



HAL
open science

Maximum Flow and Minimum-Cost Flow in Multi-Interface Networks

Gianlorenzo d'Angelo, Gabriele Di Stefano, Alfredo Navarra

► **To cite this version:**

Gianlorenzo d'Angelo, Gabriele Di Stefano, Alfredo Navarra. Maximum Flow and Minimum-Cost Flow in Multi-Interface Networks. 5th International Conference on Ubiquitous Information Management and Communication, Feb 2011, Seoul, South Korea. pp.19, 10.1145/1968613.1968637 . hal-00644073

HAL Id: hal-00644073

<https://inria.hal.science/hal-00644073v1>

Submitted on 23 Nov 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Maximum Flow and Minimum-Cost Flow in Multi-Interface Networks

Gianlorenzo D'Angelo,
Gabriele Di Stefano
Dept. of Electrical & Information Engineering
University of L'Aquila 67100 L'Aquila, Italy
{gianlorenzo.dangelo,
gabriele.distefano}@univaq.it

Alfredo Navarra
Dept. of Computer Science & Mathematics
University of Perugia 06123 Perugia, Italy
navarra@dmi.unipg.it

ABSTRACT

In heterogeneous networks, devices can communicate by means of multiple wired or wireless interfaces. By switching among interfaces or by combining the available interfaces, each device might establish several connections. A connection is established when the devices at its endpoints share at least one active interface. Each interface is assumed to require an activation cost, and provides a communication bandwidth. In this paper, we consider two fundamental optimization problems. In the first one, we aim to activate a set of interfaces in the network $G = (V, E)$ in order to guarantee the maximal bandwidth between two given nodes. Nodes V represent the devices, edges E represent the connections that can be established according to the availability of the interfaces in the devices. In the second problem, we look for activating the cheapest set of interfaces among a network in order to guarantee a minimum bandwidth B of communication between two specified nodes. We show that the first problem is polynomially solvable while the second one is *NP*-Hard. However, we experimentally analyzed an algorithm for the second problem, showing that in practical cases it guarantees a low approximation ratio which allows us to use it in real-world networks.

Categories and Subject Descriptors

C.2 [Computer-communication Networks]; C.2.1 [Network Architecture and Design]; [Network communications]; F.2 [Analysis of Algorithms and Problem Complexity]

General Terms

Algorithms, Experimentation

Keywords

Multi-Interface Networks, Flow, Approximation factor, Lower and Upper bounds

1. INTRODUCTION

The interest in wireless networks has rapidly grown during the last decades. Their success is certainly due to the wide range of applications for which such networks are designed. A very important issue is constituted by the heterogeneity of the devices which might interact in order to exchange data. Wireless networks are, in fact, composed of devices with different characteristics like computational power, energy consumption, radio interfaces, supported communication protocols, and so forth. In this paper, we are mainly interested in devices equipped with multiple interfaces (like Bluetooth, WiFi, GPRS, etc.). A connection between two or more devices might be accomplished by means of different communication networks according to connectivity and quality of service requirements. The selection of the most suitable interface for a specific connection might depend on various factors. Such factors include: its availability in specific devices, the required communication bandwidth, the cost (in terms of energy consumption) of maintaining an active interface, the available neighbors, and so forth. While managing such connections, a lot of effort must be devoted to energy consumption issues. Devices are, in fact, usually battery powered and the network survivability might depend on their persistence in the network.

We study communication problems in wireless networks supporting multiple interfaces. In the considered model, the input network is described by a graph $G = (V, E)$, where V represents the set of wireless devices and E is the set of possible connections induced by the proximity of devices and the available interfaces that they may share. Each node $v \in V$ and each edge $e \in E$ is associated with a set of available interfaces $W(v)$ and $X(e)$, respectively. The set of all the possible interfaces available in the network is then determined by $\bigcup_{v \in V} W(v)$; we denote the cardinality of this set by k . We say that a connection is satisfied (or covered) when the endpoints of the corresponding edge share at least one active interface. If an interface x is activated at some node u , then u consumes some energy $c(x)$ for maintaining x as active, and it provides a maximum communication bandwidth $b(x)$ with all its neighbors which share interface x . In this setting, we study two optimization problems whose aim is to establish a connection between two selected nodes $s, t \in V$, taking into account the bandwidth constraints. First, we study the problem of finding a communication sub-network between two selected nodes $s, t \in V$ with the maximal bandwidth possible. In detail, we look for a set of active interfaces in the input network in such a way that the bandwidth

guarantee between s and t is maximal. Then, we study the problem of establishing a communication sub-network between two selected nodes $s, t \in V$ of minimum cost in terms of energy consumption, while guaranteeing a minimum communication bandwidth B . In other words, we look for the minimum cost set of active interfaces among the input network, in such a way that s is guaranteed to exchange data with t at least with some bandwidth B . In both cases, between s and t not necessarily a path of covered edges must be established but a more complex graph might be required according to the topology and to the available interfaces.

Related work. Multi-interface wireless networks have recently been studied in a variety of contexts, usually focusing on the benefits of multiple radio devices of each node. Many basic problems of standard wireless network optimization can be reconsidered in such a setting [6], in particular, focusing on issues related to routing [12] and network connectivity [8, 13]. The study of combinatorial problems on multi-interface wireless networks has originated from [7]. That paper, as well as [15] investigate the so called *Coverage* problem, where the goal is the activation of the minimum cost set of interfaces in such a way that all the edges of G are covered. *Connectivity* issues have been addressed in [5, 9, 16]. The goal becomes to activate the minimum cost set of interfaces in G in order to guarantee a path of communication between every pair of nodes. In [16], the attention has been devoted to the so called *Cheapest path* problem. This corresponds to the well-known shortest path problem but in the context of multi-interface networks. A natural continuation on investigating such kind of networks is certainly to consider also quality of service constraints in the problem. To the best of our knowledge the only work which cope with bandwidth issues is [10] where only theoretical results are given.

Our results. In this paper, we are interested in two fundamental optimization problems which take into account bandwidth constraints in the network.

The first problem, called *Maximum Flow in Multi-Interface Networks (MFMI)*, aims to find a sub-network of a given input network where the communication bandwidth between two given nodes is maximal. Such problem is similar to the fundamental problem of finding the maximum flow between two nodes in a network. The main difference resides in the fact that, in *MFMI*, the bandwidth capacities are associated to the interfaces instead of edges. Therefore, if a node v uses (part of) the bandwidth of an interface i to communicate with a neighbor u , it cannot use i to communicate with another neighbor w , even if i belongs to both v and w . We show that this problem is optimally solvable in polynomial time by giving an algorithm to solve it.

