Flow problems in multi-interface networks

Gianlorenzo D'Angelo, Gabriele Di Stefano, Alfredo Navarra

Flow problems in multi-interface networks

Gianlorenzo D'Angelo, Gabriele Di Stefano, and Alfredo Navarra

***Abstract*—In heterogeneous networks, devices communicate by means of multiple wired or wireless interfaces. By switching among interfaces or by combining the available ones, each device might establish several connections. A connection may be established when the devices at its endpoints share at least one active interface.**

In this paper, we consider two fundamental optimization problems. In the first one (*Maximum Flow in Multi-Interface Networks*, $MFMI$), we aim to establish the maximal bandwidth that can be guaranteed between two given nodes of the input network. In the second problem (*Minimum-Cost Flow in Multi-Interface Networks*, $MCFMI$), 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 $MFMI$ is polynomially solvable while $MCFMI$ is NP-hard even for a bounded number of different interfaces and bounded degree networks. Moreover, we provide polynomial approximation algorithms for $MCFMI$ and exact algorithms for relevant sub-problems. Finally, we experimentally analyze the proposed approximation algorithm, showing that in practical cases it guarantees a low approximation ratio.

***Index Terms*—Multi-Interface Networks, Flow, Computational complexity, Approximation algorithms, Experimental analysis**

I. Introduction

The interest in heterogeneous 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. One of the most relevant property is the variety of the devices which might interact in order to exchange data. Heterogeneous networks are, in fact, composed of devices with different characteristics like computational power, energy consumption, communication interfaces, communication protocols, and so forth. In this paper, we are mainly interested in devices equipped with multiple interfaces (like Ethernet, ADSL, 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: the availability of an interface 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.

G. D'Angelo is with the MASCOTTE Project, I3S(CNRS/UNSA)/INRIA, France. e-mail: gianlorenzo.d_angelo@inria.fr

G. Di Stefano is with Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica, Università degli Studi dell'Aquila, Via Gronchi 18, I-67100, L'Aquila – Italy. e-mail: gabriele.distefano@univaq.it

A. Navarra is with Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli, 1 I-06123, Perugia – Italy. e-mail: alfredo.navarra@unipg.it

Preliminary results about this work have been presented in [1], [2].

We study communication problems in heterogeneous networks supporting multiple interfaces. In the considered model, a network is described by a graph $G = (V, E)$, where V represents the set of devices and E is the set of possible connections defined according to the distance between devices and the available interfaces that they share. Each $e \in E$ is associated with a set of interfaces $X(e)$ that are assigned to both its endpoints. The set of all the possible available interfaces in the network is then determined by $\bigcup_{e\in E} X(e)$; we denote the cardinality of this set by k. We say that a connection is established when the endpoints of the corresponding edge share at least one active interface. If an interface x is activated at both the endpoints of some edge $e = \{u, v\}$, then nodes u and v consume some energy $c(x)$ for maintaining x as active, and they provide a maximum communication bandwidth $b(x)$ with all their neighbors which share interface x. It follows that a device holding interface x has both the incoming and the outgoing bandwidths bounded by $b(i)$. In the paper, we assume that the connections are point-to-point. In this setting, we study two optimization problems whose aim is to guarantee a connection between two selected nodes $s, t \in V$, taking into account bandwidth constraints. First, we study the problem of finding the maximal possible bandwidth between two selected nodes $s, t \in V$. In detail, we consider all the interfaces of the network as active, so that all the connections in E are established. Then, we look for a suitable flow function that guarantees the maximum communication bandwidth between s and t. Successively, 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 network so that s is guaranteed to transfer data to t with a bandwidth of at least B. In general, the solution is not a path between s and t, but a more complex graph consisting of nodes with active interfaces might be required according to the topology and the available interfaces.

A. Related work

Multi-interface networks have recently been studied in a variety of contexts, usually focusing on the benefits of multiple interfaces available at each node. Many basic problems of standard network optimization can be reconsidered in such a setting [3], in particular, focusing on issues related to routing [4] and network connectivity [5], [6], [7]. The study of combinatorial problems on multi-interface networks has originated from [8]. That paper, as well as [5], [9], investigates the *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 established. *Connectivity* issues have been addressed in [5], [10], [11]. 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 [11], the attention has been devoted to the *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. Studies on the maximization of some network utility function (e.g., the throughput) while taking care of possible interferences between different wireless communications in multi-channel multi-radio wireless networks can be found in [12], [13], [14]. To the best of our knowledge, pure bandwidth issues have been never treated before in this context in terms of simple flow problems.

TABLE I

COMPLEXITY RESULTS ACHIEVED FOR *MCFMI* BY VARYING ON THE MAXIMUM DEGREE Δ AND THE NUMBER OF AVAILABLE INTERFACES k.

Δ	k	Complexity
$\Delta = 1$	Fixed	Optimally solvable in $O(1)$ time
	Unbounded	NP-hard (equiv. MinKnapsack), $(1+\epsilon)$-apx in $O(\frac{k^2}{\epsilon})$
$\Delta = 2$	Fixed	Optimally solvable in $O(\lvert V\rvert)$
	Unbounded	NP-hard; $(2+\epsilon)$-apx in $O(\lvert V\rvert\frac{k^2}{\epsilon})$ for paths
Fixed $\Delta \geq 3$	Fixed $k \geq 2$	NP-hard (from $X3C$)
	Fixed $k \geq 3$	Not apx within $\Omega(\log B)$, or within $\Omega(\log\log \lvert V\rvert)$
Any	$k = 1$	Opt. solvable in $O(\lvert V\rvert + \lvert E\rvert)$ (equiv. shortest path)
	Any	$\frac{b_{\max}}{M}$-apx (optimal for constant bandwidth)

B. Our results

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

The first problem, called *Maximum Flow in Multi-Interface Networks* ($MFMI$), aims to find the maximal communication bandwidth that can be guaranteed between two given nodes. Such problem is similar to the classical 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, a node v can communicate with many other nodes by means of a single interface i but, if v uses the whole bandwidth of i to transmit to (receive from, resp.) a neighbor u, it cannot use i to transmit to (receive from, resp.) another neighbor w, even if i belongs to both v and w. Therefore, we assume that the communications are point-to-point. We show that this problem is optimally solvable in polynomial time, and we provide 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* [15]. 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 there exists a one-to-one mapping between interfaces and connections, that is, each connection can be established by means of one interface different from any other, the two problems $MCFMI$ and Minimum Edge-Cost Flow coincide. Hence, it is not surprising that $MCFMI$ turns out to be NP-hard when the number k of interfaces is unbounded. However, in practical cases it is more realistic to consider a bounded number of interfaces. Despite the expectations, we show that the problem is NP-hard even when k is a fixed small number. In detail, we prove that the problem is NP-hard for any fixed $k \geq 2$ and $\Delta \geq 3$, where Δ is the maximum degree of the network, while it is polynomially solvable when $k = 1$, or $\Delta \leq 2$ and $k = O(1)$. Moreover, we show that the problem is not approximable within $\Omega(\log B)$ or $\Omega(\log\log |V|)$ for any fixed $k \geq 3$, $\Delta \geq 3$, unless $P = NP$. We then provide an approximation algorithm with ratio guarantee of $\frac{b_{\max}}{M}$, where $b_{\max}$ is the maximum communication bandwidth allowed among all the available interfaces and M is the greatest common divisor among the bandwidths allowed by the interfaces and B. Hence, when the bandwidth is constant for all the interfaces, the optimal solution is provided. We also focus on particular cases by providing complexity results and polynomial algorithms for $\Delta \leq 2$. Surprisingly, when k is unbounded and the network reduces to a single edge the problem remains NP-hard. Table I summarizes the results. Finally, we experimentally analyzed the $\frac{b_{\max}}{M}$-approximation algorithm, showing that, in practical cases, it guarantees a low approximation ratio which allows us to use it in real-world networks.

