Computer Networks 99 (2016) 125–133

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Scheduling wireless links by vertex multicoloring in the physical interference model

Fabio R.J. Vieira*, José F. de Rezende, Valmir C. Barbosa

Programa de Engenharia de Sistemas e Computação, COPPE, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21941-972 Rio de Janeiro, RJ, Brazil

ARTICLE INFO

Article history:
Received 21 October 2015
Revised 3 February 2016
Accepted 4 February 2016
Available online 17 February 2016

Keywords:
Wireless mesh networks
Link scheduling
Physical interference model
Vertex coloring
Vertex multicoloring

ABSTRACT

Scheduling wireless links for simultaneous activation in such a way that all transmissions are successfully decoded at the receivers and moreover network capacity is maximized is a computationally hard problem. Usually it is tackled by heuristics whose output is a sequence of time slots in which every link appears in exactly one time slot. Such approaches can be interpreted as the coloring of a graph's vertices so that every vertex gets exactly one color. Here we introduce a new approach that can be viewed as assigning multiple colors to each vertex, so that, in the resulting schedule, every link may appear more than once (though the same number of times for all links). We report on extensive computational experiments, under the physical interference model, revealing substantial gains for a variety of randomly generated networks.

1. Introduction

Let L be a set of wireless links, each link $i \in L$ characterized by a sender node s_i and a receiver node r_i. Depending on the spatial disposition of such nodes, activating more than one link simultaneously creates interference that may hamper the receivers' ability to decode what they receive. In the physical interference model [1], the chief quantity governing receiver r_i's ability to decode what it receives from s_i when all links of a set S containing link i are active is the signal-to-interference-and-noise ratio (SINR), given by

$$\mathrm{SINR}(i, S) = \frac{P/d^{\alpha}_{s_i r_i}}{N + \sum_{j \in S \setminus \{i\}} P/d^{\alpha}_{s_j r_i}}, \tag{1}$$

where P is a sender's transmission power (assumed the same for all senders), N is the noise floor, d_{ab} is the Euclidean distance between nodes a and b, and $\alpha > 2$ determines the law of power decay with Euclidean distance. We say that a nonempty subset S of L is *feasible* if no two of its members share a node (in case $|S| > 1$) and moreover $\mathrm{SINR}(i, S) \geq \beta$ for all $i \in S$, where β is a parameter related to a receiver's decoding capabilities (assumed the same for all receivers) and is chosen so that $\beta > 1$.

Several strategies have been devised to maximize network capacity, either through the self-contained scheduling of the links in L for activation [2–20] or by combining link scheduling with other techniques [21–29]. All these strategies revolve around formulations as NP-hard optimization problems, so all rely on some form of heuristic procedure drawing inspiration from various sources, some more of an intuitive nature [3–7,9,10,12–17,19–24,26,28], others more formally grounded on graph-theoretic notions [2,8,11,18,29]. Often the problem is formulated in a spatial time-division multiple access (STDMA) framework, that is, assuming essentially that time is divided into time slots, each one accommodating a certain number of simultaneous link activations. In this case, the problem is to find T feasible subsets of L, here denoted by $S_1, S_2, \ldots, S_T$, minimizing T while ensuring that every link appears in exactly one of the T subsets.

* Corresponding author.
E-mail address: fabiorjvieira@gmail.com (F.R.J. Vieira).

http://dx.doi.org/10.1016/j.comnet.2016.02.008

There is a sense in which this formulation can be interpreted in the context of coloring a graph's vertices. To see this, first recall that to color a graph's vertices is to partition them into independent sets (that is, into vertex subsets that contain no two neighbors in the graph), each of these sets corresponding to a color. An optimal vertex coloring is obtained when no partition into fewer independent sets is possible. Depending on the application at hand, and letting $\mathcal{I}$ denote the set of all the graph's independent sets, it may be necessary to rule out some of the members of $\mathcal{I}$, i.e., to forbid their appearance in any partition. In the specific case of scheduling the links in L for simultaneous activation, we begin by defining a basic conflict graph, denoted by C, whose set of vertices is the set of links L and whose set of edges consists of all pairs of links that are not feasible sets. That is, the edge set of C contains all link pairs (x, y) such that $\{s_x, r_x\} \cap \{s_y, r_y\} \neq \emptyset$ (links x and y have nodes in common), or $\mathrm{SINR}(x, \{x, y\}) < \beta$, or $\mathrm{SINR}(y, \{x, y\}) < \beta$. If we attempted to schedule the links in L by coloring the vertices of graph C and using each of the resulting independent sets as the set of links to be scheduled in each time slot, clearly some independent set S with $|S| > 2$ might turn up as part of the solution such that $\mathrm{SINR}(x, S) < \beta$ for some link $x \in S$. Such a schedule would not do, so we must further restrict the independent sets that the partitioning for vertex coloring can choose from, specifically by forbidding any independent set that is not feasible.

With this notion of a generalized form of graph coloring in place, the schedule given by the sequence $S_1, S_2, \ldots, S_T$ of feasible link sets can be regarded as the product of coloring the vertices of graph C with T colors in such a way that all vertices in S_k get color k. This interpretation suggests a further generalization, now allowing every link to appear not in exactly one of the T subsets but in any number of them, provided this number is the same for all links. What we have now is no longer simply our generalized form of vertex coloring, but a generalized form of vertex multicoloring. To multicolor the vertices of a graph in this generalized sense, and assuming that $q \geq 1$ is the number of distinct colors to be assigned to each vertex, is to identify a certain number of independent sets of graph C (avoiding the forbidden ones) such that every vertex belongs to exactly q of them. To do so optimally is no longer to minimize the total number of colors, but rather to minimize the ratio of such a number to q. Returning to the scheduling context, we no longer look to minimize the number T of time slots, but look instead for the values of T and q that minimize T/q. Now the schedule $S_1, S_2, \ldots, S_T$ of feasible link sets can be regarded as resulting from multicoloring the vertices of C with T colors in such a way that color k is assigned to all vertices in S_k and that every vertex receives exactly q distinct colors.[1]

[1] The reader familiar with the theory of hypergraphs will notice that forbidding independent sets of graph C while coloring or multicoloring its vertices is equivalent to coloring or multicoloring, respectively, the vertices of a hypergraph [30]. In this hypergraph, the vertices are the same as in graph C and the hyperedges are the nonempty subsets of links that are not feasible (and therefore include those pairs of links that are edges in C).

The potential advantages of this multicoloring-based formulation are tantalizing. If the original formulation leads to a number T of time slots while the new one leads to $T' > T$ time slots for some $q > 1$, the latter schedule is preferable to the former, even though it requires more time slots, provided only that $T'/q < T$ (or $qT > T'$). To see that this is so, first note that the longer schedule promotes an overall number of link activations given by $q|L|$ in T' time slots. In order for the shorter schedule to achieve this same number of activations, it would have to be repeated q times in a row, taking up $qT > T'$ time slots.