The second problem aims to establish the cheapest way of communication between two given nodes while guaranteeing a minimum bandwidth of communication. Such problem, called *Minimum-Cost Flow in Multi-Interface Networks (MCFMI)* is similar to the better known *Minimum Edge Cost Flow* [14]. Again, we do not consider costs and capacities for the edges of the network but we have to cope with interfaces at the nodes that require some costs and can manage some maximum bandwidths. In the special case where each connection can be established by means of a different interface, the two problems coincide. Hence, it is not surprising that *MCFMI* turns out to be *NP*-hard when the number k of interfaces is unbounded [10]. In [10] the author also

give an approximation algorithm with ratio guarantee b_{max} , where b_{max} is the maximum communication bandwidth allowed among all the available interfaces. This algorithm optimally solves the problem in the case that the bandwidth is constant for all the interfaces. In this paper, we analyze the practical performances of such algorithm. In detail, we implemented it and tested over a large set of networks. The network topologies, the interface bandwidths, and the interface costs are generated by using two different models which simulate real wireless networks. We show that although the theoretical approximation ratio is high, it is very low in the practical cases analyzed, as a consequence, the algorithm can be effectively used in real-world networks.

2. PROBLEM STATEMENTS AND THEORETICAL RESULTS

Given a network, we denote by V the set of nodes. A global assignment of the interfaces to the nodes in V is given in terms of an appropriate interface assignment function $W: V \rightarrow 2^{\{1,2,\dots,k\}}$. Further, for each pair of nodes in V , the function $X: V \times V \rightarrow 2^{\{1,2,\dots,k\}}$ denotes the set of interfaces that the two nodes can use to communicate. The properties of the functions W and X are the following:

- for u in V , $X(u, u) = \emptyset$;
- for u, v in V , $X(u, v) = X(v, u)$;
- for u, v in V , $X(u, v) \subseteq W(u) \cap W(v)$.

Note that, the above definitions of V , W and X induce a graph $G = (V, E)$ where $\{u, v\} \in E$ if and only if $X(u, v) \neq \emptyset$. We say that the function W covers the graph G and that the graph G is induced by the function X .

Unless otherwise stated, the graph $G = (V, E)$ representing the network is assumed to be undirected and connected. In the remainder, for a graph G , we denote by Δ its maximum node degree.

The cost of activating an interface i is given by the cost function $c: \{1, 2, \dots, k\} \rightarrow \mathbb{Z}_0^+$ and it is denoted as $c(i)$. The bandwidth allowed by a given interface i is defined by the bandwidth function $b: \{1, 2, \dots, k\} \rightarrow \mathbb{Z}_0^+$ and it is denoted as $b(i)$. It follows that each node holding an interface i pays the same cost $c(i)$ and provides the same bandwidth $b(i)$ by activating i .

The considered *MFMI* and *MCFMI* optimization problems are formulated as follows.

Input: A set of nodes V , a source node $s \in V$, a target node $t \in V$, a set of interfaces $I = \{1, 2, \dots, k\}$, an allocation of available interfaces $W: V \rightarrow 2^I$ covering the graph $G = (V, E)$ induced by a function $X: V \times V \rightarrow 2^I$, and an interface bandwidth function $b: I \rightarrow \mathbb{Z}_0^+$.

Sol.: An allocation of active interfaces $W_A: V \rightarrow 2^I$, $W_A(v) \subseteq W(v)$ for all $v \in V$ and a flow function $f: V \times V \times I \rightarrow \mathbb{Z}_0^+$ such that:

- $f(u, v, i) = 0$ if $W_A(u) \cap W_A(v) \cap X(u, v) = \emptyset$ for all $u, v \in V$ and $i \in I$;
- $\sum_{v \in V: f(u, v, i) > 0} f(u, v, i) \leq b(i)$ for all $u \in V$ and $i \in I$;
- $f(u, v, i) = -f(v, u, i)$ for all $u, v \in V$ and $i \in I$;
- $\sum_{v \in V, i \in I} f(u, v, i) = 0$ for all $u \in V \setminus \{s, t\}$;

Goal: Maximize the total flow from s to t , $F = \sum_{v \in V, i \in I} f(s, v, i) = \sum_{v \in V, i \in I} f(v, t, i)$.

Input: A set of nodes V , a source node $s \in V$, a target node $t \in V$, a set of interfaces $I = \{1, 2, \dots, k\}$, an allocation of available interfaces $W: V \rightarrow 2^I$ covering the graph $G = (V, E)$ induced by a function $X: V \times V \rightarrow 2^I$, an interface cost function $c: I \rightarrow \mathbb{Z}_0^+$, an interface bandwidth function $b: I \rightarrow \mathbb{Z}_0^+$ and a bound $B \in \mathbb{Z}_0^+$.

Sol.: An allocation of active interfaces $W_A: V \rightarrow 2^I$, $W_A(v) \subseteq W(v)$ for all $v \in V$ and a flow function $f: V \times V \times I \rightarrow \mathbb{Z}_0^+$ such that:

- $f(u, v, i) = 0$ if $W_A(u) \cap W_A(v) \cap X(u, v) = \emptyset$ for all $u, v \in V$ and $i \in I$;
- $\sum_{v \in V: f(u, v, i) > 0} f(u, v, i) \leq b(i)$ for all $u \in V$ and $i \in I$;
- $f(u, v, i) = -f(v, u, i)$ for all $u, v \in V$ and $i \in I$;
- $\sum_{v \in V, i \in I} f(u, v, i) = 0$ for all $u \in V \setminus \{s, t\}$;
- $\sum_{v \in V, i \in I} f(s, v, i) = \sum_{v \in V, i \in I} f(v, t, i) \geq B$;

Goal: Minimize the total cost of the active interfaces, $c(W_A) = \sum_{v \in V} \sum_{i \in W_A(v)} c(i)$.

In the following we give a polynomial time algorithm to solve the *MFMI* problem, then we report a recent result about *MCFMI* which shows that the problem is *NP-hard* and hence, it cannot be solved in polynomial time. Finally, we report an approximation algorithm with a theoretical approximation guarantee for *MCFMI*.

Both the algorithms for *MFMI* and *MCFMI* are based on a transformation of the graph $G = (V, E)$ induced by the network and the set I of interfaces, into a directed graph $G' = (V', A)$. G' defined in such a way that bandwidths and costs are associated to arcs rather than to interfaces. The transformation is defined as follows.