Outline: In the next section, we give the statements of the problems and introduce some useful notation. In Section III we give some preliminary results that will be used in the subsequent Sections IV and V. In Section IV, we study the computational complexity of the two problems by identifying the cases where they are NP-hard or solvable in polynomial time. In Section V, we study the approximation properties of the two problems by giving inapproximability lower bounds and approximation algorithms. In Section VI, we give an experimental study on one of the approximation algorithms proposed in Section V. Finally, in Section VII, we provide some concluding remarks.

II. DEFINITIONS AND NOTATION

Given a network, we denote by V the set of nodes. For each pair of nodes in V, the *sharing* function $X \colon V \times V \to 2^{\{1,2,\ldots,k\}}$ denotes the set of interfaces that the two nodes can use to communicate. Function X must satisfy the following properties: for each u in V, $X(u,u) = \emptyset$; for each u,v in V, $X(u,v) = X(v,u)$. Function X induces a global assignment of the interfaces to the nodes in V given in terms of an appropriate interface *assignment* function $W \colon V \to 2^{\{1,2,\ldots,k\}}$

defined as $W(v) = \bigcup_{u \in V} X(u,v)$. If an interface i is in $X(u,v)$ for some nodes u and v, then $i \in W(u)$, $i \in W(v)$, and u and v are close enough to communicate via interface i. It follows that, for each u, v in V, $X(u,v) \subseteq W(u) \cap W(v)$. The use of those functions represents a generalization of the model w.r.t. earlier works on the subject, including [1], [2]. Note that, the above definitions of V, X induce a graph $G = (V,E)$ where $\{u,v\} \in E$ if and only if $X(u,v) \neq \emptyset$. We say that G is induced by the sharing function X. Unless otherwise stated, the graph G representing the network is assumed to be undirected and connected. In the remainder, we denote by Δ the maximum node degree in G. The cost of activating an interface i is given by the cost function $c\colon \{1,2,\ldots,k\} \to \mathbb{Z}_0^+$ and it is denoted as $c(i)$. The bandwidth allowed by a given interface i is defined by the bandwidth function $b\colon \{1,2,\ldots,k\} \to \mathbb{Z}_0^+$ 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. However, *MFMI* does not require to minimize the cost of activating interfaces and therefore in this case we assume that all the interfaces are activated.

Problems *MFMI* and *MCFMI* are formulated as follows.

MFMI: Maximum Flow in Multi-Interface Networks

In: A set of nodes V, a source node $s \in V$, a target node $t \in V$, a set of interfaces $I = \{1,2,\ldots,k\}$, a sharing function $X\colon V \times V \to 2^I$, and an interface bandwidth function $b\colon I \to \mathbb{Z}_0^+$.

Sol: A flow function $f : V \times V \times I \to \mathbb{Z}_0^+$ such that:

1) $f(u,v,i) = -f(v,u,i) \ \forall\ u,v \in V,\ i \in I$;
2) $f(u,v,i) = 0$ if $X(u,v) = \emptyset\ \forall\ u,v \in V,\ i \in I$;
3) $\sum_{v \in V : f(u,v,i)>0} f(u,v,i) \le b(i)\ \forall u \in V,\ i \in I$; $\sum_{v \in V : f(v,u,i)>0} f(v,u,i) \le b(i)\ \forall u \in V,\ i \in I$;
4) $\sum_{v \in V, i \in I} f(u,v,i) = 0\ \forall u \in V \setminus \{s,t\}$;

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

MCFMI: Minimum-Cost Flow in Multi-Interface Networks

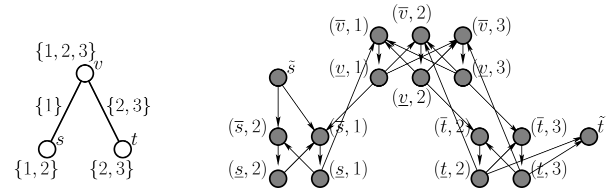
In: A set of nodes V, a source node $s \in V$, a target node $t \in V$, a set of interfaces $I = \{1,2,\ldots,k\}$, a sharing function $X\colon V \times V \to 2^I$, an interface cost function $c\colon I \to \mathbb{Z}_0^+$, an interface bandwidth function $b\colon I \to \mathbb{Z}_0^+$ and a bound $B \in \mathbb{Z}_0^+$.

Sol: An allocation of active interfaces $W_A\colon V \to 2^I$, $W_A(v) \subseteq \bigcup_{u \in V} X(u,v), \forall\ v \in V$, and a flow function $f : V \times V \times I \to \mathbb{Z}_0^+$ such that:

1) $f(u,v,i) = -f(v,u,i) \ \forall\ u,v \in V,\ i \in I$;
2) $f(u,v,i) = 0$ if $W_A(u) \cap W_A(v) \cap X(u,v) = \emptyset$ $\forall\ u,v \in V,\ i \in I$;
3) $\sum_{v \in V : f(u,v,i)>0} f(u,v,i) \le b(i)\ \forall u \in V,\ i \in I$; $\sum_{v \in V : f(v,u,i)>0} f(v,u,i) \le b(i)\ \forall u \in V,\ i \in I$;
4) $\sum_{v \in V, i \in I} f(u,v,i) = 0\ \forall u \in V \setminus \{s,t\}$;
5) $\sum_{v \in V, i \in I} f(s,v,i) = \sum_{v \in V, i \in I} f(v,t,i) \ge B$.

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

Fig. 1. The graph G and its transformation in the direct graph G'. $W(s) = \{1,2\}$, $W(v) = 1,2,3$, $W(t) = X(v,t) = \{2,3\}$, and $X(s,v) = \{1\}$.



For both problems *MFMI* and *MCFMI*, we denote by $G = (V,E)$ the graph induced by the sharing function X, also referred as the *input graph*. Graph G can be easily computed from X in time $O(|V| + |E|)$. Note that we can consider two variants of the *MCFMI* problem: the parameter k can be considered as part of the input (this is called the *unbounded case*), or k may be a fixed constant (the *bounded case*). In both cases we assume $k \ge 2$, since the case $k = 1$ admits an obvious solution given by a *shortest path* connecting s to t of maximum bandwidth $b(1)$. The case where the cost function is constant for each interface is called the *unit cost* case.

III. Preliminary results

The algorithms given in this paper for the general cases of *MFMI* and *MCFMI* are both based on a transformation of the graph $G = (V,E)$ into a directed graph $G' = (V', A)$. G' is defined so that bandwidths and costs are associated to arcs rather than to interfaces. 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. As each of these arcs is associated with the cost and the bandwidth of the interface it represents, the activation of an interface is modeled with the usage of one of these arcs, preserving the bandwidth constraints and the activation costs. Moreover, although the arcs are directed, the possibility to communicate towards and from every node of the original graph is preserved (see also Fig. 1).

