = y,$$
$$E' = \{\{(x, y), (y, z)\} \mid (x, y) \in V', (y, z) \in V'\}.$$

Select a positive

$$\varepsilon' < \min\left\{ \frac{\sqrt{c} - 1}{2}, \frac{1 - 1/\sqrt{c}}{2k\Delta} \right\},$$

and use the PTAS to solve the constructed link scheduling instance within factor $(1 + \varepsilon')$. Let the length of the schedule be L.

Let the fractional chromatic number of G be χ_f. If $\chi_f \leq k$, we may use a fractional colouring of size k to construct a feasible schedule of length k for the link scheduling instance as follows. Interpret the fractional colouring as a schedule in which the vertices may be active or inactive. Without loss of generality we may assume that the schedule is exact; that is, each vertex is active for exactly 1 time unit. Whenever $v \in V$ is active, transmit data on links (v_2, v_3) and (v_2, u_2) for each vertex u adjacent to v in G; this is possible since the vertices adjacent to v cannot be active and thus are not transmitting at the same time. Whenever $v \in V$ is inactive, transmit data on (v_1, v_2). Note that each vertex is active for 1 time unit and inactive for $k - 1$ time units, implying that the requirements $r(v')$ are met. Observe that if $\chi_f \leq k$, we have $L \leq (1 + \varepsilon')k$ since the length of the schedule cannot be more than k and we solved the link scheduling instance within factor $(1 + \varepsilon')$.

On the other hand, if $L \leq (1 + 2\varepsilon')k$, we may use a schedule of length $(1 + 2\varepsilon')k$ to construct a fractional colouring. Let $T(v')$ be the union of all time intervals when the link $v' \in V'$ is active in the schedule, and let $|T(v')|$ be the total length of these time intervals. Consider an arbitrary vertex $v \in V$. Let $V'(v)$ be the set of all links $(v_2, u_2) \in V'$ with $u \in V$. We have

$$T((v_1, v_2)) \cap T(v') = \emptyset$$

for all $v' \in V'(v)$. Furthermore, $|T(v_1, v_2)| \geq k - 1$, and the total schedule length equals $k + 2\varepsilon'k$. Thus,

$$\left| \bigcup_{v' \in V'(v)} T(v') \right| \leq 1 + 2\varepsilon'k.$$

On the other hand, $|T(v')| \geq 1$ for each $v' \in V'(v)$. Observe that

$$\left| \bigcap_{v' \in V'(v)} T(v') \right| \geq 1 - 2\varepsilon'k|V'(v)|$$

since each new edge introduced to the intersection shortens the intersection by at most $2\varepsilon'k$ units. Also observe that $|V'(v)| \leq \Delta$.

A non-isolated vertex v in the fractional colouring problem may be active in time intervals

$$\bigcap_{v' \in V'(v)} T(v')$$

because none of its neighbours may be active at the same time; for an isolated vertex (the case of an empty $V'(v)$), colouring is trivial. Thus, we have a partial fractional covering of length $(1 + 2\varepsilon')k$ that covers each vertex for at least $1 - 2\varepsilon'k\Delta$ units of time. Multiply all time assignments by $1/(1 - 2\varepsilon'k\Delta)$ to obtain a fractional colouring of size

$$\frac{(1 + 2\varepsilon')k}{1 - 2\varepsilon'k\Delta} < \frac{(1 + (\sqrt{c} - 1))k}{1 - (1 - 1/\sqrt{c})} = ck$$

that covers each vertex for at least 1 unit of time. Thus, $\chi_f < ck$.

To summarise, $\chi_f \leq k$ implies $L \leq (1 + \varepsilon')k$ and $\chi_f \geq ck$ implies $L > (1 + 2\varepsilon')k$. This shows that we can use a PTAS to distinguish in polynomial time between the two cases in Lemma 15. $\square$

Remark

It is not known whether Theorem 14 could be obtained as a simple corollary of Theorem 3. For example, the conversion method by Erlebach and Jansen [20] cannot be applied directly as it requires that the family of graphs is closed not only under the deletion of vertices but also under the duplication of vertices.