2.1 Graph transformation

Informally, for each interface of each node, there is an arc which has the same cost and bandwidth of the considered interface. The head of each of such arcs is connected to the tail of another arc of the same kind if they share an interface or they represent different interfaces of the same node. Formally, there are two nodes in V' for each node in V and for each interface of each node:

$$V' = \{(\bar{v}, i), (\underline{v}, i) \mid v \in V, i \in W(v)\} \cup \{\bar{s}, \bar{t}\},$$

The arcs are the following:

$$A = \{((\bar{v}, i), (\underline{v}, i)) \mid v \in V, i \in W(v)\} \cup$$

$$\{((\underline{v}, i), (\bar{v}, j)) \mid v \in V, i, j \in W(v) \text{ s.t. } i \neq j\} \cup$$

$$\{((\underline{u}, i), (\bar{v}, i)) \mid i \in X(u, v)\} \cup$$

$$\{(\bar{s}, (\bar{s}, i)), ((\bar{t}, j), \bar{t}) \mid i \in W(s), j \in W(t)\}.$$

The capacity of each arc $a = ((\bar{v}, i), (\underline{v}, i))$ is set to $b'((\bar{v}, i), (\underline{v}, i)) = b(i)$ whereas the capacity of each other arc in A is unlimited and it is 0 for each pair in $V \times V \setminus A$. The cost $c'(a)$ of each arc $((\bar{v}, i), (\underline{v}, i))$ is set to $c(i)$ and it is 0 for the remaining arcs.

Given an flow function f' from \bar{s} to \bar{t} for G' , we define a flow function f from s to t in G as follows:

$$f(u, v, i) = \begin{cases} f'((\underline{u}, i), (\bar{v}, i)) - f'((\underline{v}, i), (\bar{u}, i)) & \text{if } i \in X(u, v) \\ 0 & \text{otherwise.} \end{cases}$$

The allocation of active interfaces at node u , is defined as $W_A(u) = \{i \in W(u) \mid \exists v \in V \text{ s.t. } f(u, v, i) \neq 0\}$. Note that both functions f and W_A can be computed in polynomial time once function f' is known.

2.2 Polynomial time algorithm for Maximum Flow in Multi-Interface Networks

We denote by ALG_1 the algorithm defined in this section to solve *MFMI*.

Given an instance I_1 of *MFMI*, ALG_1 first transforms the graph G and the function b of I_1 into a graph G' and a function b' as described above, obtaining an instance I_2 of the classical maximum flow problem. Then, in polynomial time it finds an optimal flow function f' for I_2 by using a maximum flow algorithm. Finally, ALG_1 transform f' into an optimal flow function f for I_1 and define an allocation of active interfaces W_A , again by using the transformation given above. The following theorem shows that f is optimal for I_1 .

THEOREM 2.1. *ALG₁ finds optimal solutions for MFMI.*

PROOF. We recall that, by definition of maximal flow function:

$$0 \leq f'(x, y) \leq b'(x, y), \quad \text{for each } (x, y) \in A \quad (1)$$

$$f'(x, y) = -f'(y, x), \quad \text{for each } x, y \in V' \quad (2)$$

$$\sum_{x \in V'} f'(x, y) = 0, \quad \text{for each } y \in V' \setminus \{\bar{s}, \bar{t}\} \quad (3)$$

$$\sum_{y \in V'} f'(\bar{s}, y) \quad \text{is maximal.} \quad (4)$$

We show that function f fulfills the properties stated in the definition of *MFMI* and that it is maximal.

- $f(u, v, i) = -f(v, u, i)$ for all $u, v \in V$ and $i \in I$. If $i \notin X(u, v)$, then $f(u, v, i) = f(v, u, i) = 0$. Otherwise, by definition of f ,

$$f(u, v, i) = f'((\underline{u}, i), (\bar{v}, i)) - f'((\underline{v}, i), (\bar{u}, i)) =$$

$$- (f'((\underline{v}, i), (\bar{u}, i)) - f'((\underline{u}, i), (\bar{v}, i))) = -f(v, u, i).$$

- $f(u, v, i) = 0$ if $W_A(u) \cap W_A(v) \cap X(u, v) = \emptyset$ for all $u, v \in V$ and $i \in I$. If $W_A(u) \cap W_A(v) \cap X(u, v) = \emptyset$, then for each $i \in I$ either $i \notin W_A(u)$ or $i \notin W_A(v)$, or $i \notin X(u, v)$. If $i \notin W_A(u)$, then by definition of $W_A(u)$ $f(u, v, i) = 0$. If $i \notin W_A(v)$, then by definition of $W_A(v)$, $f(v, u, i) = 0$, moreover by the above property, $f(u, v, i) = -f(v, u, i) = 0$. If $i \notin X(u, v)$, then by definition of f , $f(u, v, i) = 0$.

- $\sum_{v \in V: f(u, v, i) > 0} f(u, v, i) \leq b(i)$ for all $u \in V$ and $i \in I$. By the definition of f ,

$$\sum_{v \in V: f(u, v, i) > 0} f(u, v, i) =$$

$$\sum_{v \in V: f(u, v, i) > 0} (f'((\underline{u}, i), (\bar{v}, i)) - f'((\underline{v}, i), (\bar{u}, i))).$$

By Property (1), for each $v \in V$, $f'((\underline{u}, i), (\bar{u}, i)) \geq 0$ and $f'((\underline{u}, i), (\bar{v}, i)) \geq 0$, then

$$\sum_{v \in V: f(u, v, i) > 0} (f'((\underline{u}, i), (\bar{v}, i)) - f'((\underline{v}, i), (\bar{u}, i))) \leq$$

$$\sum_{v \in V: f(u, v, i) > 0} f'((\underline{u}, i), (\bar{v}, i)) \leq$$

by the definition of b' ,

$$\sum_{v \in V} f'((\underline{u}, i), (\bar{v}, i)).$$

By Property (3), applied to node (\underline{u}, i) ,

$$\sum_{v \in V} f'((\underline{u}, i), (\bar{v}, i)) =$$

$$f'((\bar{u}, i), (\underline{u}, i)) - \sum_{j \in I \setminus \{i\}} f'((\underline{u}, i), (\bar{u}, j)).$$

Again by Property (1), $f'((\underline{u}, i), (\bar{u}, j)) \geq 0$, for each $j \in I \setminus \{i\}$, then

$$f'((\bar{u}, i), (\underline{u}, i)) - \sum_{j \in I \setminus \{i\}} f'((\underline{u}, i), (\bar{u}, j)) \leq$$

$$f'((\bar{u}, i), (\underline{u}, i)) \leq b(i).$$

The last inequality directly follows from Property (1).