Formally, for each node $v \in V$ and each interface $i \in W(v)$, there are two nodes in V' denoted as $(\overline{v}, i)$ and $(\underline{v}, i)$:

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

The arcs are the following:

$$A = \{((\overline{v}, i), (\underline{v}, i)) \mid v \in V, i \in W(v)\} \cup$$
$$\{((\underline{v}, i), (\overline{v}, j)) \mid v \in V, i, j \in W(v) \text{ s.t. } i \neq j\} \cup$$
$$\{((\underline{u}, i), (\overline{v}, i)) \mid i \in X(u,v)\} \cup$$
$$\left\{(\tilde{s}, (\overline{s}, i)), ((\underline{t}, j), \tilde{t}) \mid i \in W(s), j \in W(t)\right\}.$$

The capacity of each arc $((\overline{v}, i), (\underline{v}, i))$ is set to $b'((\overline{v}, i), (\underline{v}, i)) = b(i)$ 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 $a = ((\overline{v}, i), (\underline{v}, i))$ is set to $c(i)$ and it is 0 for the remaining arcs.

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

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

The allocation of active interfaces at node u for *MCFMI* is defined as $W_A(u) = \{i \in W(u) \mid \exists v \in V \text{ s.t. } f(u,v,i) \neq 0\}$. In order to address concurrently *MCFMI* and *MFMI*, we define W_A for *MFMI* as equivalent to the assignment function W induced by the sharing function X. Note that both functions f and W_A can be computed in polynomial time once function f' is known.

The next lemma shows that, if we apply the above transformation to an instance I of *MFMI* or *MCFMI* and we compute a flow function f and an assignment of interfaces W_A for I by using the above definition on some flow function f' of G', then f satisfies Properties 1–4 needed by both the definitions of *MFMI* and *MCFMI*.

Lemma 3.1: Let f and W_A be a flow function and an assignment of interfaces defined as above, then

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

Proof: We recall that, by definition of a flow function for a directed flow network:

$$0 \leq f'(x,y) \leq b'(x,y), \qquad \text{for each } (x,y) \in A \qquad \text{(a)}$$
$$f'(x,y) = -f'(y,x), \qquad \text{for each } x,y \in V' \qquad \text{(b)}$$
$$\sum_{x \in V'} f'(x,y) = 0, \qquad \text{for each } y \in V' \setminus \{\tilde{s},\tilde{t}\}. \quad \text{(c)}$$

In the following we prove the four properties.

1) If $i \notin X(u,v)$, then $f(u,v,i) = f(v,u,i) = 0$. In fact, by definition of f and by Property (b)

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

2) 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, $f(u,v,i) = 0$.

3) We formally provide only the proof for the first inequality as the second one follows from similar arguments. By definition of f, $\sum_{v \in V : f(u,v,i)>0} f(u,v,i) = \sum_{v \in V : f(u,v,i)>0} \left(f'((\underline{u},i),(\overline{v},i)) - f'((\underline{v},i),(\overline{u},i))\right)$.

By Property (a), for each $v \in V$, $f'((\underline{v},i),(\overline{u},i)) \geq 0$ and $f'((\underline{u},i),(\overline{v},i)) \geq 0$, then $\sum_{v \in V : f(u,v,i)>0} \left(f'((\underline{u},i),(\overline{v},i)) - f'((\underline{v},i),(\overline{u},i))\right) \leq \sum_{v \in V : f(u,v,i)>0} f'((\underline{u},i),(\overline{v},i))$.

By the definition of b' and Property (a),

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

By Property (c), applied to $(\underline{u},i)$, $\sum_{v \in V} f'((\underline{u},i),(\overline{v},i)) = f'((\overline{u},i),(\underline{u},i)) - \sum_{j \in I \setminus \{i\}} f'((\underline{u},i),(\overline{u},j))$. Again by Property (a), $f'((\underline{u},i),(\overline{u},j)) \geq 0$, for each $j \in I \setminus \{i\}$, then $f'((\overline{u},i),(\underline{u},i)) - \sum_{j \in I \setminus \{i\}} f'((\underline{u},i),(\overline{u},j) \leq f'((\overline{u},i),(\underline{u},i)) \leq b(i)$. The last inequality directly follows from Property (a).

4) By definition of f, $f(u,v,i) = 0$ if $v = u$ or $i \notin X(u,v)$, hence $\sum_{v \in V, i \in I} f(u,v,i) = \sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u,v)}} (f'((\underline{u},i),(\overline{v},i)) - f'((\underline{v},i),(\overline{u},i)))$.

By Property (c), applied to nodes $(\underline{u},i)$, $\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u,v)}} f'((\underline{u},i),(\overline{v},i)) =$
$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u,v)}} \left(f'((\overline{u},i),(\underline{u},i)) - \sum_{\substack{j \in X(u,v) \\ j \neq i}} f'((\underline{u},i),(\overline{u},j)) \right)$.

Again, by Property (c), applied to nodes $(\overline{u},i)$, $\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u,v)}} f'((\underline{v},i),(\overline{u},i)) =$
$\sum_{\substack{v \in V \setminus \{u\} \\ i \in X(u,v)}} \left(f'((\overline{u},i),(\underline{u},i)) - \sum_{\substack{j \in X(u,v) \\ j \neq i}} f'((\underline{u},j),(\overline{u},i)) \right)$.

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

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

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 $f'((\underline{u},q),(\overline{u},p)) - f'((\underline{u},p),(\overline{u},q))$, on the contrary, when $i = q$ and $j = p$, it is $f'((\underline{u},p),(\overline{u},q)) - f'((\underline{u},q),(\overline{u},p))$ and hence the overall sum is 0. ∎

IV. Computational complexity

In this section we study the computational complexity of *MFMI* and *MCFMI*. We first prove that *MFMI* is optimally solvable in polynomial time in the general case, we then focus on *MCFMI*. We prove that *MCFMI* is *NP*-hard even in the restricted case of unit cost, fixed $k \geq 2$, and fixed $\Delta \geq 3$. Then we consider graphs of bounded degree $\Delta \leq 2$. As announced in Table I, we prove that, when the number of interfaces k is fixed, the problem can be optimally solved in polynomial time. On the other hand, if k is unbounded, we show that the problem remains *NP*-hard. Moreover, when the bandwidth function b is a constant, then *MCFMI* is solvable in polynomial time (see Corollary 5.3 in the next section).

A. General Case

Let $\mathcal{A}$ be an algorithm that finds a maximum flow in a graph $H = (V_H, E_H)$ in polynomial time $P_{\mathcal{A}}(|V_H| + |E_H|)$.

Theorem 4.1: *MFMI* is optimally solvable within $O(|V|k^2 + |E| + P_{\mathcal{A}}(|V|k^2 + |E|))$ time.