The possibility of multicoloring-based link scheduling in the physical interference model seems to have been overlooked so far, despite the recent demonstration of its success in the protocol-based interference model [29]. Here we introduce a heuristic framework to obtain multicoloring-based schedules from the single-color schedules produced by any rank-based heuristic (i.e., one that decides the time slot in which to activate a given link based on how it ranks relative to the others with respect to some criterion; cf., e.g., [3,5,9,11,13,17,20]). We use two iconic single-color heuristics (GreedyPhysical [3], for its simplicity, and ApproxLogN [9,17], for its role in establishing new bounds on network capacity), as well as a third one that we introduce in response to improvement opportunities that we perceived in the former two. Incidentally, the latter heuristic, called MaxCRank, is found to perform best both as a stand-alone, single-color strategy and as a base for the multicoloring scheme. All three single-color heuristics run in time polynomial in $|L|$.

Before continuing, we note that the problem we address, that of maximizing network capacity by link scheduling in the physical interference model, though the same as the one considered in [1,3,9,17], is only one of a great variety of problems that likewise must face the many constraints imposed by the need to circumvent the effects of electromagnetic interference in wireless networks. Such problems relate to various aspects of network design, such as node placement [31–34] and frequency assignment [35,36], to name two prominent ones. Some of them take into account some form of end-to-end communication demand [31–35], while others, as in our case in this paper, do not [36]. In a similar vein, the adoption of vertex-coloring and -multicoloring notions to inform our approach is by no means exclusive. In fact, often the proposed solutions to those related problems are closely based on some form of vertex coloring [34–36], including in the case of [35] – and, incidentally, of GreedyPhysical with non-unit demands [3] as well – the possibility of assigning more than one color to the same vertex (though not in as strict a meaning of vertex multicoloring as the one we adopt, since in those cases the number of distinct colors to be assigned to each vertex is fixed beforehand, as opposed to being part of the solution).

We proceed by first discussing single-color schedules in Section 2, where the three heuristics mentioned above are explained in relation to a single overarching template. Then we move to multicoloring-based schedules in Section 3, introducing our heuristic framework for single-color schedules to be automatically turned into multicoloring-based ones. Our computational results are given in Sections 4 and 5, which explain our experimental setup and present the results proper, respectively. Conclusions appear in Section 6.

2. Single-color schedules

Rank-based heuristics for single-color scheduling are usually monotonic, in the sense that first S_1 is determined, then S_2 out of the set R of links that remain to be scheduled, then S_3 out of a smaller R, and so on, until R becomes empty. Choosing a link to add to the current S_k depends on the feasibility of the resulting set and also on a ranking criterion that is specific to each heuristic. The ranking criterion establishes the order in which the links in R are to be considered for inclusion in S_k.

The following is the general outline of such a heuristic.

S1. Let $k := 1$, $S_k := \emptyset$, and $R := L$. Order R according to the ranking criterion.
S2. If a link $i \in R$ exists such that $S_k \cup \{i\}$ is feasible, then move the top-ranking such i from R to S_k and go to Step S3. If none exists, then let $k := k + 1$, $S_k := \emptyset$, and go to Step S2.
S3. If $R \neq \emptyset$, then reorder R according to the ranking criterion and go to Step S2.
S4. Let $T := k$ and output $S_1, S_2, \ldots, S_T$.

Steps S1–S4 amount to scanning the set R of unscheduled links and moving to the current S_k (in Step S2) the top-ranking link $i \in R$ whose inclusion in S_k preserves feasibility. Whenever such a move does occur, an opportunity is presented for R to be reordered (in Step S3) according to the ranking criterion.

Assessing the time complexity of this heuristic hinges on two main observations. The first one is that every feasibility test, in Step S2, can be performed in $O(|L|^2)$ time, since both checking all link pairs for disjointness and accounting for the summation in the denominator of Eq. (1) for all pertinent links require quadratic time in $|S_k|$. The other observation is that, in the absence of any further information as part of the input to the problem, ordering or reordering the current R, respectively in Steps S1 and S3, requires $f(|R|) + O(|R| \log |R|)$ time, where $f(|R|)$ is the time needed to evaluate how each link in R stands with respect to the ranking criterion (i.e., to compute the number that will be used for each link by the sorting procedure) and $O(|R| \log |R|)$ is the corresponding time complexity of sorting by comparison [37]. Below we rely on these two observations when considering a heuristic's time complexity.

It is easy to see that both GreedyPhysical and ApproxLogN can be cast in the above sequence of steps in a straightforward manner. The ranking criterion for GreedyPhysical is nonincreasing and refers, for link i, to the number of links in L with which i can never share a time slot; that is, links $j \in L\backslash\{i\}$ such that $\{i, j\}$ is not feasible. It is then an immutable ranking criterion and consequently the reordering in Step S3 is moot. We have that $f(|L|)$ is $O(|L|^2)$, so this is the time complexity of Step S1 for GreedyPhysical. This does not impact the time complexity of the whole procedure, though, since in this case the remaining steps can be implemented to run in $O(|L|^3)$ time.

As for ApproxLogN, its ranking criterion is nondecreasing and refers to the Euclidean distance between the sender and the receiver in each link. This criterion, too, is fixed and as such renders the reordering in Step S3 once again moot.[2] It thus follows that $f(|L|)$ is $O(|L|)$, so the time complexity of Step S1 is $O(|L| \log |L|)$. As in the case of GreedyPhysical, the remaining steps, and therefore the whole procedure, can be implemented to complete in $O(|L|^3)$ time.

We now introduce a new heuristic that can also be viewed as instantiating Steps S1–S4, but with a ranking criterion that is both more stringent than the two just described and also inherently dynamic, thus justifying the reordering in Step S3. We call it MaxCRank to highlight its core principle, which is to maximize the number of links in R that still have a chance of joining the current S_k (i.e., remain "Candidates") once a decision is made on which one of them, say i, is to be moved from R to S_k. The corresponding ranking criterion is nonincreasing and refers to the number of links $j \in R\backslash\{i\}$ for which $S_k \cup \{i, j\}$ is feasible. That is, the link i that MaxCRank moves from set R to set S_k in Step S2 is the one that maximizes the number of links in $R\backslash\{i\}$ such that $S_k \cup \{i, j\}$ is feasible.