- $\sum_{v \in V, i \in I} f(u, v, i) = 0$ for all $u \in V \setminus \{s, t\}$. By definition of f , $f(u, v, i) = 0$ if $v = u$ or $i \notin X(u, v)$, hence

$$\sum_{v \in V, i \in I} f(u, v, i) =$$

$$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u, v)}} (f'((\underline{u}, i), (\bar{v}, i)) - f'((\underline{v}, i), (\bar{u}, i))).$$

By Property (3), applied to nodes (\underline{u}, i) ,

$$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u, v)}} f'((\underline{u}, i), (\bar{v}, i)) =$$

$$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u, v)}} (f'((\bar{u}, i), (\underline{u}, i)) - \sum_{j \in X(u, v) \setminus \{i\}} f'((\underline{u}, i), (\bar{u}, j))).$$

Again, by Property (3), applied to nodes (\bar{u}, i) ,

$$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u, v)}} f'((\underline{v}, i), (\bar{u}, i)) =$$

$$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u, v)}} (f'((\bar{u}, i), (\underline{u}, i)) - \sum_{j \in X(u, v) \setminus \{i\}} f'((\underline{u}, j), (\bar{u}, i))).$$

Hence,

$$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u, v)}} (f'((\underline{u}, i), (\bar{v}, i)) - f'((\underline{v}, i), (\bar{u}, i))) =$$

$$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u, v)}} \left(\sum_{j \in X(u, v) \setminus \{i\}} f'((\underline{u}, j), (\bar{u}, i)) - \sum_{j \in X(u, v) \setminus \{i\}} f'((\underline{u}, i), (\bar{u}, j)) \right) = 0,$$

in fact, for any pair of interfaces p and q such that $p \neq q$, we have that, when $i = p$ and $j = q$, the related term of the above sum is

$$f'((\underline{u}, q), (\bar{u}, p)) - f'((\underline{u}, p), (\bar{u}, q))$$

contrarily, when $i = q$ and $j = p$, it is

$$f'((\underline{u}, p), (\bar{u}, q)) - f'((\underline{u}, q), (\bar{u}, p))$$

and hence the overall sum is 0.

We show that f is maximal by contradiction. Let us suppose that there exists a flow function $f'' : V \times V \times I \rightarrow \mathbb{Z}_0^+$ for I_1 such that $\sum_{v \in V, i \in I} f''(s, v, i) > \sum_{v \in V, i \in I} f(s, v, i)$. We define a flow function $f''' : V \times V \rightarrow \mathbb{Z}_0^+$ for I_2 as follows, if $i \in X(u, v)$,

$$f'''((\underline{u}, i), (\bar{v}, i)) = \begin{cases} f(u, v, i) & \text{if } f(u, v, i) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

In edges in $\{(\bar{v}, i), (\underline{v}, i) \mid v \in V, i \in W(v)\} \cup \{(\underline{v}, i), (\bar{v}, j) \mid v \in V, i, j \in W(v) \text{ s.t. } i \neq j\}$, f''' is defined in order to satisfy flow conservation constraints, and it is 0 for any other pairs in $V \times V$.

Similar arguments as above can be used to show that f''' fulfills the properties of flow functions and that

$$\sum_{v \in V'} f'''(\bar{s}, v) = \sum_{v \in V, i \in I} f''(s, v, i)$$

$$\sum_{v \in V, i \in I} f(s, v, i) = \sum_{v \in V'} f'(\bar{s}, v).$$

It follows that

$$\sum_{v \in V'} f'''(\bar{s}, v) = \sum_{v \in V, i \in I} f''(s, v, i) >$$

$$\sum_{v \in V, i \in I} f(s, v, i) = \sum_{v \in V'} f'(\tilde{s}, v),$$

a contradiction. \square

2.3 Complexity and approximation algorithm for Minimum-Cost Flow in Multi-Interface Networks

In this section we report a recent result about *MCFMI* which shows that the problem is *NP*-hard and hence, it cannot be solved in polynomial time. Therefore, we focus on approximation algorithms by giving an algorithm (taken from [10]) with a theoretical approximation guarantee for *MCFMI*. In the next section, we experimentally analyze such algorithm.

THEOREM 2.2. [10] *MCFMI is NP-hard even when restricted to the unit cost interface case for any fixed $\Delta \geq 3$ and $k \geq 2$.*

Theorem 2.2 also holds when the number of interfaces is unbounded. We now describe the b_{\max} -approximation algorithm for *MCFMI* given in [10], where b_{\max} is the maximum bandwidth value among the interfaces in I . We denote this algorithm by ALG_2 . It consists in relaxing *MCFMI* to the well-known *Integral Minimum-Cost Flow (IMCF)* problem [2]. That is, we transform an instance of *MCFMI* into an instance of *IMCF*. In [10], it has been shown that such a transformation guarantees an approximation factor of b_{\max} . Let \mathcal{A} be an algorithm which optimally solves *IMCF* in a graph $H = (V_H, E_H)$ in polynomial time $P_{\mathcal{A}}(|V_H| + |E_H|)$.

Given an instance I_1 of *MCFMI*, ALG_2 works in four phases. First, it transforms an instance I_1 on a graph $G = (V, E)$ of *MCFMI* into an instance I_2 of an equivalent problem defined on a directed graph $G' = (V', A)$ as described in Section 2.1. Then, it transforms I_2 into an instance I_3 of *IMCF*. In the third phase, ALG_2 solves I_3 by using a known algorithm and, finally, it transforms the obtained solution for I_3 into a solution for I_2 .

Given a solution for I_2 , which defines a flow function f_2 , we can define a solution for I_1 by assigning a flow function f_1 as described in Section 2.1. That is, for each $v, u \in V$, $f_1(v, u, i) = f_2(\underline{v}, i, \bar{u}, i) - f_2(\underline{u}, i, \bar{v}, i)$ if $i \in X(u, v)$, and $f_1(v, u, i) = 0$ otherwise. Vice versa, given a solution for I_1 , which defines a flow function f'_1 , we can define a solution for I_2 by assigning a flow function f'_2 such that $f'_2(\underline{v}, i, \bar{u}, i) = f'_1(v, u, i)$, if $f'_1(v, u, i) > 0$ and $f'_2(\underline{v}, i, \bar{u}, i) = 0$ otherwise, for each $v, u \in V$ and $i \in X(u, v)$. The flows in the remainder of A are set in order to satisfy flow conservation constraints. It is not difficult to note that the feasibility of f_2 (f'_1 , resp.) implies the feasibility of f_1 (f'_2). Moreover, the cost of f_2 (f'_1 , resp.) is equal to the cost of f_1 (f'_2) as the cost of arcs $((\bar{v}, i), (\underline{v}, i))$ in A is $c(i)$ and it is 0 for any other arc. By the above discussion it follows that we can solve I_1 by solving I_2 .

We find an approximate solution for I_2 by using an *IMCF* instance. The *IMCF* problem consists of finding an integral flow greater than or equal to a given quantity between two nodes in a directed graph H where each arc a has a capacity $\beta(a)$ and cost $\chi(a)$. The objective is to minimize the function $\sum_{a \in A^+} \chi(a) \cdot f(a)$, where $f(a)$ is the flow on arc a and A^+ is the set of arcs with positive flow. This problem admits a polynomial time algorithm (see, e.g., [17]).

We obtain an *IMCF* instance I_3 from I_2 by setting $H = G'$, and for each $a = (p, q) \in A$, $\beta(a) = b'(p, q)$, and $\chi(a) = c'(a)/b'(p, q)$.

The following theorem shows that solving *IMCF* for I_3 provides a b_{\max} -approximation for I_2 and therefore ALG_2 provides a b_{\max} approximation guarantee and that it requires polynomial time.