Proof: Given an instance I_1 of *MFMI*, the algorithm first transforms the graph G and the function b of I_1 into a graph G' and a function b' as described in Section III, obtaining an instance I_2 of the classical maximum flow problem. Then, in polynomial time, it finds a maximal flow function f' for I_2 by using a maximum flow algorithm. Finally, the algorithm obtains a maximal flow function f for I_1 from f' by using the transformation given in Section III. The computational time required by such an algorithm is given by the cost of transforming I_1 into I_2 and that of solving I_2. As the graph defined for I_2 has $O(|V|k)$ nodes and $O(|V|k^2 + |E|)$ edges,

Fig. 2. The subgraphs used in the proofs of Theorems 4.2 and 5.1.

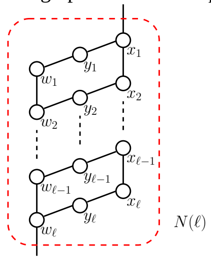
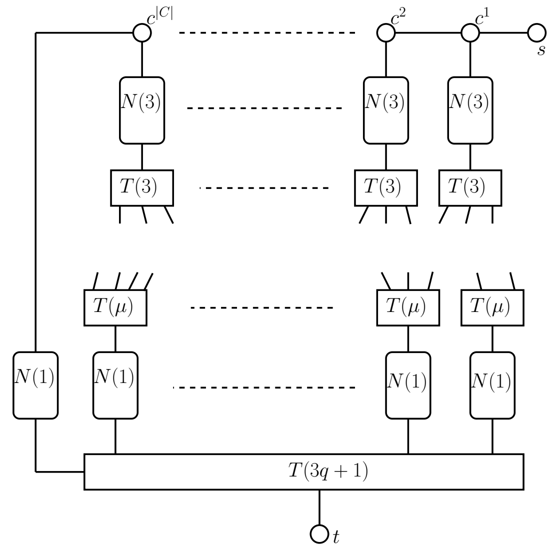


Fig. 3. The graph G in the transformation from $X3C$ to $MCFMI_D$.



the first cost is $O(|V|k^2 + |E|)$ while the second one is $O(P_{\mathcal{A}}(|V|k^2 + |E|))$.

We now show that f is an optimal solution for I_1. By Lemma 3.1, f satisfies properties 1–4 of the definition of $MFMI$. We show that f is maximal by contradiction. We recall that by definition of maximal flow function, $\sum_{v \in V'} f'(\tilde{s}, v)$ is maximal. By contradiction, let us suppose that there exists a flow function $f'' : V \times V \times I \to \mathbb{Z}_0^+$ 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 $f''' : V' \times V' \to \mathbb{Z}_0^+$ for I_2 as follows, if $i \in X(u,v)$,

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

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

Similar arguments as Lemma 3.1 can be used to show that f''' fulfills the properties of flow functions and that

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

It follows that $\sum_{v \in V'} f'''(\tilde{s}, v) = \sum_{v \in V, i \in I} f''(s,v,i) > \sum_{v \in V, i \in I} f(s,v,i) = \sum_{v \in V'} f'(\tilde{s}, v)$, a contradiction to the maximality of f'. ■

Theorem 4.2: MCFMI is strongly NP-hard even when restricted to the unit cost interface case for any fixed $\Delta \geq 3$ and $k \geq 2$.

Proof: We prove that the underlying decisional problem, denoted by $MCFMI_D$, is in general NP-complete. We need to add one further bound $D \in \mathbb{Z}_0^+$ such that the problem consists in deciding whether there exists an activation function which induces a total cost of the active interfaces of at most D.

Given an allocation function of active interfaces for an instance of $MCFMI_D$, checking whether the induced subgraph allows a bandwidth greater than or equal to B of total cost smaller than or equal to D requires linear time in the number of edges of the input graph G. Then, $MCFMI_D$ is in NP. The proof proceeds by a polynomial reduction from the well-known *Exact Cover by* 3*-Sets* problem. The problem is known to be NP-complete [15] and it can be stated as follows:

$X3C$: Exact Cover by 3-Sets

***Input*:** Set S with $|S| = 3q$ and a collection C of 3-element subsets of S.

***Question*:** Is there an exact set cover for S, i.e. a subset $C' \subseteq C$ such that $|C'| = q$ and every element of S belongs to exactly one member of C'?

Given an instance of $X3C$, we construct an instance of $MCFMI_D$ where the graph G consists of copies of subgraphs $N(\ell)$ and $T(\ell)$, $\ell \geq 1$ (see Fig. 2). Subgraph $N(\ell)$ consists of 3ℓ nodes $\{x_1, x_2, \ldots, x_\ell\} \cup \{y_1, y_2, \ldots, y_\ell\} \cup \{w_1, w_2, \ldots, w_\ell\}$ and edges $\{x_i, x_{i+1}\}$, $\{w_i, w_{i+1}\}$, for $i = 1, 2, \ldots, \ell - 1$ and $\{x_i, y_i\}$, $\{y_i, w_i\}$, for $i = 1, 2, \ldots, \ell$. Subgraph $T(\ell)$ is a binary tree consisting of a complete binary tree BT with $2^{\lceil \log_2 \ell \rceil} - 1$ nodes, and ℓ nodes adjacent to the leaves of BT. These nodes are the only leaves of $T(\ell)$, i.e. every leaf of BT is connected to at least one leaf of $T(\ell)$. We call r the root of $T(\ell)$. Note that, each path from r to a leaf of $T(\ell)$ is constituted of $\lceil \log_2 \ell \rceil + 1$ nodes. Moreover, when $\ell = 1$, BT is empty and $T(\ell)$ consists of a single node.

For the sake of simplicity, in this proof we first define the graph G and then we define functions W and X accordingly. See Fig. 3 for a visualization of G. Let s and t be two nodes of G. For each element C_i of C, $i = 1, 2, \ldots, |C|$, G contains a node c^i, a copy of $N(3)$, denoted as $N^i(3)$ and a copy of $T(3)$, denoted as $T^i(3)$, with root r^i and leaves l_1^i, l_2^i, l_3^i. Nodes x_1^i and w_3^i of $N^i(3)$ are adjacent to c^i and r^i, respectively. All nodes c^i form a path P in G, that is $\{c^i, c^{i+1}\}$ is an edge of G, for $i = 1, 2, \ldots, |C| - 1$. Node s of G is adjacent to c^1, while node $c^{|C|}$ is adjacent to node x_1^0 belonging to a copy $N^0(1)$ of $N(1)$ with nodes x_1^0, y_1^0 and w_1^0.

Let e_j, $j = 1, 2, \ldots, 3q$, be the elements of S and let $\mu(e_j)$ be the number of sets $C_i \in C$ containing e_j, for each j. Let $\mu = \max_j\{\mu(e_j)\}$. For each element e_j, G contains a copy of $T(\mu)$, called $T^j(\mu)$, with root r^j, and a copy $N^j(1)$ of $N(1)$, with nodes x_1^j, y_1^j and w_1^j. Root r^j is adjacent to $x_1^j \in N^j(1)$, for each $j = 1, 2, \ldots, 3q$. If e_j is in C_i, for some i and j, then there is an edge from a leaf of $T^i(3)$ to a leaf of $T^j(\mu)$.

These edges are pairwise disjoint. Note that, even if each leaf of $T^i(3)$, $i = 1, 2, \dots, |C|$ is adjacent to a leaf in $T^j(\mu)$, for some $j \in \{1, 2, \dots, 3q\}$, the contrary is not true: there could be a leaf of $T^j(\mu)$, for some j, not adjacent to any leaf of $T^i(3)$, $i = 1, 2, \dots, |C|$.