7. JOINT ROUTING AND SCHEDULING

We generalise the link scheduling LP (1) by letting the link capacity requirements $r(v)$ be variables constrained by a flow routing task in the communication network. We focus on one basic task: routing and scheduling m end-to-end flows between pairs of devices.

Let $F = (D, V)$ be a directed graph that models the communication network. The set D consists of the devices in the network and the set V consists

of the directed wireless links between devices. Let G be the associated conflict graph for the links V. For $k = 1, 2, \ldots, m$, let (s_k, t_k) be an ordered pair of devices, $s_k, t_k \in D$, and let $R(k) > 0$ be the amount of data that is to be routed from the source device s_k to the target device t_k. Denote by $\mathcal{P}_k$ the set of all directed (simple) paths of links from s_k to t_k in the network F. Let $\mathcal{P} = \bigcup_{k=1}^{m} \mathcal{P}_k$. For a path $P \in \mathcal{P}$ and a link $v \in V$, we write $v \in P$ to indicate that the link v occurs in P. Associated with each path P is a variable $r(P)$ giving the data rate on P.

Definition 16. In the *end-to-end routing and scheduling problem*, the input consists of the communication network F, the conflict graph G, and the source–target pairs (s_k, t_k) with their data amounts $R(k) > 0$, $k = 1, 2, \ldots, m$. The task is to

$$
\begin{aligned}
\text{minimise } & \sum_I x(I) \\
\text{subject to } \quad & \sum_{I:v\in I} x(I) \geq r(v) \quad \text{for all } v, \\
& \sum_{P:v\in P} r(P) \leq r(v) \quad \text{for all } v, \\
& \sum_{P:P\in\mathcal{P}_k} r(P) \geq R(k) \quad \text{for all } k, \\
& x(I), r(v), r(P) \geq 0 \qquad \text{for all } I, v, P,
\end{aligned}
\tag{2}
$$

where v ranges over the links V, I ranges over the independent sets of G, and P ranges over the paths $\mathcal{P}$.

In other words, the task is to route the $\sum_k R(k)$ units of data in the minimum possible total time subject to the interference constraints. The first constraint guarantees that each link v is active for a period of time long enough to transfer $r(v)$ data units. The second constraint guarantees that the data volume on paths through v does not exceed the allocated volume $r(v)$. The third constraint guarantees that in total $R(k)$ data units are routed on paths from s_k to t_k. A solution is assumed to consist of the paths $P \in \mathcal{P}$ with $r(P) > 0$ and the independent sets I with $x(I) > 0$, both accompanied by their respective values $r(P)$ and $x(I)$.

A general result of Jansen [2] implies that we can approximate (2) if we can approximate MWIS.

Theorem 17. *The end-to-end routing and scheduling problem admits an α-approximation algorithm if MWIS admits an α-approximation algorithm.*

Proof (sketch). An α-approximation algorithm to MWIS solves the following problem in polynomial time: for a given conflict graph G and link weights

$w(v) \geq 0$, find an independent set J with

$$\alpha \sum_{v \in J} w(v) \geq \max \left\{ \sum_{v \in I} w(v) : I \text{ is an independent set} \right\}. \tag{3}$$

Assume such an algorithm exists.

The dual of (2) is to

$$
\begin{array}{lll}
\text{maximise} & \sum_k R(k) z(k) & \\
\text{subject to} & \sum_{v \in I} w(v) \leq 1 & \text{for all } I, \\
& \ell(v) \leq w(v) & \text{for all } v, \\
& z(k) \leq \sum_{v \in P} \ell(v) & \text{for all } k \text{ and } P \in \mathcal{P}_k, \\
& w(v), \ell(v), z(k) \geq 0 & \text{for all } v, k.
\end{array} \tag{4}
$$

Let $\mathcal{K} \subseteq \mathbb{R}^n$ be the polytope defined by the dual constraints (4). Represent $\mathcal{K}$ as $\mathcal{K} = \mathcal{S} \cap \mathcal{R}$, where $\mathcal{S}$ and $\mathcal{R}$ are the polyhedra defined by the constraints indexed by I and all the other constraints, respectively.