THEOREM 2.3. [10] *ALG₂ is a b_{\max} -approximation algorithm for MCFMI and requires $O(|V|k^2 + |E| + P_{\mathcal{A}}(|V|k^2 + |E|))$ time.*

PROOF. Here, we report only a sketch of the proof whose full version can be found in [10].

Let us denote as f^* and f^{IMCF} two optimal flow functions for I_2 and I_3 , respectively and as A^* and A^{IMCF} the corresponding sets of arcs with positive flow. By definition, $\text{OPT} = \sum_{a=(p,q) \in A^*} c'(a)$. As $f^*(a) \leq b'(p, q)$, it follows that

$$\sum_{a=(p,q) \in A^*} c'(a) \geq \sum_{a=(p,q) \in A^*} c'(a) \cdot \frac{f^*(a)}{b'(p, q)} = \sum_{a \in A^*} \chi(a) \cdot f^*(a).$$

By the optimality of A^{IMCF} it follows that

$$\begin{aligned} \sum_{a \in A^*} \chi(a) \cdot f^*(a) &\geq \sum_{a \in A^{IMCF}} \chi(a) \cdot f^{IMCF}(a) \\ &= \sum_{a=(p,q) \in A^{IMCF}} \frac{c'(a)}{b'(p, q)} \cdot f^{IMCF}(a). \end{aligned}$$

As $f^{IMCF}(a) \in \mathbb{Z}_0^+$, for each $a \in A$, then $f^{IMCF}(a) \geq 1$, for each $a \in A^{IMCF}$. Moreover, $b_{\max} \geq b'(p, q)$, for each $a = (p, q) \in A^{IMCF}$.

Therefore,

$$\sum_{a=(p,q) \in A^{IMCF}} \frac{c'(a)}{b'(p, q)} \cdot f^{IMCF}(a) \geq \frac{1}{b_{\max}} \sum_{a \in A^{IMCF}} c'(a).$$

Which concludes the proof. \square

If the bandwidth function is a constant, *MCFMI* can be solved in polynomial time. In fact, if $b = 1$, then ALG_2 optimally solves *MCFMI*. Otherwise, it is enough to solve the problem with required bandwidth of $\bar{B} = \lceil \frac{B}{b} \rceil$ and bandwidth $\bar{b}(i) = 1$, for each interface i . This is stated in the next corollary.

COROLLARY 2.4. [10] *Let $b \in \mathbb{Z}_0^+$. If $b(i) = b$ for each $i \in I$, MCFMI is solvable within $O(|V|k^2 + |E| + P_{\mathcal{A}}(|V|k^2 + |E|))$.*

3. EXPERIMENTAL ANALYSIS

In this Section, we report the results of our experimental study on the approximation algorithm ALG_2 given in the previous section.

The experiments have been carried out on a workstation equipped with a 2.66 GHz processor (Intel Core2 Duo E6700 Box) and 8Gb RAM running Linux 2.6 kernel and Gcc compiler, version 4.3.5.

We implemented algorithm ALG_2 by using the LEMON Graph Library [1] framework. In order to solve the *IMCF* instances required by ALG_2 we used the Network Simplex algorithm [11] provided by LEMON as it is the most experimentally efficient in general cases.

3.1 Input data and executed tests.

For our experiments we used instances of *MCFMI* generated at random by using two different models, namely the *Balls-into-bins* model [18, 19] and the *Barabási-Albert* power-law model [3].

The balls-into-bins model is used to simulate wireless nodes thrown at random in a two-dimensional space [18]. In this model, each instance of *MCFMI* is made of a graph $G_{BIB} = (V_{BIB}, E_{BIB})$, a set of interfaces $I_{BIB} = \{1, 2, \dots, k\}$ along with cost and bandwidth functions c_{BIB} , and b_{BIB} , and two allocation functions $W_{BIB} : V_{BIB} \rightarrow 2^{I_{BIB}}$ and $X_{BIB} : V_{BIB} \times V_{BIB} \rightarrow 2^{\{1, 2, \dots, k\}}$. First, nodes in V_{BIB} are generated and, to each of them, a uniformly random position in a unit size square is associated. From the “balls-into-bins” theory [19], we know that throwing randomly n points in a unit square, the probability that no nodes are inside a circle of diameter $d = \sqrt{\frac{\gamma \log n}{n}}$ is smaller than $n^{-\frac{\gamma}{4}}$, hence, for $\gamma > 4$ and large n , this probability is very low. Therefore, to generate edges and interfaces we proceed as follows. For each interface $i \in I_{BIB}$, the radius $r_i > 0$ of the circle covered by interface i is generated uniformly at random in $\left[\frac{1}{|V_{BIB}|}, \sqrt{\gamma \frac{\log(|V_{BIB}|)}{|V_{BIB}|}} - \frac{1}{|V_{BIB}|}\right]$. In this way, interfaces cover a circle having an average diameter of $\sqrt{\frac{\gamma \log |V_{BIB}|}{|V_{BIB}|}}$. Then function W_{BIB} is defined by independently assigning the generated interfaces to nodes with probability 0.5. Given two nodes $u, v \in V_{BIB}$, let (x_u, y_u) and (x_v, y_v) be their associated coordinates in the unit square. If $\sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} \leq r_i$, for some $i \in W_{BIB}(u) \cap W_{BIB}(v)$, an edge $\{u, v\}$ is added to E_{BIB} and interface i is added to $X_{BIB}(u, v)$, i.e., $X_{BIB}(u, v) = W_{BIB}(u) \cap W_{BIB}(v)$. In this way, for large values of $|V_{BIB}|$ and $\gamma > 4$, we have a high probability to obtain a connected network. Finally, functions c_{BIB} and b_{BIB} are defined as $c_{BIB}(i) = r_i^\alpha$ and $b_{BIB}(i) = r_i^\beta$, for each $i \in I_{BIB}$ and for suitable tuning parameter α , and β which are fixed to 1.5, and 2, respectively in the experiments. Source and target nodes are chosen as the nodes with the biggest Euclidean distance.

Barabási–Albert networks have been proven to model many real-world networks such as the Internet, the World Wide Web, citation graphs, and some social networks [4]. A Barabási–Albert topology is generated by iteratively adding one node at a time, starting from a given connected graph with at least two nodes. A newly added node is connected to any other existing nodes with a probability that is proportional to the degree that the existing nodes already have. Hence, the more connected a node is, the more likely it is to receive new connections to the new node. This mechanism is known as *preferential attachment* and it has been observed in many real-world networks.