G also contains a copy of $T(3q+1)$, having the root adjacent to node t, and leaves adjacent to nodes w_1^j, $j = 0, 1, \dots, 3q$. The set of interfaces I is $\{1, 2\}$, with $c(1) = c(2) = 1$ and $b(1) = 1$, $b(2) = 3q + 1$.

Function W is defined as follows. All the nodes in G have interface 2 apart from nodes labeled y in the copies of $N(1)$ and $N(3)$. All the nodes in the copies of $N(1)$ and $N(3)$ have interface 1: no further node in G has interface 1. Function X is defined as follows, given two nodes u, v in G,

$$X(u,v) = \begin{cases} W(u) \cap W(v) & \text{if } \{u,v\} \in E \\ \emptyset & \text{otherwise.} \end{cases}$$

When all the interfaces of the nodes in copies of $N(\ell)$ ($T(\ell)$, resp.), for a certain $\ell \geq 0$, are active the total cost is 5ℓ ($2^{\lceil \log_2 \ell \rceil} - 1 + \ell$, resp.). In $T(\ell)$, when only the interfaces of the nodes in a single path from r to a leaf are active, the total cost is $\lceil \log_2 \ell \rceil + 1$. Let $B = 3q + 1$ and $D = |C| + q(42 + 3\lceil \log_2 \mu \rceil) + 2^{\lceil \log_2 (3q+1) \rceil} + 7$.

Assume that $X3C$ has a positive answer, i.e., there exists an exact set cover $C' = \{C_{i_1}, C_{i_2}, \dots, C_{i_q}\} \subseteq C$ for S. We show that also $MCFMI_D$ has a positive answer, i.e., there exists an activation function W_A of the available interfaces such that the bandwidth allowed from s to t is bigger than or equal to B and the total cost is smaller than or equal to D. Function W_A is defined as follows. Along with interfaces of nodes s, t, all the interfaces of nodes in $T(3q+1)$, $N^j(1)$, $j = 0, 1, \dots, 3q$, and c^i, $i = 1, 2, \dots, |C|$, are active. All the interfaces of nodes in $N^{i_j}(3)$ and $T^{i_j}(3)$, for each $C_{i_j} \in C'$, $j = 1, 2, \dots, q$, are active. Moreover, if $e_i \in S$ is covered by $C_j \in C'$, then all the interfaces of nodes in $T^j(\mu)$ belonging to the path from r^j to a leaf in $T^i(3)$ are active. No further interface is active. The flow function is defined as 1 in nodes y of active copies of $N(1)$ and $N(3)$ and in the remainder of G it is defined to satisfy the flow conservation constraints.

The total cost of active interfaces is given by 2, for nodes s and t; $|C|$, for nodes $c^i \in P$, $i = 1, 2, \dots, |C|$; $15q + 6q$ for nodes in $N^{i_j}(3)$ and $T^{i_j}(3)$, $j = 1, 2, \dots, q$; $3q(\lceil \log_2 \mu \rceil + 1)$ for nodes in $T^j(\mu)$, $j = 1, 2, \dots 3q$; $5(3q + 1)$ for nodes in $N^j(1)$, $j = 0, 1, \dots 3q$; and $2^{\lceil \log_2 (3q+1) \rceil} + 3q$ for nodes in $T(3q + 1)$. Summing up all the values we obtain a cost equal to D.

Regarding the total bandwidth, note that a copy of $N(\ell)$ has a maximum bandwidth of ℓ. As $X3C$ has a positive answer, each element of S is covered, then the flow through each subgraph $N^j(1)$, $j = 1, 2, \dots, 3q$, is exactly $3q$. As all the interfaces in P are active, we also have a flow of $3q + 1$ through $N^0(1)$ that reaches t through the $T(3q+1)$ subgraph. Then $MCFMI_D$ has a positive answer.

Now, let us assume we have a positive answer to $MCFMI_D$. As the total flow received by t is greater than or equal to $B = 3q+1$, there is a flow of value 1 in each subgraph $N^j(1)$, $j = 0, 1, \dots, 3q$, meaning that each element of S is covered. Let us suppose, by contradiction, that the flow reaching the $N^j(1)$, $j = 1, 2, \dots, 3q$ subgraphs, implies the activation of the interfaces in $q' > q$ subgraphs among the $N^i(3)$, $i = 1, 2, \dots, |C|$ copies of $N(3)$. In this case there will be q'_1 subgraphs having one unit of flow, q'_2 subgraphs having 2 units of flow, and q'_3 subgraphs having 3 units of flow such that $q'_1 + 2q'_2 + 3q'_3 = 3q$.

The total cost for the interfaces activation is: 2, for nodes s and t; $|C|$, for nodes in P (all the interfaces in P are active as $N^0(1)$ receives one unit of flow); $7q'_1 + 11q'_2 + 15q'_3$ for nodes in $N^i(3)$; $6q$ for nodes in $T^i(3)$, $i = 1, 2, \dots, q$; $3q(\lceil \log_2 \mu \rceil + 1)$ for nodes in $T^j(\mu)$, $j = 1, 2, \dots 3q$; $5(3q + 1)$ for nodes in $N^j(1)$, $j = 0, 1, \dots 3q$, and $2^{\lceil \log_2 (3q+1) \rceil} + 3q$ for nodes in $T(3q + 1)$.

Then the total cost is $|C| + q(27q + 3\lceil \log_2 \mu \rceil) + 2^{\lceil \log_2 (3q+1) \rceil} + 7 + 7q'_1 + 11q'_2 + 15q'_3$. As $7q'_1 + 11q'_2 + 15q'_3 > 5(q'_1 + 2q'_2 + 3q'_3) = 15q$, the total cost is greater than D, a contradiction. Hence there are exactly q subgraphs $N^{i_j}(3)$, $j = 1, 2, \dots, q$ with 3 units of flow each and the corresponding sets C_{i_j}, $j = 1, 2, \dots, q$, represent a solution for $X3C$. ∎

B. Particular cases for *MCFMI*, $\Delta \leq 2$

Theorem 4.2 shows that *MCFMI* is *NP*-hard even for fixed $\Delta \geq 3$ and $k \geq 2$. As the case where $k = 1$ is trivial, we now focus on the case that $\Delta \leq 2$. For $\Delta \leq 1$, the input graph can be composed of either one single node or two nodes connected by one edge. In the first case, there are no interfaces to be activated, as the source and the destination coincide. In the second case, the problem already starts to be interesting.

Theorem 4.3: *MCFMI* is polynomially solvable within $O(1)$ time in the bounded case with $\Delta = 1$.

Proof: *MCFMI* can be solved by an exhaustive search among all the possible combinations of interfaces shared by s and t. The number of such combinations is $O(2^k)$. Among them, a resolution algorithm has to choose the cheapest one that guarantees at least B bandwidth. ∎

For the unbounded case, i.e., when k is not a given constant, the same arguments of Theorem 4.3 do not apply to *MCFMI* as the provided algorithm would show an exponential behavior. Surprisingly, in this setting the problem turns out to be already *NP*-hard by means of a simple polynomial transformation from the well known Minimization Knapsack problem [16], [17].