We now claim that we have a polynomial-time algorithm A that given $(w, \ell, z) \in \mathbb{Q}^n$ either (i) asserts that $(w, \ell, z) \in (\alpha \mathcal{S}) \cap \mathcal{R}$ or (ii) finds an inequality in (4) violated by (w, ℓ, z) and thus witnessing that $(w, \ell, z) \notin \mathcal{S} \cap \mathcal{R}$. Indeed, we first check whether (w, ℓ, z) violates an inequality defining $\mathcal{R}$; this can be done in time polynomial in the size of the input. To see this, observe that testing nonnegativity and $\ell(v) \leq w(v)$ is trivial, and that the inequalities indexed by $P \in \mathcal{P}_k$ and k can be checked by a shortest-path computation from s_k to t_k in F (for example, using Dijkstra's algorithm) with the nonnegative link lengths $\ell(v)$. If all inequalities defining $\mathcal{R}$ are satisfied, we run our α-approximation algorithm for MWIS to obtain an independent set J, and check whether $\sum_{v \in J} w(v) \leq 1$ holds. If not, J yields a witness to (ii), otherwise we can assert (i) because of the approximation guarantee (3).

In the terminology used by Jansen [2], algorithm A constitutes a strong approximate separation oracle for the fractional packing polytope $\mathcal{K}$. Observing that $\mathcal{K}$ contains the all-zero vector, [2, Theorem 4.2] enables the α-approximate solution of (4) using the central cut ellipsoid algorithm [21, §3.2] and the method of continued fractions [21, §5.1]. Furthermore (see the proof of [2, Theorem 5.1] for a detailed development), the witness inequalities returned by algorithm A during the α-approximate solution of (4) give, by LP duality, a polynomial-size explicit restriction of the primal (2). The optimum of this restriction is within a factor α of the optimum of (2). Solving the restriction gives an α-approximate solution of (2). $\qquad\square$

It is immediate that the previous proof is to a large extent oblivious to the actual routing task at hand; essentially the only requirement is that the "routing polyhedron" $\mathcal{R}$ has a polynomial-time strong separation oracle (see [21]). Thus, besides routing end-to-end flows, an analogue of Theorem 17 can be established for many other basic routing tasks.

8. BEYOND PAIRWISE INTERFERENCE

Throughout this work, we have studied the case of pairwise conflicts which can be represented as a conflict graph. In real wireless networks, more complicated patterns of interference occur. In this section, we show how to extend our positive results to more general settings.

In wireless networks, the joint effect of simultaneous interfering transmissions fundamentally depends on the *signal-to-interference-plus-noise ratio* (SINR) at the receiver, that is, the ratio of the signal power received from the intended transmitter to the sum of the noise and the interfering power received from other transmitters [7, 22]. The SINR not only determines whether the reception is successful at all, but also determines the *capacity* of the wireless link. In practice, wireless devices may switch to a more robust modulation with a lower symbol rate in the case of a lower SINR.

In Section 4, we have presented an approximation algorithm where we have used so-called contributions $C(v, I)$. Let us redefine $C(v, I)$ as the (approximate) capacity of the wireless link v, assuming that the set of active transmissions is $I \subseteq V$. We no longer have a conflict graph G but we still require that each device is embedded in the two-dimensional space and that the bounded density of the devices holds.

The assumptions on $C(v, I)$ made in Section 4 can now be rephrased as follows:

- We can obtain the estimate $C(v, I)$ by some efficient algorithm.

- If the link v is not active, no data is transmitted; the capacity is zero.