In this model, each generated instance of *MCFMI* is made of a graph $G_{BA} = (V_{BA}, E_{BA})$, a set of interfaces $I_{BA} = \{1, 2, \dots, k\}$ along with cost and bandwidth functions c_{BA} , and b_{BA} , and two allocation functions $W_{BA} : V_{BA} \rightarrow 2^{I_{BA}}$ and $X_{BA} : V_{BA} \times V_{BA} \rightarrow 2^{\{1, 2, \dots, k\}}$. The graph generation algorithm works as follows. We start from a graph made of two nodes connected by an edge and add one node at a time. A new node v is connected to an existing node u with probability

$$p(v, u) = \frac{\deg(u)}{2m},$$

where $\deg(u)$ is the degree of node u before adding v and m is the number of edges that already exist when v is added. Interfaces I_{BA} and related costs and bandwidth functions are generated in a way similar to the balls-into-bins model, that is, for each interface $i \in I_{BA}$, a number r_i is generated uniformly at random in $\left[\frac{1}{|V_{BA}|}, \sqrt{\gamma \frac{\log(|V_{BA}|)}{|V_{BA}|}} - \frac{1}{|V_{BA}|}\right]$, then $c_{BA}(i)$ and $b_{BA}(i)$ are set to $c_{BA}(i) = r_i^\alpha$ and $b_{BA}(i) = r_i^\beta$. Parameters γ , α , and β are set to 5, 1.5, and 2, respectively. For each edge $e \in E_{BA}$, interface $i \in I_{BA}$ is added to $X_{BA}(e)$ with probability 0.5. For each node v , $W_{BA}(v)$ is induced by the interfaces associated in $X_{BA}(v, u)$ for each edge $\{v, u\}$ incident to v . Source and target nodes are chosen at random among the generated nodes.

For each of the defined instances in both the models above, we considered four values of required flow equally distributed between the minimal bandwidth assigned to an interface b_{\min} and the maximum flow possible F_{\max} , computed by the algorithm given in Section 2.2. That is, we required a flow of $b_{\min} + i \cdot \frac{F_{\max} - b_{\min}}{3}$, for $i = 0, 1, 2, 3$. In the remainder, we will not consider the case where the required flow is b_{\min} (i.e. $i = 0$) as in this case we are able to find an optimal solution to *MCFMI* by computing a shortest path connecting source and destination. Moreover, abusing notation, we denote $\frac{F_{\max} - b_{\min}}{3}$ by δ , where the instance considered is clear from the context.

In order to measure the approximation ratio in the above settings, we need to know the optimal value of each *MCFMI* instance. As it is *NP*-hard to compute such value, we measured the ratio between the objective function value computed by our algorithm and a lower bound to the optimal value, obtaining an upper bound to the actual approximation ratio. In detail, we computed two lower bounds to the optimal value and then we use the maximum among them to get a better estimate of the approximation ratio. One lower bound is simply given by the optimal solution of the *IMCF* instance defined in *ALG*₂. Another lower bound to the optimal value is computed by observing that if we relax the bandwidth constraints by increasing the bandwidth of an interface, we decrease the optimal value. Hence, we computed a lower bound to the optimal value as the optimal value of an instance obtained by setting the bandwidth of each interface to the maximum bandwidth assigned to the original instance. Such a value can be polynomially computed by using Corollary 2.4.

Table 1 reports the size of the input data used in the experiments. We perform two kind of experiments: we fix k to 3, 6, and 9 and we let vary the number of nodes in the graphs from 50 to 1000; we fix the number of nodes in a graph to 10, 100, 1000, and 10000 and let vary k from 2 to 16. In each setting, we considered three values of required flow as explained above. Finally, for each of the above test configurations we performed 10 different experiments and, in the next section, we report average values and standard deviations.

3.2 Analysis of experimental results

The results of our experiments are reported in Figures 1–8 and in Table 2. For as better visualization, all the obtained values are normalized to $|V_{BIB}|$ for the experiments referring to graphs G_{BIB} (Figures 1–4), and to $|V_{BA}|$ for the experiments referring to graphs G_{BA} (Figures 5–8). This equals to consider each instance graph $G = (V, E)$ inside a $|V| \times |V|$

Table 1: Size of the input data.

Graph	$ V $	k
Balls-into-bins	$\{50, 100, \dots, 1000\}$	$\{3, 6, 9\}$
	$\{10, 100, 1000, 10000\}$	$\{2, 4, \dots, 16\}$
Barabási–Albert	$\{50, 100, \dots, 1000\}$	$\{3, 6, 9\}$
	$\{10, 100, 1000, 10000\}$	$\{2, 4, \dots, 16\}$

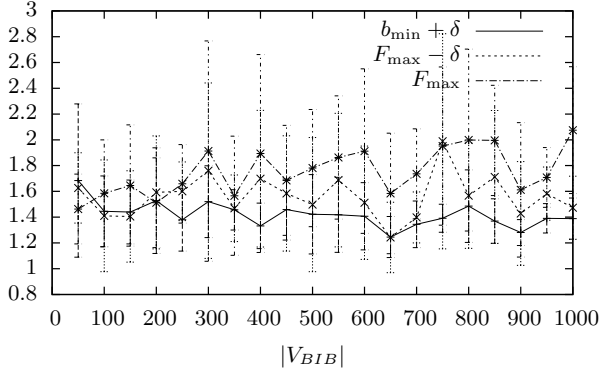


Figure 1: Graphs G_{BIB} : average value of approximation ratio for $|V_{BIB}| \in \{50, 100, \dots, 1000\}$, $k = 9$ and three values of required flow.

square instead of a unitary square. The figures show the average values and the standard deviations of the computed upper bound to the approximation ratio of our algorithm. Each figure contains three curves, one for each considered required flow. In particular, for each instance, we consider three possible values of required flow equally distributed in the interval $\{b_{\min}, \dots, F_{\max}\}$. Namely, the curves refer to $b_{\min} + \delta$, $F_{\max} - \delta$ and F_{\max} , as for b_{\min} we can compute the optimal value.

Figure 1 shows the average values and the standard deviations of the computed upper bounds on the approximation ratio as a function of the number of nodes in the network $|V_{BIB}|$, ranging from 50 to 1000, when the number of interfaces k is 9. The maximum value obtained is 3.94, achieved by an instance of 400 nodes and 8055 edges, when the required flow is F_{\max} . However, there are very few instances with an approximation ratio in $[3, 4)$. In detail, for 11 instances it is in $[3, 4)$, for 79 instances it is in $[2, 3)$, for 493 instances it is in $(1, 2)$ and for all the other 17 instances it ensures the optimal value. On average, the ratio is always smaller than 2.07. Moreover, these are only upper bounds to the real ratio as we computed the ratio between the objective function value of our algorithm and a lower bound to the optimum. The curves do not show a strict dependency from the number of nodes $|V_{BIB}|$. Conversely, there exists a small dependency from the required flow, that is the approximation ratio slightly increases with the required flow. The relevance of the obtained results is also given by the difference between the obtained upper bounds to the approximation ratios and the values of b_{\max} guaranteed by the theoretical analysis of [10]. The value b_{\max} can be in fact very much higher than the experimented results. Consider, for instance, that for the networks providing Figure 1, the average value for b_{\max} is more than 10.000. This confirms the interest in studying the algorithm for practical instances

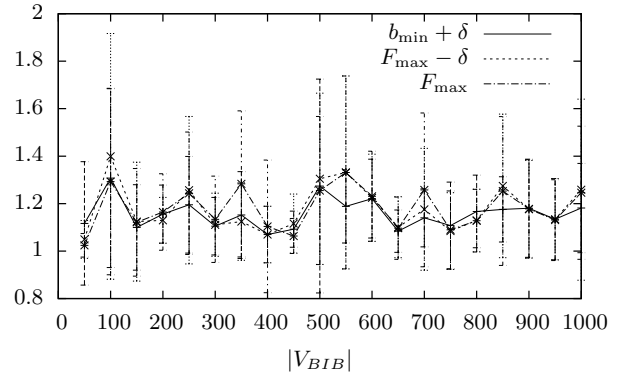


Figure 2: Graphs G_{BIB} : average value of approximation ratio for $|V_{BIB}| \in \{50, 100, \dots, 1000\}$, $k = 3$ and three values of required flow.