Algorithmically, applying the ranking criterion of MaxCRank in Steps S1 and S3 can be achieved as follows. For each link $i \in R$ for which $S_k \cup \{i\}$ is feasible, calculate the number of links $j \in R\backslash\{i\}$ for which $S_k \cup \{i, j\}$ remains feasible. The required ordering (or reordering) of R can then be achieved by sorting its links in a nonincreasing order of these numbers. To the best of our knowledge, MaxCRank is the first rank-based heuristic to reorder R in Step S3.

During Step S1, MaxCRank can be regarded as doing the exact opposite of GreedyPhysical, since the link i that it ranks first is the one maximizing the number of links $j \in L\backslash\{i\}$ such that $\{i, j\}$ is feasible. It follows that the time complexity of Step S1 is the same as that in GreedyPhysical, $O(|L|^2)$. Each reordering in Step S3 is such that $f(|R|)$ is $O(|L|^4)$, and therefore so is the overall time complexity of the reordering itself. It follows that MaxCRank can be implemented to run in $O(|L|^5)$ time.

3. Multicoloring-based schedules

The link sets $S_1, S_2, \ldots, S_T$ output by Steps S1–S4 of Section 2 promote a number of link activations given by $|L|$, one activation per link. If this schedule were to be repeated q times in a row for some $q > 1$, the total number of link activations would grow by a factor of q and so would the number of time slots used. That the same growth law should apply both to how many links are activated and to how many time slots elapse indicates that the most basic scheduling unit is $S_1, S_2, \ldots, S_T$ itself, not any number of repetitions thereof.

However, activating the links in S_1 the second time around does not necessarily have to be restricted to time slot $T + 1$. Instead, it may be possible to take advantage of

[2] ApproxLogN replaces the requirement of feasibility in Step S2 by conditions that are sufficient for it to be satisfied. This is done to make sure that certain algorithmic performance guarantees hold, but that is of no concern to us here.

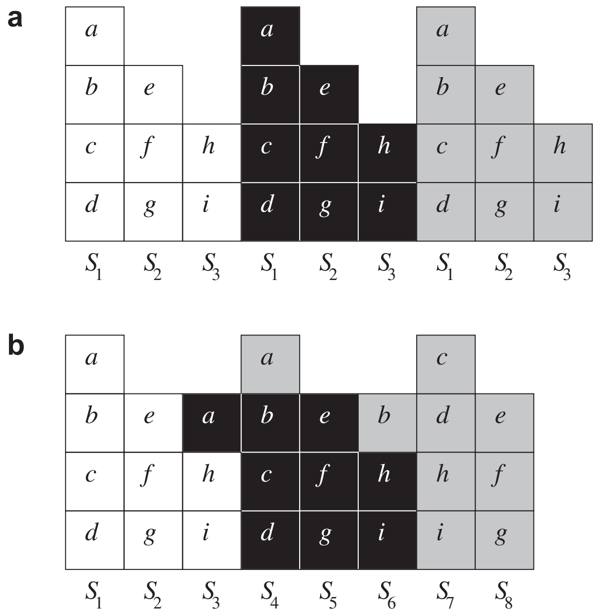


Fig. 1. Two scheduling possibilities for the link set $L = \{a, b, c, d, e, f, g, h, i\}$. The first possibility (a) is based on continually repeating the single-color schedule $S_1 = \{a, b, c, d\}$, $S_2 = \{e, f, g\}$, and $S_3 = \{h, i\}$ (three repetitions shown). The second possibility (b) is likewise based on repetitions, but now the basic scheduling unit is the multicoloring-based schedule $S_1 = \{a, b, c, d\}$, $S_2 = \{e, f, g\}$, $S_3 = \{a, h, i\}$, $S_4 = \{a, b, c, d\}$, $S_5 = \{e, f, g\}$, $S_6 = \{b, h, i\}$, $S_7 = \{c, d, h, i\}$, and $S_8 = \{e, f, g\}$ (one repetition shown).

some room left in previous time slots for at least one of the links in S_1. With this type of precaution in mind, advancing link activations in such a manner might result in a sequence of link sets $S_1, S_2, \ldots, S_{T'}$ containing exactly q activations of every link in L for some $q > 1$ but with $T < T' < qT$. Clearly, in this case the most basic scheduling unit would be $S_1, S_2, \ldots, S_{T'}$, not $S_1, S_2, \ldots, S_T$ any more. Not only this, but the new basic scheduling unit would be preferable to the previous one, since a total of $q|L|$ link activations would be attainable in fewer time slots (T' rather than qT).

This is illustrated in Fig. 1, where panel (a) depicts three repetitions of a single-color schedule for which $T = 3$ and panel (b) depicts a single repetition of a multicoloring-based schedule for which $q = 3$ and $T' = 8$. The latter is therefore preferable, since in both cases all links are activated the same number of times (viz., 3) and $8 = T' < qT = 9$.

A heuristic to find the $q > 1$ with $T' < qT$ for which T'/q is smallest, if any exists, is simply to wrap Steps S1–S4 in an outer loop that iterates along with $q = 1, 2, \ldots$ while preventing S_k from being reset to $\emptyset$ any later than the first time it is considered. At the end of each iteration, say the qth, the value of T' is updated (to the number of time slots elapsed since the beginning) and the ratio T'/q is computed. The iterations continue while this ratio is strictly decreasing. At the end of the first iteration we get $T' = T$, but successful further iterations will produce a sequence of strictly decreasing T'/q values.

This heuristic is given as the following steps.

M1. Let $q := 1$, $k := 1$, $S_k := \emptyset$, and $R := L$. Order R according to the ranking criterion.

M2. If a link $i \in R$ exists such that $S_k \cup \{i\}$ is feasible, then move the top-ranking such i from R to S_k and go to Step M3. If none exists, then let $k := k + 1$, let $S_k := \emptyset$ if S_k is being considered for the first time, and go to Step M2.

M3. If $R \neq \emptyset$, then reorder R according to the ranking criterion and go to Step M2.

M4. If $q = 1$, then let $T := k$. If $q > 1$, then let $X := T'$. Let $T' := k$.

M5. If $q = 1$, or $q > 1$ and $T'/q < X/(q-1)$, then let $q := q + 1$, $k := 1$, and $R := L$; order R according to the ranking criterion and go to Step M2.

M6. Undo all additions to $S_1, S_2, \ldots, S_X$ during the qth iteration, let $T' := X$, and let $q := q - 1$. Output $S_1, S_2, \ldots, S_{T'}$ and q.

Steps M2 and M3 are essentially equivalent to Steps S2 and S3, respectively, of the single-color case, but now encased by further instructions to control not only the progress of k (the current time slot under consideration), but also that of q (the number of colors per link). For this reason, the heuristic given by Steps M1–M6 continues to depend on the choice of a rank-based, single-color heuristic to lie at its core. Just as with Steps S1–S4, any mention to reordering R is moot for both GreedyPhysical and ApproxLogN.