- If the link v is active, activating other transmissions cannot increase the capacity $C(v, I)$.

- Activating transmitters which are far from the receiver of v do not affect the capacity $C(v, I)$.

The first assumption is reasonable even if we base our estimates on real measurements on the received power. For example, we can measure and tab-

ulate the received power for each transmitter–receiver pair (there is a polyno-
mial number of these), and we can use this information to estimate the SINR
at the receiver of the link v, assuming the links I are active. Then we can use
the SINR to determine the capacity $C(v, I)$.

The next two assumptions are clear, but the fourth assumption — locality
of interference — may call for some justification. Intuitively, we assume that
even the joint effect of all interfering transmitters very far from us all around
the world is insignificant in comparison with the noise when we estimate the
SINR. A theoretical justification can be derived if we assume that, due to
obstacles, the received power is approximately proportional to the distance
raised to the power a for some value $a > 2$.

With this reinterpretation of $C(v, I)$, our approximation algorithm from
Section 4 is able to find a set of radio transmissions which approximately
maximises the weighted sum of link capacities. We are no longer confined
to pairwise interference but we consider the joint effect of interfering trans-
mitters close to the receiver; furthermore, we allow for networks where each
link is able to operate at different bit rates depending on the interference and
noise. The observations in Sections 6 and 7 also generalise to this setting —
we are able to approximate link scheduling and joint scheduling and routing
even under this generalisation.

9. SCALABLE DISTRIBUTED APPROXIMATION SCHEME

The existence and nonexistence of centralised approximation algorithms
gives us insight into the computational complexity of fundamental network
coordination tasks. However, centralised algorithms are hardly practical in
very large-scale wireless networks, and approximation within large constant
factors may not be an attractive alternative to, say, time-proven heuristics in
real-world tasks.

To conclude this work, we have a look at the possibility of constructing
a family of efficient *distributed* algorithms for the link scheduling problem
with which we can achieve *any* constant approximation guarantee $1 + \varepsilon$ where
$\varepsilon > 0$. By Theorem 14, further assumptions on the structure of the input are
needed beyond the assumptions (i) and (ii) mentioned in Definition 1. In
particular, we focus on the following assumption, bounded range of radio
transmissions:

(iii) For all $v \in V$ it holds that $d(\tau(v), \rho(v)) < 1$.

We immediately obtain a centralised PTAS for this restricted variant, as we can simply omit all parts that deal with the "long links" in the approximation algorithm and its analysis in Section 4. However, obtaining a highly scalable distributed algorithm is less straightforward. First, the shifting strategy that we used to obtain the approximation algorithm for MWIS has some inherent centrality [23]; for example, in our algorithm the final step involves comparing all candidate solutions $I \in \mathcal{N}$ and choosing the best one. Second, even if we could overcome this hurdle, we would still need to turn Corollary 13 into its distributed counterpart in order to obtain a distributed approximation algorithm for link scheduling. Fortunately, a more direct alternative is available: we may be able to solve link scheduling by a distributed approximation algorithm *without* solving MWIS as an intermediate step. Before presenting the algorithm, we need to introduce the distributed setting.

In the centralised setting, we have assumed that the conflict graph G is available at some central location; we have ignored the details of gathering this information and propagating it in some communication network. In the distributed setting, we cannot ignore these details any more; we need to make the communication infrastructure explicit, as well as the computational agents who are involved in the distributed computation.

Let us first choose the computational agents. For simplicity, we assume that each vertex $v \in V$ is an autonomous computational agent. In practice, the computation related to the wireless link $v \in V$ could be realised, for example, at the transmitter $\tau(v)$. We assume that the agent v knows the coordinates of $\tau(v)$ and $\rho(v)$ as well as its neighbours in the graph G.

Then we need to introduce the communication infrastructure which enables the vertices $v \in V$ to communicate with each other. We model the communication infrastructure as a graph $H = (V, E_H)$ where an edge $\{u, v\} \in E_H$ indicates that the computational agents u and v can directly exchange messages with each other.