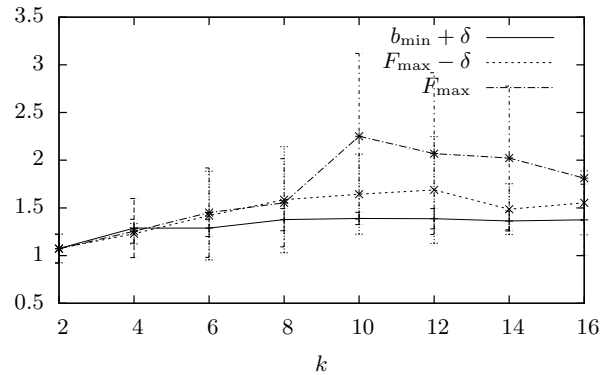


Figure 3: Graphs G_{BIB} : average value of approximation ratio for $|V_{BIB}| = 10000$, $k \in \{2, 4, \dots, 16\}$ and three values of required flow.

in order to better understand its performances.

Figure 2 shows the three curves when $k = 3$ and the other parameters are in the same setting as Figure 1. As expected, the approximation ratio is improved here. This is due to the fact that reducing the number of interfaces, the possible overhead at each node is also reduced. In detail, the maximum approximation ratio obtained is 2.54, achieved by an instance of 500 nodes and 5922 edges, when the required flow is F_{\max} . The upper bound to approximation ratio is in $[2, 3)$ for 12 instances, in $(1, 2)$ for 381 instances and the algorithm ensures the optimal value for the remainder 207 instances.

Figures 3 and 4 refer to the cases where $|V_{BIB}|$ is fixed to 10000 and 100, respectively, and the number of interfaces k ranges from 2 to 16. Also in this case, the approximation ratio is very small. In detail, in the worst cases it achieves 4.50 and 2.64, respectively. The curves show that there is not a strict dependency from the number of interfaces k , apart for small values of it ($k \leq 4$) and that, also in this case there exists a small dependency from the required flow. In fact, in Figure 3, the approximation ratio slightly increases with the required flow. However, in Figure 4 this phenomenon is not evident as the three curves intersect each other.

We can conclude that, in graphs G_{BIB} , the approximation

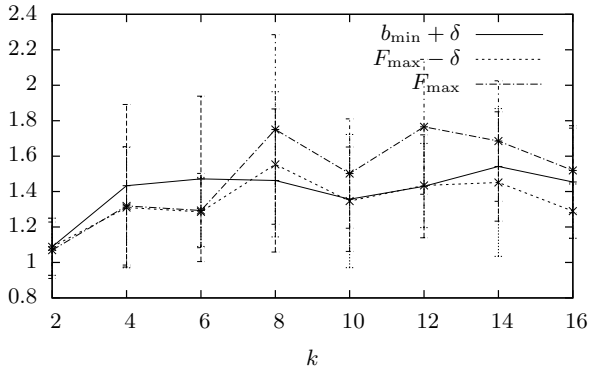


Figure 4: Graphs G_{BIB} : average value of approximation ratio for $|V_{BIB}| = 100$, $k \in \{2, 4, \dots, 16\}$ and three values of required flow.

ratio is always very small and it does not depend neither on the number of nodes nor on the number of interfaces, while there is a small dependency from the required flow.

Figures 5–8 show the experimental results in the same settings as Figures 1–4 for graphs G_{BA} . In these cases, the inferred properties do not change. In fact, the approximation ratio is small and it does not depend neither on the number of nodes nor on the number of interfaces. However, we can observe a worsening of the performances of the algorithm. In detail, although in most of the cases the approximation ratio is the same as for graphs G_{BIB} there are some instances where it is much higher than the average. For instance, Figure 6 shows a case where the average value is 6.48 and the standard deviation is 11.72 which is due to an instance where the approximation ratio is 41.40. Similar instances also appear in the experiments shown in Figures 5, 7, and 8. It is worth to note that the bad approximation bounds on these particular cases are mainly due to the bad estimation of the optimal value rather than the behavior of ALG_2 .

Concerning the execution time of the algorithm, Table 2, reports the worst case time for some large instances. It is clear that the algorithm is fast enough to be used in large scale networks. In detail, the computational time goes from few microseconds in the smaller instances to some seconds in large instances made of 10000 nodes and 16 interfaces.

4. CONCLUSION

We have considered two fundamental optimization problems which take into account bandwidth constraints in Multi-Interface Networks: $MFMI$ and $MCFMI$. In $MFMI$, we aim to activate a set of interfaces in the network in order to guarantee the maximal bandwidth between two given nodes. In $MCFMI$, we look for activating the cheapest set of interfaces among a network in order to guarantee a minimum bandwidth of communication between two specified nodes. The obtained results have shown that $MFMI$ is polynomially solvable while $MCFMI$ is NP -hard. However, we experimentally analyzed algorithm ALG_2 for $MCFMI$, showing that in practical cases it guarantees a low approximation ratio which allows us to use it in real-world. Further investigation for better performing approximation algorithms or heuristics remain challenging problems. Another interesting issue is to

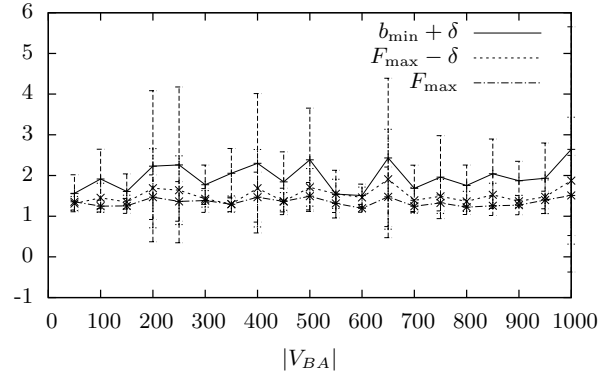


Figure 5: Graphs G_{BA} : average value of approximation ratio for $|V_{BA}| \in \{50, 100, \dots, 1000\}$, $k = 9$ and three values of required flow.