$MinKP$: Minimization Knapsack	
In.:	An integer $d \in \mathbb{Z}_0^+$ and a set of n items, each one having weight $w_i \in \mathbb{Z}_0^+$ and profit $p_i \in \mathbb{Z}_0^+$, $i = 1, 2, \dots, n$.
Sol.:	An allocation of variables $y_i \in \{0, 1\}$, for $i = 1, 2, \dots, n$, such that $\sum_{i=1}^n w_i y_i \geq d$
Aim:	Minimize $\sum_{i=1}^n p_i y_i$.

$MinKP$ problem is the corresponding minimization version of the Knapsack problem. In other words, the goal is to minimize the profits of the items that remain out of the knapsack. If x_i, $i = 1, 2, \dots, n$, are the variables selecting the items for the classical knapsack problem and $c \in \mathbb{Z}_0^+$ its capacity, then the problem can be solved by means of $MinKP$, by setting $d = \sum_{i=1}^n w_i - c$ and $y_i = 1 - x_i$, $i = 1, 2, \dots, n$.

When $\Delta = 1$, that is when the input graph G consists of a single edge from s to t, the required solution must select a subset of interfaces among the ones shared by s and t in such a way that a bandwidth of B is guaranteed, and the cost for activating such interfaces is minimized. Intuitively, this particular case of *MCFMI* is equivalent to the $MinKP$ problem.

Theorem 4.4: *MCFMI* is polynomially equivalent to $MinKP$ in the unbounded case with $\Delta = 1$.

Proof: We have to show that there exist two polynomial time algorithms $\mathcal{A}$ and $\mathcal{B}$ such that, for each instance I_1 of $MinKP$, $\mathcal{A}(I_1)$ returns an instance I_2 of *MCFMI*, for any solution σ' of I_2, $\mathcal{B}(\sigma') = \sigma$ is a solution for I_1, and the values of solutions σ and σ' are equal. Moreover, we have to show that there exist two polynomial time algorithms $\mathcal{A}^{-1}$ and $\mathcal{B}^{-1}$ such that, for each instance I_2 of *MCFMI*, $\mathcal{A}^{-1}(I_2)$ returns an instance I_1 of $MinKP$, for any solution σ of I_1, $\mathcal{B}^{-1}(\sigma) = \sigma'$ is a solution for I_2, and the values of solutions σ and σ' are equal.

We now show the first part of the above statement by defining the polynomial algorithms $\mathcal{A}$ and $\mathcal{B}$. Given an instance I_1 of $MinKP$, we consider an instance I_2 of *MCFMI* made of nodes s and t, and, for each item i of I_1, an interface i shared between s and t with cost $c(i) = \frac{1}{2}p_i$ and bandwidth $b(i) = w_i$. Hence $W(s) = W(t)$, moreover, function X is defined as $X(s,t) = W(s)$. Finally, let $k = n$ and $B = d$. Note that, if, for some i, p_i is an odd number, we can multiply all the profits p_i of a factor 2 in order to have $c(i) \in \mathbb{Z}_0^+$ for each $i = 1, 2, \ldots, n$. This does not affect the generality of the proof as it is enough to divide by 2 the value of the objective function for the solution of I_1 which will be defined in the following. A feasible solution for I_2 selects a set of interfaces W, by means of an activation function, in such a way that $B \leq \sum_{i \in W} b(i)$. As $d = B \leq \sum_{i \in W} b(i) = \sum_{i \in W} w_i$ and the cost of activating interfaces in W at both s and t is $2\sum_{i \in W} c(i) = \sum_{i \in W} p_i$, we can define algorithm $\mathcal{B}$ as the algorithm which selects items W in order to output a solution for I_1. Finally, both $\mathcal{A}$ and $\mathcal{B}$ are polynomial time algorithms. This proves the first part of the theorem. For the second part of the theorem, it is enough to note that algorithms $\mathcal{A}$ and $\mathcal{B}$ can be naturally inverted. ■

Corollary 4.5: *MCFMI* is *NP*-hard in the unbounded case with $\Delta = 1$ and it admits a pseudo-polynomial-time algorithm.

For $\Delta = 2$, the input graph of *MCFMI* is either a path or a cycle. Clearly, from Corollary 4.5, *MCFMI* remains *NP*-hard in the unbounded case. The following theorem gives a polynomial time algorithm for the bounded case. In the next section, we will derive a pseudo-polynomial-time algorithm for the unbounded case.

In the remainder, for a set of interfaces W, we denote as $c(W)$ the cost of activating the interfaces in W, formally: $c(W) = \sum_{i \in W} c(i)$.

Theorem 4.6: *MCFMI* is solvable within $O(|V|)$ time in the bounded case when the input graph is a path.

Proof: Let us denote the input path as a sequence of n nodes: $s \equiv x_0, x_1, \ldots, x_{n-1} \equiv t$. Given a node x_ℓ, $1 \leq \ell \leq n-1$, $\gamma(x_\ell)$ denotes the set of sub-sets of interfaces of x_ℓ, shared with $x_{\ell-1}$, whose total bandwidth is greater than or equal to B, formally: $\gamma(x_\ell) = \left\{W \subseteq X(x_\ell, x_{\ell-1}) \;\middle|\; \sum_{i \in W} b(i) \geq B\right\}$. Then, the minimum cost is given by:

$$\begin{aligned} \text{OPT} = \min \{ & 2c(W_1) + c(W_2 \setminus W_1) + c(W_2) + \ldots \\ & \ldots + c(W_{n-1} \setminus W_{n-2}) + c(W_{n-1}) \mid \\ & W_1 \in \gamma(x_1), W_2 \in \gamma(x_2), \ldots, W_{n-1} \in \gamma(x_{n-1})\}, \end{aligned}$$

where $2c(W_1)$ is the cost of interfaces used to connect x_0 to x_1 and $c(W_\ell \setminus W_{\ell-1}) + c(W_\ell)$ is the cost of interfaces used to connect $x_{\ell-1}$ to x_ℓ, $2 \leq \ell \leq n-1$. In particular, $c(W_\ell \setminus W_{\ell-1})$ is the cost of activating in $x_{\ell-1}$ the interfaces in W_ℓ not contained in $W_{\ell-1}$ and $c(W_\ell)$ is the cost of activating interfaces W_ℓ in x_ℓ.