We do not assume that the structure of G and H is directly connected with each other — both are graphs on the same set of vertices, but we do not assume that an edge in G implies an edge in H or vice versa. However, we need to make *some* assumptions on the graph H so that the agents can efficiently co-operate. For the sake of concreteness, we make the following assumption on the structure of H.

(iv) There is a constant $\sigma > 0$ such that the graph H is a geometric σ-spanner in the following sense: $d_H(u, v) \leq \sigma \lceil d(\tau(u), \tau(v)) \rceil$ for all $u, v \in V$, where d_H is the shortest-path hop count between u and v in the communication graph H.

Intuitively, we assume that the computational agents which are physically close to each other are also able to communicate with each other through a multi-hop path with some bounded number of hops. We arrive at the problem family similar to the one studied in Floréen et al. [24].

Now for each pair (i, j), the agent v can determine whether $v \in X_{ij}$ or not. Furthermore, if $v \in X_{ij}$, it holds that $v \in X_{ij\beta}$ for some β. The agent v does not know the value of β, but it is able to determine which other agents $u \in V$ are contained in the same set $X_{ij\beta}$, as follows. If $u, v \in X_{ij\beta}$ then $d(u, v)$ is bounded by a constant. Therefore $d_H(u, v)$ is bounded, and v can reach u by querying all agents in its constant-radius neighbourhood in the communication graph H. By comparing $(\tau(u), \rho(u))$ with $(\tau(v), \rho(v))$, the agent v can determine if $u, v \in X_{ij\beta}$ for some β.

For each $X_{ij\beta}$, one agent $v \in X_{ij\beta}$ is chosen as the leader — we can use the coordinates to break the symmetry. The leader v finds an optimal link schedule for the subproblem $X_{ij\beta}$; the size of the subproblem is bounded by a constant. The leader communicates the schedule to all other agents in $X_{ij\beta}$.

Next we apply the same strategy as presented for activity scheduling in Floréen et al. [25, §4.3]. We proceed in short time steps, by using globally synchronised clocks. We iterate repeatedly through k^2 different pairs (i, j); for each time step, we fix a particular pair (i, j). We execute the local solutions of $X_{ij\beta}$ for any β independently in parallel; the lengths of the local solutions are scaled down to span exactly one time step.

The schedule obtained this way is feasible: for any pair (i, j), at any point in time, the set of active links within each $X_{ij\beta}$ forms an independent set, and there are no edges in the conflict graph between $X_{ij\beta}$ and $X_{ij\beta'}$ for $\beta \neq \beta'$.

To see that the schedule is near-optimal, interpret the subproblem $X_{ij\beta}$ as being derived from the original problem by removing some constraints. Therefore the optimum of the subproblem is at most as high as the optimum of the original problem. If $v \in X_{ij\beta}$ for some β, that is, $v \in X_{ij}$, then the relative amount of time that the link v is active is at least as high as in the global optimum of the original problem. It suffices to show that for any given link v, only a small fraction of pairs (i, j) are such that $v \notin X_{ij}$. This is immediate if k is large; cf. Lemma 10.

Thus the entire network executes a link schedule which is valid and which is able to satisfy all data transmission needs $r(v)$ in a near-optimal time. At no point we need to gather the full problem instance or the full solution at any central location: the input for a particular agent was locally available, and the output was used locally to schedule data transmissions.

In summary, we have obtained a distributed approximation scheme for link scheduling. Moreover, the algorithm is *local* [26, 27]: the decision of each agent only depends on its constant-radius neighbourhood in the graph H. To obtain this result, we exploited the assumptions (i), (ii) and (iii) on the conflict graph G, and the assumption (iv) on the communication graph H.

Acknowledgements

References

[1] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. *Wireless Networks*, 11(4):471–487, 2005.

[2] Klaus Jansen. Approximate strong separation with application in fractional graph coloring and preemptive scheduling. *Theoretical Computer Science*, 302(1–3):239–256, 2003.