To see that Step M6 is eventually reached, consider that whenever the value of q is incremented in Step M5, right before that increment we have a multicoloring of the conflict graph C requiring a total of T' colors and assigning q of them to each vertex. A sequence of rationals $T'_1/1, T'_2/2, \ldots$ ensues and Steps M2–M5 are repeated while it is strictly decreasing. If none of the corresponding multicolorings of C had to avoid certain independent sets as explained in Section 1, then by definition this sequence of rationals would have as an infimum the so-called multichromatic (or fractional chromatic) number of C. By a well-known result [38], this infimum is in fact a minimum, hence a value of q would eventually be reached beyond which the sequence would no longer be strictly decreasing and the heuristic would enter Step M6. The existence of forbidden independent sets only means that such a minimum can be higher (occur earlier in the sequence, since some of the rationals T'_q/q may be higher than otherwise), so the heuristic enters Step M6 nonetheless.

A new quantity of interest is the gain G incurred by Steps M1–M6, that is, the ratio of T to the value of T'/q at the end, hence $G = qT/T'$. The least possible value of G, of course, is $G = 1$, which corresponds to the case in which the iterations reach Step M6 already for $q = 2$.

4. Experimental setup

We give results for two families of randomly generated networks, henceforth referred to as type-I and type-II networks. As will become apparent, type-I networks are more realistic, since they emerge naturally from the physical placement of the nodes. We use type-II networks as

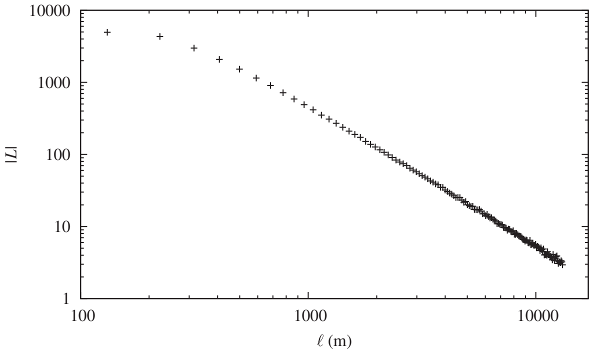


Fig. 2. Average number of links in type-I networks as a function of the square side ℓ for $n = 100$. All data points are averages over 1 000 network instances. Additional relevant parameters are $P = 300$ mW, $N \approx 8 \times 10^{-14}$ W (for a bandwidth of 20 MHz at room temperature), $\alpha = 4$, and $\beta = 25$ dB.

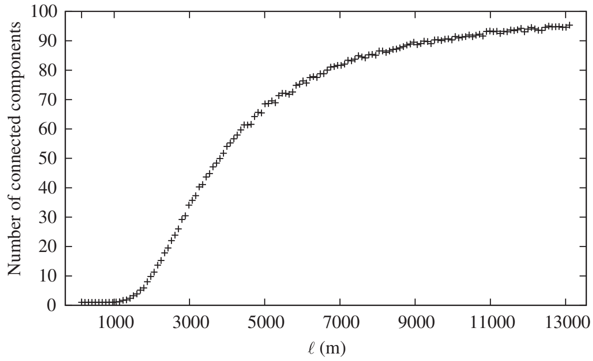


Fig. 3. Average number of connected components in type-I networks as a function of the square side ℓ for $n = 100$. All data points are averages over 1 000 network instances. Additional relevant parameters are $P = 300$ mW, $N \approx 8 \times 10^{-14}$ W (for a bandwidth of 20 MHz at room temperature), $\alpha = 4$, and $\beta = 25$ dB.

well because they were used in the performance evaluation of ApproxLogN [9,17] and thus provide a more direct basis for comparison. A network's number of nodes is henceforth denoted by n.

A type-I network is generated by first placing all n nodes inside a square of side ℓ uniformly at random. A node's neighbors are then determined as a function of the value of $d_{s_i r_i}$ for which $\text{SINR}(i, \{i\}) = \beta$, that is, as a function of the greatest distance between sender and receiver that still allows successful decoding upon reception when no other transmission interferes. By Eq. (1), denoting such a distance by ρ yields $\rho = (P/\beta N)^{1/\alpha}$, so a node's neighbor set is the set of nodes to which the Euclidean distance does not surpass ρ. Any two nodes that are neighbors of each other according to this criterion become a link in L, sender and receiver being decided uniformly at random (so that a node may, e.g., be the sender in a link and the receiver in another). For fixed n, increasing ℓ causes the number of links, $|L|$, to decrease precipitously, though in the heavy-tailed manner of an approximate power law (Fig. 2). It also causes the network's number of connected components to increase from about 1 to nearly n (a component per node) through a sharp transition in between (Fig. 3).

In a type-II network, the number n of nodes is necessarily even. Of these, $n/2$ are senders and $n/2$ are receivers. Unlike the case of a type-I network, in a type-II network no node may appear in more than one link (so, in particular, no node is the sender in a link and the receiver in another). A type-II network is generated by first placing the receivers uniformly at random inside a square of side ℓ and then, for each receiver, placing the corresponding sender inside a circle of radius ρ centered at it, also uniformly at random. A type-II network has $|L| = n/2$, which is also its number of connected components. Varying ℓ affects interference only.

Sample networks of both types are given in Fig. 4, where panels (a) and (b) refer to type-I networks and panels (c) and (d) refer to type-II networks. For each network type, both an example with low link density (number of links per unit area) and one with high link density are given. As will become apparent in Section 5, the two type-I samples refer to ℓ values that are only two out of many that were used. As for the type-II samples, the two values of $|L|$ are the smallest ones used by the authors of ApproxLogN when evaluating it computationally [9,17]. We follow their choices of $|L|$ in our own evaluation in Section 5. In either case (type-I or -II networks), the reader is to note that not all parameterizations lead to scenarios likely to be encountered in practice. Many are meant, rather, solely as means to stretch the networks' possibilities when it comes to capacity.

5. Results

We give results for all three single-color heuristics mentioned in Section 2, namely GreedyPhysical, ApproxLogN, and MaxCRank, and also for their multicoloring-based versions, obtained as explained in Section 3. These results are given as $T/|L|$ in the former case (the normalized schedule length, since $|L|$ is a clear upper bound on T), and as the gain G in the latter.

The data in Fig. 5 refer to type-I networks and as such are given as a function of the square side ℓ. The number of nodes is fixed throughout (at $n = 100$), so the networks get sparser (fewer links, more connected components) as ℓ is increased. In the single-color cases (panel (a) of the figure), all three heuristics start out with $T = |L|$ for the very dense networks (very small ℓ), but smaller densities quickly reduce interference so that T falls significantly below $|L|$. MaxCRank is the best performer throughout, followed by GreedyPhysical and ApproxLogN. The value of $T/|L|$ for MaxCRank can be as low as about 73% of the value for GreedyPhysical and about 41% of the value for ApproxLogN.