For each $1 \leq \ell \leq n-1$, let us define the function $C_\ell : \gamma(x_\ell) \to \mathbb{Z}_0^+$ as the minimum cost needed to establish a communication path from s to node x_ℓ with bandwidth guarantee greater than or equal to B by activating interfaces $W \in \gamma(x_\ell)$ in x_ℓ, formally:

$$\begin{aligned} C_\ell(W) = \min \{ & 2c(W_1) + c(W_2 \setminus W_1) + c(W_2) + \ldots \\ & \ldots + c(W \setminus W_{\ell-1}) + c(W) \mid \\ & W_1 \in \gamma(x_1), W_2 \in \gamma(x_2), \ldots, W_{\ell-1} \in \gamma(x_{\ell-1})\}. \end{aligned}$$

By definition, $\text{OPT} = \min_{W \in \gamma(x_{n-1})} C_{n-1}(W)$. Hence it is enough to show that functions C_ℓ, for each $1 \leq \ell \leq n-1$, can be computed in $O(n)$ time. By cut-and-paste arguments, it follows that:

$$\begin{aligned} C_\ell(W) = \min_{W_{\ell-1} \in \gamma(x_{\ell-1})} & \{C_{\ell-1}(W_{\ell-1}) \\ & + c(W \setminus W_{\ell-1}) + c(W)\}. \end{aligned}$$

Therefore, functions C_ℓ, can be computed by using dynamic programming starting from $C_1(W) = 2c(W)$, for each $W \in \gamma(x_1)$. Moreover, as k is a bounded constant, $|\gamma(x_\ell)| \leq 2^k = O(1)$. Hence, given $1 \leq \ell \leq n-1$ and $W \in \gamma(x_\ell)$, computing $C_\ell(W)$ requires $O(1)$ time and computing function C_ℓ requires $O(1)$ time. Then, all the functions C_ℓ, for all $1 \leq \ell \leq n-1$, can be computed in $O(n)$ time. ■

When the input graph is a cycle, since there are two paths from s to t, it is not always clear how the bandwidth B must be split among the two possible ways. However, the following theorem can be stated for the bounded case.

Theorem 4.7: *MCFMI* is solvable within $O(|V|)$ time in the bounded case when the input graph is a cycle.

Proof: Let P_1 and P_2 be the two edge-disjoint paths from s to t composing the input cycle. As by definition, $b(i) \in \mathbb{Z}_0^+$, $1 \leq i \leq k$, the required flow B is provided by summing two integers β_1 and β_2 that are the contributions to the total flow passing via P_1 and P_2, respectively. The values β_1 and β_2 vary among all the integers obtainable by summing the bandwidths provided by each possible subset of interfaces, i.e., β_1 and β_2 can assume at most 2^k values. For each subset of interfaces of s and for each subset of interfaces of t, the algorithm proposed by Theorem 4.6 is applied to solve the *MCFMI* instance arising for P_1 with bound β_1, and the one arising for P_2 with bound $\beta_2 = B - \beta_1$. The over all trials are at most 2^{2k}, each of them requires 2^k tests, one for each possible value of β_1. As $k = O(1)$, then $2^{3k} = O(1)$. Among the obtained solutions, we choose the cheapest one which guarantees a flow of at least B from s to t. Such algorithm requires to run $O(1)$ times the algorithm in Theorem 4.6 and hence it requires $O(|V|)$ overall computational time. ■

V. Approximation

In this section, we study the approximation properties of *MCFMI*. We first show that, unless $P = NP$, *MCFMI* cannot be approximated within a factor of $\Omega(\log B)$, or within a factor of $\Omega(\log\log|V|)$, even for fixed $\Delta \geq 3$ and fixed $k \geq 3$. We also provide a $\frac{b_{\max}}{M}$-approximation algorithm for the general case, where $b_{\max} = \max_{i\in I} b(i)$ and M is the greatest common divisor among the bandwidths allowed by the interfaces and the required bandwidth B. Finally, we analyze the case of fixed $\Delta \leq 2$. As in the previous section, it has been shown that, in this case, *MCFMI* is polynomially solvable if k is fixed, then we focus on the approximability of the unbounded case. We give two results which show that, if $\Delta = 1$ or the input graph is a path, then *MCFMI* admits a FPTAS, while in the case that the input graph is a cycle the approximability of *MCFMI* remains open.

A. General case

Theorem 5.1: *MCFMI* cannot be approximated within a factor of $\Omega(\log B)$, or within a factor of $\Omega(\log\log|V|)$, for any fixed $\Delta \geq 3$ and $k \geq 3$, unless $P = NP$.

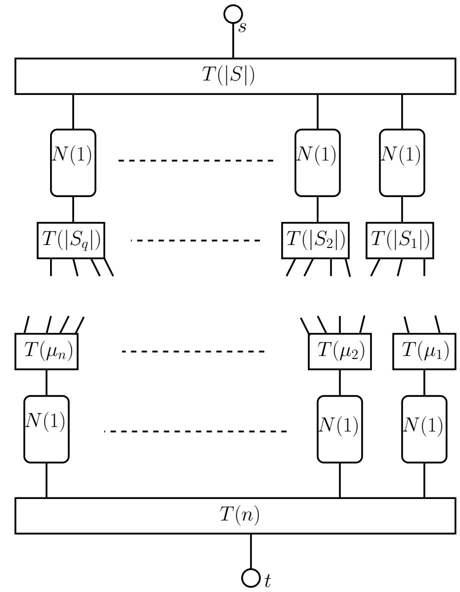
Proof: We will show the statement by providing an *approximation gap preserving reduction* [18] from the *Set Cover* (SC) problem to *MCFMI*.

SC: Set Cover	
***In*:**	A set U with n elements and a collection $S = \{S_1, S_2, \ldots, S_q\}$ of subsets of U.
***Sol*:**	A cover for U, i.e. a subset $S' \subseteq S$ such that every element of U belongs to at least one member of S'.
***Aim*:**	Minimize $|S'|$.

We first show that there exists a polynomial time algorithm that transforms any instance I_1 of SC into an instance I_2 of *MCFMI* such that the optimum value SOL^*_{SC} on I_1 for the problem SC is greater than or equal to the optimum value SOL^*_{MCFMI} on I_2 for the problem *MCFMI*.

The transformation is similar to the one provided for Theorem 4.2 (see Fig. 4). The graph G is given by two nodes s and t where s is adjacent to the root node of a copy of $T(|S|)$ and t to the root node of a copy of $T(n)$. For each element S_i of S, $i = 1, 2, \ldots, |S|$, G contains a copy of $N(1)$, denoted by $\overline{N}^i(1)$ and a subgraph $T(|S_i|)$ with root r^i adjacent to node $\overline{w}^i_1 \in \overline{N}^i(1)$. Each node $\overline{x}^i_1 \in \overline{N}^i(1)$ is adjacent to a different leaf of $T(|S|)$. Let u_j, $j = 1, 2, \ldots, |U|$, be the elements of U and let μ_j be the number of sets $S_i \in S$ containing u_j, for each j. For each element u_j, G contains a subgraph $T(\mu_j)$, with root r^j, and a copy $N^j(1)$ of $N(1)$, with nodes x^j_1, y^j_1 and w^j_1. Root r_j is adjacent to $x^j_1 \in N^j(1)$, and each node $x^j_1 \in N^j(1)$ is adjacent to a different leaf of $T(n)$, for each $j = 1, 2, \ldots, n$. If $u_j \in S_i$, for some i and j, then there is an

Fig. 4. The graph G in the transformation from $X3C$ to $MCFMI_D$.



edge from a leaf of $T(|S_i|)$ to a leaf of $T(\mu_j)$. These edges are pairwise disjoint.