[3] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *Proc. 42nd Annual Symposium on Foundations of Computer Science (FOCS, Las Vegas, NV, USA, October 2001)*, pages 538–546, Los Alamitos, CA, USA, 2001. IEEE Computer Society Press.

[4] Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[5] Subhash Khot. Improved inapproximability results for MaxClique, chromatic number and approximate graph coloring. In *Proc. 42nd Annual Symposium on Foundations of Computer Science (FOCS, Las Vegas, NV, USA, October 2001)*, pages 600–609, Los Alamitos, CA, USA, 2001. IEEE Computer Society Press.

[6] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[7] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, Cambridge, UK, 2005.

[8] Bhaskar Krishnamachari. *Networking Wireless Sensors*. Cambridge University Press, Cambridge, UK, 2005.

[9] Jukka Suomela. Approximability of identifying codes and locating-dominating codes. *Information Processing Letters*, 103(1):28–33, 2007.

[10] Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*. The Mathematical Association of America, Washington, DC, USA, 1984.

[11] Magnús M. Halldórsson. Approximations of independent sets in graphs. In *Proc. Workshop on Approximation Algorithms for Combinatiorial Optimization Problems (APPROX, Aalborg, Denmark, July 1998)*, volume 1444 of *Lecture Notes in Computer Science*, pages 1–13, Berlin, Germany, 1998. Springer-Verlag.

[12] Magnús M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *Journal of Graph Algorithms and Applications*, 4(1):1–16, 2000.

[13] Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34(6):1302–1323, 2005.

[14] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.

[15] Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.

[16] Harry B. Hunt, III, Madhav V. Marathe, Venkatesh Radhakrishnan, S. S. Ravi, Daniel J. Rosenkrantz, and Richard E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238–274, 1998.

[17] Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28(1):164–191, 1999.

[18] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[19] V. Chvátal and C. Ebenegger. A note on line digraphs and the directed max-cut problem. *Discrete Applied Mathematics*, 29(2–3):165–170, 1990.

[20] Thomas Erlebach and Klaus Jansen. Conversion of coloring algorithms into maximum weight independent set algorithms. *Discrete Applied Mathematics*, 148(1):107–125, 2005.

[21] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, Germany, 2nd edition, 1993.

[22] Claude E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. Reprinted in *Proceedings of the IEEE*, 86(2):447–457, 1998.

[23] Fabian Kuhn, Tim Nieberg, Thomas Moscibroda, and Roger Wattenhofer. Local approximation schemes for ad hoc and sensor networks. In *Proc. Joint Workshop on Foundations of Mobile Computing (DIALM-POMC, Cologne, Germany, September 2005)*, pages 97–103, New York, NY, USA, 2005. ACM Press.

[24] Patrik Floréen, Petteri Kaski, and Jukka Suomela. A distributed approximation scheme for sleep scheduling in sensor networks. In *Proc. 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON, San Diego, CA, USA, June 2007)*, pages 152–161, Piscataway, NJ, USA, 2007. IEEE.

[25] Patrik Floréen, Petteri Kaski, Topi Musto, and Jukka Suomela. Local approximation algorithms for scheduling problems in sensor networks. In *Proc. 3rd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors, Wrocław, Poland, July 2007)*, volume 4837 of *Lecture Notes in Computer Science*, pages 99–113, Berlin, Germany, 2008. Springer-Verlag.

[26] Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[27] Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995.

[28] Petteri Kaski, Aleksi Penttinen, and Jukka Suomela. Coordinating concurrent transmissions: A constant-factor approximation of maximum-weight independent set in local conflict graphs. In *Proc. 6th International Conference on Ad-Hoc Networks & Wireless (AdHoc-NOW, Morelia, Mexico, September 2007)*, volume 4686 of *Lecture Notes in Computer Science*, pages 74–86, Berlin, Germany, 2007. Springer-Verlag.