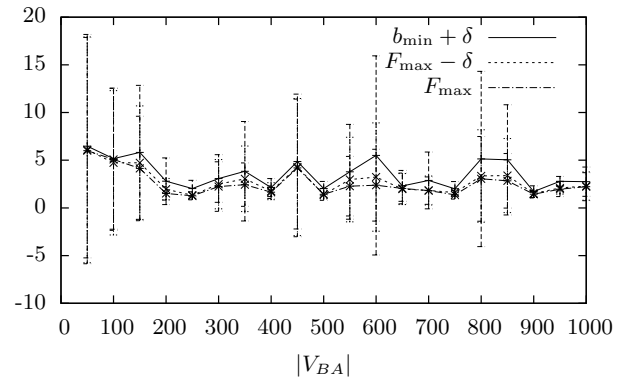


Figure 6: Graphs G_{BA} : average value of approximation ratio for $|V_{BA}| \in \{50, 100, \dots, 1000\}$, $k = 3$ and three values of required flow.

study the problem from a distributed point of view.

5. REFERENCES

- [1] LEMON graph library. <http://lemon.cs.elte.hu/trac/lemon/>.
- [2] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.
- [3] R. Albert and A.-L. Barabási. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
- [4] R. Albert and A.-L. Barabási. Statistical mechanics of complex network. *Reviews of Modern Physics*, 74:47–97, 2002.
- [5] S. Athanassopoulos, I. Caragiannis, C. Kaklamanis, and E. Papaioannou. Energy-efficient communication in multi-interface wireless networks. In *34th Proc. of Int. Symp. on Mathematical Foundations of Computer Science (MFCS)*, volume 5743 of *Lecture Notes in Computer Science*, pages 102–111. Springer, 2009.
- [6] P. Bahl, A. Adya, J. Padhye, and A. Walman. Reconsidering wireless systems with multiple radios. *SIGCOMM Comput. Commun. Rev.*, 34(5):39–46, 2004.

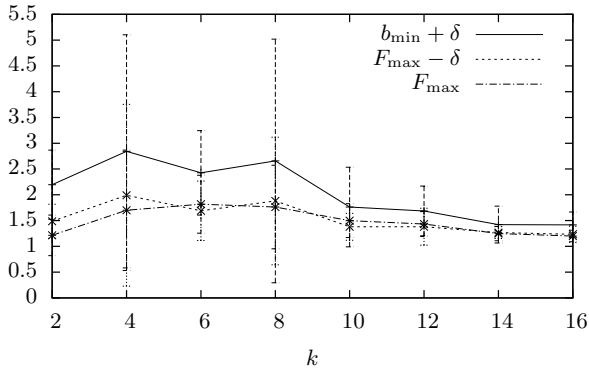


Figure 7: Graphs G_{BA} : average value of approximation ratio for $|V_{BA}| = 10000$, $k \in \{2, 4, \dots, 16\}$ and three values of required flow.

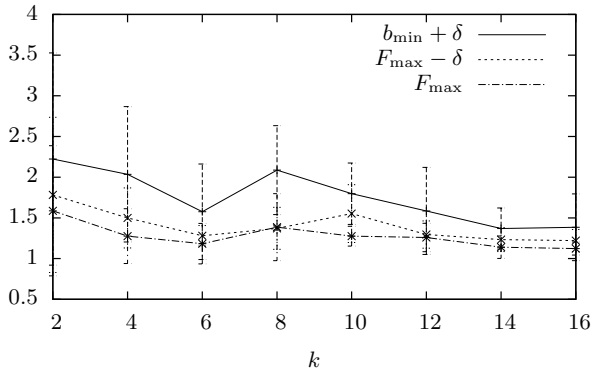


Figure 8: Graphs G_{BA} : average value of approximation ratio for $|V_{BA}| = 100$, $k \in \{2, 4, \dots, 16\}$ and three values of required flow.

- [7] M. Caporuscio, D. Charlet, V. Issarny, and A. Navarra. Energetic Performance of Service-oriented Multi-radio Networks: Issues and Perspectives. In *6th Int. Workshop on Software and Performance (WOSP)*, pages 42–45. ACM Press, 2007.
- [8] D. Cavalcanti, H. Gossain, and D. Agrawal. Connectivity in multi-radio, multi-channel heterogeneous ad hoc networks. In *IEEE 16th Int. Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1322–1326. IEEE, 2005.
- [9] G. D’Angelo, G. Di Stefano, and A. Navarra. Minimizing the Maximum Duty for Connectivity in Multi-Interface Networks. In *Proceedings of the 4th Annual International Conference on Combinatorial Optimization and Applications (COCOA)*, volume 6509 Part II of LNCS, pages 254–267. Springer, 2010.
- [10] G. D’Angelo, G. Di Stefano, and A. Navarra. Bandwidth constrained multi-interface networks. In *Proc. of 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, Lecture Notes in Computer Science. Springer, 2011. To appear.

Table 2: Average computational Time.

Graph	k	$ V $	time (ms)
Balls-into-bins	2	10	0.06
		100	1.52
		1000	55.24
		10000	1466.61
	16	10	0.38
		100	9.98
		1000	507.40
		10000	25031.20
Barabási-Albert	2	10	0.04
		100	0.26
		1000	3.72
		10000	101.00
	16	10	0.44
		100	5.85
		1000	107.85
		10000	4981.41

- [11] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University, 1963.
- [12] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of 10th annual international conference on Mobile computing and networking (MobiCom)*, pages 114–128. ACM, 2004.
- [13] A. Faragó and S. Basagni. The effect of multi-radio nodes on network connectivity—a graph theoretic analysis. In *IEEE Int. Workshop on Wireless Distributed Networks (WDM)*. IEEE, 2008.
- [14] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
- [15] R. Klasing, A. Kosowski, and A. Navarra. Cost Minimization in Wireless Networks with a Bounded and Unbounded Number of Interfaces. *Networks*, 53(3):266–275, 2009.
- [16] A. Kosowski, A. Navarra, and M. Pinotti. Exploiting Multi-Interface Networks: Connectivity and Cheapest Paths. *Wireless Networks*, 16(4):1063–1073, 2010.
- [17] E. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, and Winston, 1976.
- [18] A. Papadopoulos, J. McCann, and A. Navarra. Connectionless probabilistic (cop) routing: an efficient protocol for mobile wireless ad-hoc sensor networks. In *Proc. of 24th IEEE Int. Performance, Computing, and Communications Conference (IPCCC)*, pages 73–77. IEEE, 2005.
- [19] M. Raab and A. Steger. “balls into bins” – a simple and tight analysis. In *Proc. of 2nd Int. Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, volume 1518 of *Lecture Notes in Computer Science*, pages 159–170. Springer, 1998.