As for the heuristics' multicoloring-based versions (panel (b)), there is practically no gain for the densest networks, but again this is reversed as interference abates with increasing ℓ. In fact, G reflects an improvement by up to about 32% for MaxCRank, up to about 25% for GreedyPhysical, and up to about 10% for ApproxLogN. Not only in these relative terms, but also in absolute terms, MaxCRank is still the top performer and ApproxLogN the bottom one

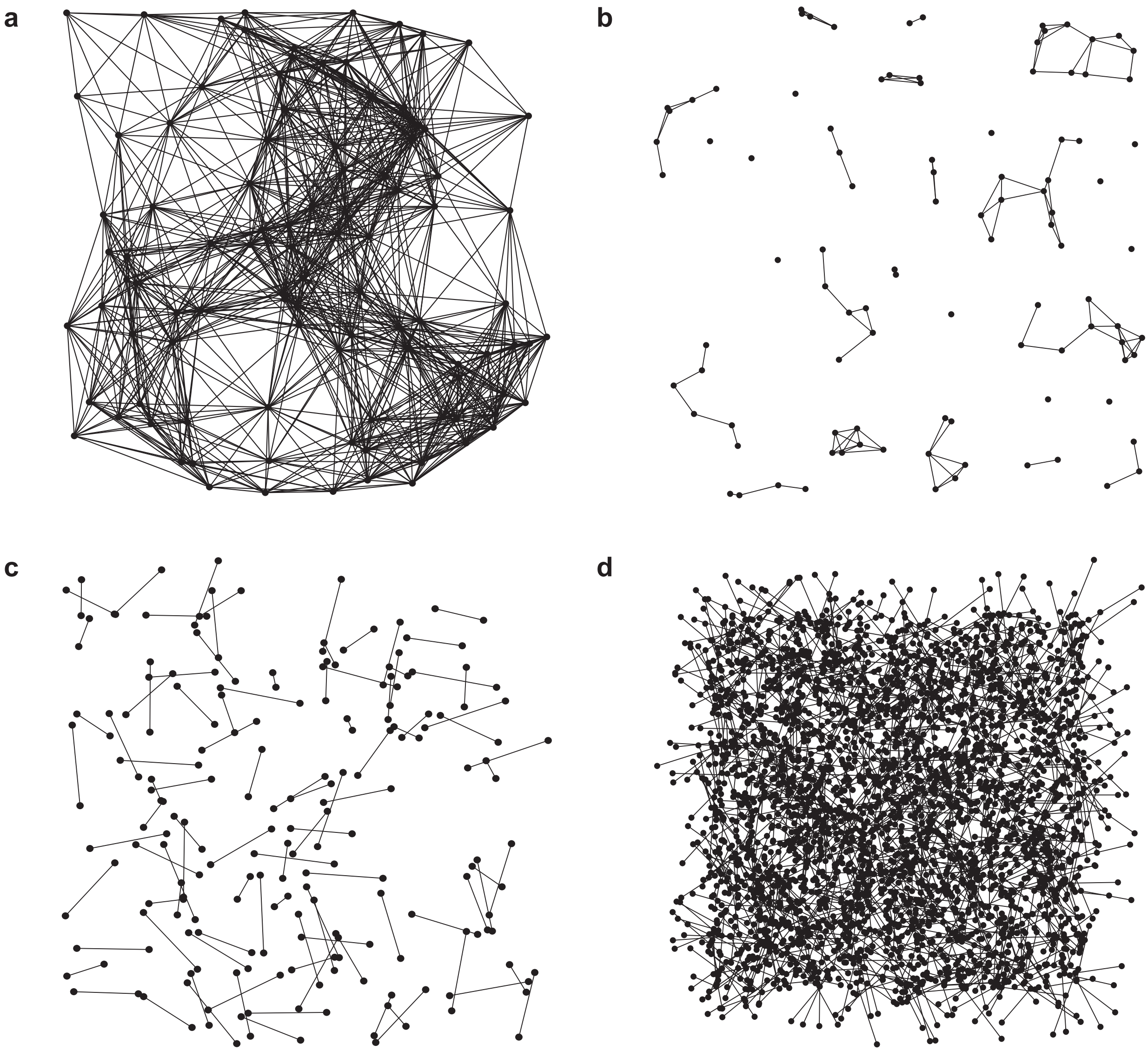


Fig. 4. Two sample networks of type I and two of type II. The type-I samples are given for $\ell = 524$ m (a) and $\ell = 2\,096$ m (b), with $n = 100$ in both cases. The type II samples are given for $|L| = 100$ (c) and $|L| = 3\,200$ (d), with $\ell = 1\,000$ m in both cases. All networks are represented as undirected graphs, with vertices standing for network nodes and edges standing for links. For each link, the corresponding sender and receiver nodes are unidentified, for visual clarity. Additional relevant parameters are $P = 300$ mW, $N \approx 8 \times 10^{-14}$ W (for a bandwidth of 20 MHz at room temperature), $\alpha = 4$, and $\beta = 25$ dB.

(in fact, the only of the three heuristics for which $G = 1$ is sometimes attained).

The results for type-II networks, given in Fig. 6, are presented as a function of $|L|$, the number of links. Because ℓ is fixed throughout (at $\ell = 1\,000$ m), increasing $|L|$ causes the impact of accumulated interference to be felt more severely. One consequence of this is that, for the single-color heuristics (panel (a) of the figure), T increases almost linearly with $|L|$. The three heuristics stand relative to one another as they do for type-I networks, with the value of $T/|L|$ for MaxCRank being as low as about 84% of that for GreedyPhysical and 63% of the value for ApproxLogN.

Another consequence of increasing $|L|$ in type-II networks, now related to the multicoloring-based versions of the heuristics (panel (b)), is that gains above 1 are increasingly hard to come by. MaxCRank continues to be the top performer in all cases (both in relative terms with respect to the single-color heuristics and in absolute terms), followed by GreedyPhysical, then by ApproxLogN. In fact, G values indicate an improvement of about 31% for MaxCRank, about 24% for GreedyPhysical, and 9.5% for ApproxLogN.

All our network instances are characterized by the number n of nodes (fixed for type-I instances and variable as $n = 2|L|$ for type-II instances) and the number $|L|$ of links (variable as a function of the distance ρ defined in Section 4 and of the square side ℓ for type-I instances, and as one of five predetermined values for type-II instances). Finding a common ground for comparing the results summarized in Fig. 5 for type-I networks to those in Fig. 6

Table 1
Highlights from Figs. 5 and 6 for networks having approximately the same link density. Link densities are exact for the type-II networks and correspond to the five distinct values of $|L|$ in Fig. 6, for $\ell = 1\,000$ m. Link densities for the type-I networks are obtained for those values of ℓ in Fig. 5 (shown in this table's third column) for which the average $|L|$ (not shown) leads to link densities as close to those of the type-II networks as possible.

Link density		Value of ℓ for type I	GreedyPhysical		ApproxLogN		MaxCRank	
Type I	Type II		Type I	Type II	Type I	Type II	Type I	Type II
			$T/\|L\|$					
0.0001	0.0001	1 965	0.71	0.39	0.98	0.48	0.53	0.34
0.0033	0.0032	786	0.99	0.37	1.00	0.50	0.87	0.31
0.0061	0.0064	655	0.99	0.37	1.00	0.50	0.91	0.31
0.0126	0.0128	524	1.00	0.36	1.00	0.50	0.95	0.31
0.0301	0.0256	393	1.00	0.34	1.00	0.50	0.98	0.31
			G					
0.0001	0.0001	1 965	1.09	1.25	1.05	1.09	1.23	1.31
0.0033	0.0032	786	1.04	1.14	1.00	1.08	1.04	1.16
0.0061	0.0064	655	1.01	1.09	1.00	1.06	1.02	1.10
0.0126	0.0128	524	1.01	1.04	1.00	1.02	1.01	1.06
0.0301	0.0256	393	1.01	1.01	1.00	1.01	1.01	1.03

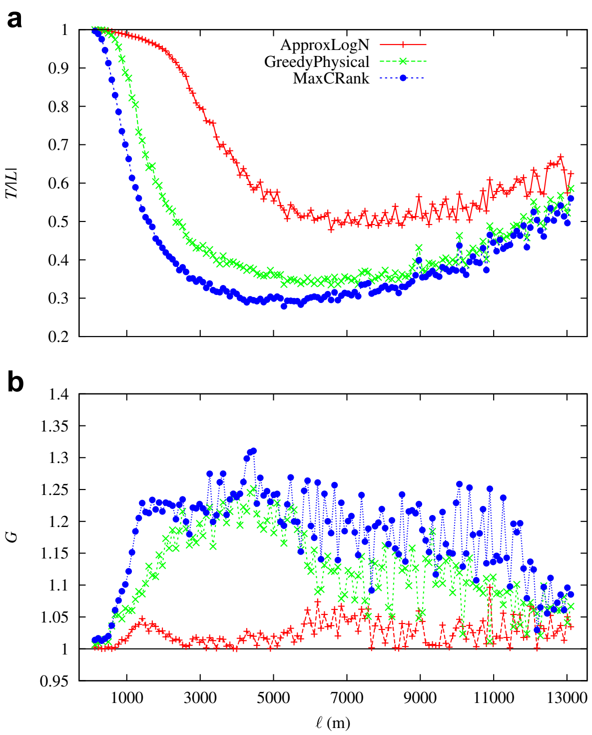


Fig. 5. Performance of GreedyPhysical, ApproxLogN, and MaxCRank on type-I networks. Data are given for the heuristics' single-color versions (a) and for their multicoloring-based versions (b). All data points are averages over 1 000 network instances. Confidence intervals are less than 1% of the mean at the 95% level, so error bars are omitted. All networks have $n = 100$ nodes. Additional relevant parameters are $P = 300$ mW, $N \approx 8 \times 10^{-14}$ W (for a bandwidth of 20 MHz at room temperature), $\alpha = 4$, and $\beta = 25$ dB.

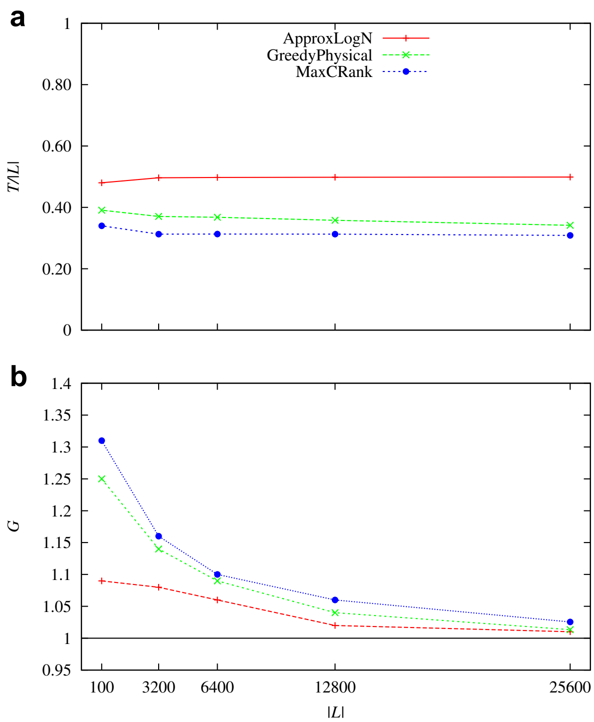


Fig. 6. Performance of GreedyPhysical, ApproxLogN, and MaxCRank on type-II networks. Data are given for the heuristics' single-color versions (a) and for their multicoloring-based versions (b). All data points are averages over 1 000 network instances. Confidence intervals are less than 1% of the mean at the 95% level, so error bars are omitted. All networks have $\ell = 1\,000$ m. Additional relevant parameters are $P = 300$ mW, $N \approx 8 \times 10^{-14}$ W (for a bandwidth of 20 MHz at room temperature), $\alpha = 4$, and $\beta = 25$ dB.

for type-II networks cannot rely on the value of n (since $n = 100$ in Fig. 5 and $n \geq 200$ in Fig. 6), but resorting to the value of $|L|$ is not only possible but also convenient, because it allows a focus on arguably one of the most important quantities as far as the effect of interference on performance is concerned: the link density, $|L|/\ell^2$.

We present such a comparison in Table 1, where the five link densities in Fig. 6 are used as a basis. For each of these densities, the value of ℓ yielding the nearest link density in Fig. 5 was identified, thus enabling the construction of the table, whose first three columns contain these link-density and ℓ values. The remaining six columns are grouped into three pairs, each pair referring to one of GreedyPhysical, ApproxLogN, and MaxCRank, each pair highlighting the data from Figs. 5 and 6 as a function of link density.

Unlike what might be expected, given the very close link densities of the type-I and type-II networks being compared in the entries of each column pair, both $T/|L|$ and G differ markedly across network types. Type-I networks are greatly inferior to their type-II counterparts when comparing $T/|L|$ values, though only somewhat inferior to them when G values are compared. These differences underscore how fundamentally different the two network types are, while emphasizing their main structural difference, which is that in type-I networks two links can involve a total of three nodes (and thereby share one of them, which can be either the sender in both links, or the receiver in both links, or the sender in one link and the receiver in the other). This, in turn, clearly impacts the choice of schedules that are available to networks of each type. After all, calling a subset of links S feasible (and therefore amenable to being scheduled in the same time slot) requires not only sufficiently high SINR(i, S) for every link $i \in S$ but also that no two links in S share a node. Therefore, given two networks of equal link density, one of them containing nodes shared by links but not the other, one should clearly expect the former to require a longer schedule (relative to its number of links) than the latter. By the same token, such schedule should be expected to be less prone to improvement by Steps M1–M6 of Section 3 than that of the latter network. This is what we find in Table 1.

6. Concluding remarks

Although it may at first seem striking that ApproxLogN has performed so poorly across most of our experiments, it should be kept in mind that this heuristic, in all likelihood, was never meant as a strong contender for single-color link scheduling. In fact, and as noted in Section 2, ApproxLogN approaches the checking of feasibility rather indirectly, verifying sufficient conditions for feasibility to hold instead of the property itself. This is bound to prevent ApproxLogN from scheduling links for activation when they could be scheduled. What must be remembered, then, is that the use of such indirect conditions has led to important performance and capacity bounds. ApproxLogN, therefore, remains an important contribution despite its performance in more practical settings.

What really is striking in our results, though, is the appearance of greater-than-1 gains practically across the board, particularly for MaxCRank or GreedyPhysical as the base, single-color heuristic. Link schedules, once determined, are meant to be used repetitively, so every link is already meant to be scheduled for activation over and over again, indefinitely. Conceptually, what our multicoloring-based wrapping of single-color heuristics tries to do is to intertwine some number of repetitions of a single-color schedule, taking up fewer time slots than the straightforward juxtaposition of the same number of repetitions of that schedule. By doing so, more link activations can be packed together in earlier time slots. As a consequence, the basic schedule to be used for indefinite repetition is now one that leads to higher network capacity and possibly higher throughput.

As we mentioned earlier, multicoloring-based link scheduling of the sort we have demonstrated has roots in the multicoloring of a graph's vertices (as well as edges, in many cases). As such, a rich body of material, relating both to computational-complexity difficulties and to workarounds in important cases, is available. Further developments should draw on such knowledge, aiming to obtain more principled, and perhaps even better performing, heuristics. Likewise, we believe that improved upper bounds on network capacity should also be obtainable. A promising way to this end seems to be to work directly on a graph whose edges (not vertices) stand for links. Doing this will bring in a considerable number of results on the multicoloring of edges, as described for example in Section 28.5 of [39], and constitutes the subject of current research.

We end with a note on the eventual practical use of our approach to perform actual link scheduling on real networks. Like so many of the approaches that precede ours, any single-color heuristic conforming to the template given by Steps S1–S4 of Section 2 is essentially a centralized procedure, that is, one not naturally suited to direct deployment for operation on a real network. Instead, such heuristics aim to demonstrate strategies to better exploit the interference patterns produced by the links in order to maximize capacity. This is therefore true of both GreedyPhysical and ApproxLogN, and of MaxCRank as well, holding moreover for their multicoloring-based generalizations through Steps M1–M6 of Section 3. Turning such heuristics into distributed algorithms for the actual scheduling of links requires algorithm design that differs substantially from what we have undertaken in this work. As such, it constitutes the subject of further research.

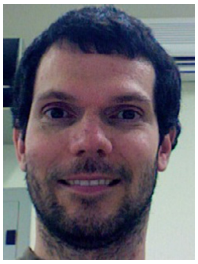
Acknowledgments

We acknowledge partial support from CNPq, CAPES, a FAPERJ BBP grant, and a joint PAPD grant from FAPERJ and CAPES. Several of our computational experiments were carried out on the Grid'5000 experimental testbed, which is being developed under the INRIA ALADDIN development action with support from CNRS, RENATER, and several universities as well as other funding bodies (see `https://www.grid5000.fr`).

References

[1] P. Gupta, P.R. Kumar, The capacity of wireless networks, IEEE Trans. Inf. Theory 46 (2000) 388–404.
[2] K. Jain, J. Padhye, V.N. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, in: Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom), 2003, pp. 66–80.
[3] G. Brar, D.M. Blough, P. Santi, Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks, in: Proceedings of the Twelfth Annual International Conference on Mobile Computing and Networking (MobiCom), 2006, pp. 2–13.
[4] D.M. Blough, G. Resta, P. Santi, Approximation algorithms for wireless link scheduling with SINR-based interference, IEEE/ACM Trans. Netw. 18 (2010) 1701–1712.

[5] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, O. Frieder, Efficient interference-aware TDMA link scheduling for static wireless networks, in: Proceedings of the Twelfth Annual International Conference on Mobile Computing and Networking (MobiCom), 2006, pp. 262–273.
[6] T. Moscibroda, Y.A. Oswald, R. Wattenhofer, How optimal are wireless scheduling protocols? in: Proceedings of the Twenty-sixth IEEE International Conference on Computer Communications (INFOCOM), 2007, pp. 1433–1441.
[7] D. Chafekar, V.S.A. Kumart, M.V. Marathe, S. Parthasarathy, A. Srinivasan, Approximation algorithms for computing capacity of wireless networks with SINR constraints, in: Proceedings of the 2008 IEEE International Conference on Computer Communications (INFOCOM), 2008, pp. 1166–1174.
[8] Q.-S. Hua, F.C.M. Lau, Exact and approximate link scheduling algorithms under the physical interference model, in: Proceedings of the Fifth International Workshop on Foundations of Mobile Computing (DIALM-FOMC), 2008, pp. 45–54.
[9] O. Goussevskaia, R. Wattenhofer, M.M. Halldorsson, E. Welzl, Capacity of arbitrary wireless networks, in: Proceedings of the 2009 IEEE International Conference on Computer Communications (INFOCOM), 2009, pp. 1872–1880.
[10] P. Santi, R. Maheshwari, G. Resta, S. Das, D.M. Blough, Wireless link scheduling under a graded SINR interference model, in: Proceedings of the Second ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC), 2009, pp. 3–12.
[11] D. Yang, X. Fang, N. Li, G. Xue, A simple greedy algorithm for link scheduling with the physical interference model, in: Proceedings of the Twenty-eighth IEEE Conference on Global Telecommunications (GLOBECOM), 2009, pp. 4572–4577.
[12] C. Boyaci, B. Li, Y. Xia, An investigation on the nature of wireless scheduling, in: Proceedings of the 2009 IEEE International Conference on Computer Communications (INFOCOM), 2010, pp. 1–9.
[13] D. Yang, X. Fang, G. Xue, A. Irani, S. Misra, Simple and effective scheduling in wireless networks under the physical interference model, in: Proceedings of the 2010 IEEE Conference on Global Telecommunications (GLOBECOM), 2010, pp. 1–5.
[14] C.H.P. Augusto, C.B. Carvalho, M.W.R.da. Silva, J.F.de Rezende, REUSE: A combined routing and link scheduling mechanism for wireless mesh networks, Comput. Commun. 34 (2011) 2207–2216.
[15] M. Leconte, J. Ni, R. Srikant, Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks, IEEE/ACM Trans. Netw. 19 (2011) 709–720.
[16] Y. Shi, Y.T. Hou, S. Kompella, H.D. Sherali, Maximizing capacity in multihop cognitive radio networks under the SINR model, IEEE Trans. Mob. Comput. 10 (2011) 954–967.
[17] O. Goussevskaia, M.M. Halldorsson, R. Wattenhofer, Algorithms for wireless capacity, IEEE/ACM Trans. Netw. 22 (2014) 745–755.
[18] M. Nabli, F. Abdelkefi, W. Ajib, M. Siala, Efficient centralized link scheduling algorithms in wireless mesh networks, in: Proceedings of the 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), 2014, pp. 660–665.
[19] C. Wang, J. Yu, D. Yu, B. Huang, S. Yu, An improved approximation algorithm for the shortest link scheduling in wireless networks under SINR and hypergraph models, J. Comb. Optim. (in press).
[20] Z.-A. Zhou, C.-G. Li, Approximation algorithms for maximum link scheduling under SINR-based interference model, Int. J. Distrib. Sens. Netw. 2015 (2015). Article ID: 120812.
[21] R.L. Cruz, A.V. Santhanam, Optimal routing, link scheduling and power control in multihop wireless networks, in: Proceedings of the Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2003, pp. 702–711.
[22] M. Alicherry, R. Bhatia, L. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, in: Proceedings of the Eleventh Annual International Conference on Mobile Computing and Networking (MobiCom), 2005, pp. 58–72.
[23] J. Wang, P. Du, W. Jia, L. Huang, H. Li, Joint bandwidth allocation, element assignment and scheduling for wireless mesh networks with MIMO links, Comput. Commun. 31 (2008) 1372–1384.
[24] A. Capone, G. Carello, I. Filippini, S. Gualandi, F. Malucelli, Routing, scheduling and channel assignment in wireless mesh networks: optimization models and algorithms, Ad Hoc Netw. 8 (2010) 545–563.
[25] A.D. Gore, A. Karandikar, Link scheduling algorithms for wireless mesh networks, IEEE Comm. Surv. Tutor. 13 (2011) 258–273.
[26] T. Kesselheim, A constant-factor approximation for wireless capacity maximization with power control in the SINR model, in: Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2011, pp. 1549–1559.
[27] P.-J. Wan, O. Frieder, X. Jia, F. Yao, X. Xu, S. Tang, Wireless link scheduling under physical interference model, in: Proceedings of the 2011 IEEE International Conference on Computer Communications (INFOCOM), 2011, pp. 838–845.
[28] I. Rubin, C.-C. Tan, R. Cohen, Joint scheduling and power control for multicasting in cellular wireless networks, EURASIP J. Wirel. Commun. 2012 (2012) 250.
[29] F.R.J. Vieira, J.F.de Rezende, V.C. Barbosa, S. Fdida, Scheduling links for heavy traffic on interfering routes in wireless mesh networks, Comput. Netw. 56 (2012) 1584–1598.
[30] C. Berge, Multicoloring of a graph, Math. Program. Stud. 8 (1978) 226–234.
[31] R. Mathar, T. Niessen, Optimum positioning of base stations for cellular radio networks, Wirel. Netw. 6 (2000) 421–428.
[32] C. Prommak, J. Kabara, D. Tipper, C. Charnsripinyo, Next generation wireless LAN system design, in: Proceedings of the 2002 International Conference for Military Communications (MILCOM), vol. 1, 2002, pp. 473–477.
[33] P. Wertz, M. Sauter, F.M. Landstorfer, G. Wölfle, R. Hoppe, Automatic optimization algorithms for the planning of wireless local area networks, in: Proceedings of the Sixtieth IEEE Vehicular Technology Conference (VTC), 2004, pp. 3010–3014.
[34] A. Bahri, S. Chamberland, On the wireless local area network design problem with performance guarantees, Comput. Netw. 48 (2005) 856–866.
[35] L. Narayanan, Channel assignment and graph multicoloring, Handbook of Wireless Networks and Mobile Computing, John Wiley & Sons, New York, NY, 2002, pp. 71–94.
[36] K.I. Aardal, S.P.M.van Hoesel, A.M.C.A. Koster, C. Mannino, A. Sassano, Models and solution techniques for frequency assignment problems, Ann. Oper. Res. 153 (2007) 79–129.
[37] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, third edition, The MIT Press, Cambridge, MA, 2009.
[38] F.H. Clarke, R.E. Jamison, Multicolorings, measures and games on graphs, Discret. Math. 14 (1976) 241–245.
[39] A. Schrijver, Combinatorial Optimization, Springer-Verlag, Berlin, Germany, 2003.



Fabio R. J. Vieira received a Ph.D. in Computer Science from Universidade Federal do Rio de Janeiro (UFRJ) and LIP6 at Université Pierre et Marie Curie (UPMC). He received his B.Sc. degree in Electronics Engineering from Universidade do Estado do Rio de Janeiro and his M.Sc. in Computer Science from UFRJ. From 2012 to 2015 he was a post-doctoral associate at UFRJ and a visiting researcher at UPMC. His research interests focus on graph models for wireless and biological networks, and on artificial intelligence methods and distributed algorithms for those models.

José F. de Rezende received the B.Sc. and M.Sc. degrees in Electronics Engineering from Universidade Federal do Rio de Janeiro (UFRJ) in 1988 and 1991, respectively. He received the Ph.D. degree in Computer Science from Université Pierre et Marie Curie in 1997, where he was an associate researcher during that year. Since 1998 he has been an associate professor at UFRJ. His research interests are in distributed multimedia applications, multipeer communication, high speed and mobile networks, and quality of service in the Internet.

Valmir C. Barbosa received a Ph.D. in Computer Science from the University of California, Los Angeles, in 1986 and is a professor at Universidade Federal do Rio de Janeiro. His research interests are in the locality and complexity aspects of natural and computational distributed systems. He is the author of An Introduction to Distributed Algorithms (The MIT Press, 1996) and two other books. He has served in the editorial boards of Parallel Computing, the IEEE Transactions on Computers, and the Journal of the Brazilian Computer Society.