The set of interfaces is $\{1, 2, 3\}$, with $c(1) = 0, c(2) = 1, c(3) = 0$ and $b(1) = n, b(2) = n, b(3) = 1$. All the nodes in G have interface 1 apart from the central nodes of the $n + |S|$ copies of the $N(1)$ graph. All the nodes in $\overline{N}^i(1)$ have interface 2, for each $i = 1, 2, \ldots, |S|$. All the nodes in $N^j(1)$ have interface 3, for each $j = 1, 2, \ldots, n$. Then,

$$X(u, v) = \begin{cases} W(u) \cap W(v) & \text{if } \{u, v\} \in E \\ \emptyset & \text{otherwise.} \end{cases}$$

Let $B = n$, we denote by $SOL_{SC}(I_1, \sigma_1)$ the value of the cost function of SC for a solution σ_1 on instance I_1, and let $SOL^*_{SC}(I_1)$ be the optimal cost for SC on instance I_1. Moreover, let us denote by $SOL_{MCFMI}(I_2, \sigma_2)$ the cost function of *MCFMI* of a solution σ_2 on instance I_2, and let $SOL^*_{MCFMI}(I_2)$ be the optimal cost for *MCFMI* on instance I_2.

Let us assume that we have an optimal solution $\{S_{i_1}, S_{i_2}, \ldots, S_{i_m}\}$ for SC. Then, by activating all the interfaces 1 and 3 in G and the interfaces 2 only in the nodes of subgraphs $\overline{N}^{i_j}(1)$, $j = 1, 2, \ldots, |S|$, we obtain a feasible solution σ for I_2 such that $SOL(I_2, \sigma) \leq 3SOL^*_{SC}(I_1)$, hence: $SOL^*_{MCFMI}(I_2) \leq 3SOL^*_{SC}(I_1)$.

Now we show that it is possible to transform in polynomial time any solution σ_2 for the instance I_2 of *MCFMI* into a solution σ_1 for the instance I_1 of SC such that $3SOL_{SC}(I_1, \sigma_1) = SOL_{MCFMI}(I_2, \sigma_2)$. A solution σ_2 consists of a flow of n units passing through each subgraph $N^j(1)$, $j = 1, 2, \ldots, n$, corresponding to a covering of all the elements in U. A solution σ_1 then consists of the sets $S_{i_1}, S_{i_2}, \ldots, S_{i_m}$ corresponding to those subgraphs $\overline{N}^{i_j}(1)$, $j = 1, 2, \ldots, m$, having a positive flow. As consequence, $3SOL_{SC}(I_1, \sigma_1) = SOL_{MCFMI}(I_2, \sigma_2)$.

If there exists an α factor approximation algorithm $\mathcal{A}$ for *MCFMI*, we would obtain an α factor approximation algorithm for SC. In fact, given an instance I_1 of SC, we could find a solution σ_1 for SC by using the above transformation from I_1 to an instance I_2 of *MCFMI* and applying $\mathcal{A}$ to find an α-approximate solution σ_2. Hence obtaining $SOL_{SC}(I_1,\sigma_1) = \frac{1}{3}SOL_{MCFMI}(I_2,\sigma_2) \leq \frac{\alpha}{3}SOL^*_{MCFMI}(I_2) \leq \alpha SOL^*_{SC}(I_1)$. In [19], the authors show that no approximation algorithm for SC exists with an approximation factor less than $\Omega(\log n)$. Then there is no algorithm for *MCFMI* with an approximation factor less than $\Omega(\log B)$, since we set $B = n$. By observing that for the instance I_2, $|V| \leq a_1|S| + a_2 n + a_3 \leq 2^n a_1 + a_2 n + a_3 \leq 2^n a_4$ for certain constants a_1, a_2, a_3 and $a_4 = 3\max\{a_1, a_2, a_3\}$, we have $n \geq \log(|V|/a_4)$. By using the same inapproximability result as before, we obtain the thesis. ■

Theorem 5.1 also holds when the number of interfaces is unbounded. We now provide a $\frac{b_{\max}}{M}$-approximation algorithm for any instance of *MCFMI*, where $b_{\max}$ is the maximum bandwidth value among the interfaces in I and M is the greatest common divisor among the bandwidths allowed by the interfaces and the required bandwidth B. The algorithm consists in relaxing *MCFMI* to the well-known *Integral Minimum Cost Flow* (*IMCF*) problem [20]. In the proof of the next theorem, we transform an instance of *MCFMI* into an instance of *IMCF*, and we show that such a transformation guarantees an approximation factor of $\frac{b_{\max}}{M}$. Let $\mathcal{A}$ be an algorithm which optimally solves *IMCF* on a graph $H = (V', E')$ in polynomial time $P_{\mathcal{A}}(|V'| + |E'|)$.

Theorem 5.2: There exists a polynomial time $\frac{b_{\max}}{M}$-approximation algorithm for *MCFMI* which requires $O(|V|k^2 + |E| + P_{\mathcal{A}}(|V|k^2 + |E|))$ time.

Proof: Given an instance I_1 of *MCFMI*, the algorithm works in four phases. First it transforms the graph G and functions b and c of I_1 into a graph G' and functions b' and c' as described in Section III. Hence, we obtain an instance I_2 of an equivalent problem defined on a directed graph $G' = (V', A)$ without using multiple interfaces but associating costs and bandwidths only to arcs in A. The aim of such problem is finding a flow function which satisfies flow constraints and such that the flow going from the source $\tilde{s}$ to the sink $\tilde{t}$ is greater than or equal to B. Then, the algorithm transforms I_2 into an instance I_3 of *IMCF*. In the third phase, the algorithm solves I_3 by using a known algorithm and, finally, it transforms the obtained solution for I_3 into a solution for I_2 made of a flow function f'. Function f' can be transformed into a solution for I_1, as described in Section III, obtaining a flow function f and an assignment of interfaces W_A.

In the following, we first show that the problems of solving I_1 and I_2 are equivalent, then we show how to approximate an optimal solution for I_2 by optimally solving I_3.

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 explained in Section III, that is,

$$f_1(u,v,i) = \begin{cases} f_2((\underline{u},i),(\overline{v},i)) - f_2((\underline{v},i),(\overline{u},i)) & \text{if } i \in X(u,v) \\ 0, & \text{otherwise.} \end{cases}$$

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 as follows, if $i \in X(u,v)$,

$$f'_2((\underline{u},i),(\overline{v},i)) = \begin{cases} f'_1(u,v,i) & \text{if } f'_1(u,v,i) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

In edges in $\{((\overline{v},i),(\underline{v},i)) \mid v \in V, i \in W(v)\} \cup \{((\underline{v},i),(\overline{v},j)) \mid v \in V, i,j \in W(v) \text{ s.t. } i \neq j\}$, f'_2 is defined in order to satisfy flow conservation constraints, and it is 0 for any other pair in $V \times V$. We now prove that the feasibility of f_2 (f'_1, resp.) implies the feasibility of f_1 (f'_2, resp.). By Lemma 3.1, properties 1–4 of the definition of *MCFMI* follows. Moreover, similar arguments can be used to show that f'_2 satisfies flow constraints and that

$$\begin{aligned} \sum_{v \in V'} f_2(\tilde{s},v) &= \sum_{v \in V, i \in I} f_1(s,v,i) \\ \sum_{v \in V, i \in I} f'_1(s,v,i) &= \sum_{v \in V'} f'_2(\tilde{s},v). \end{aligned}$$

Hence, to show property 5 of the definition of *MCFMI*, it is enough to note that if $\sum_{v \in V'} f_2(\tilde{s},v) \geq B$ ($\sum_{v \in V, i \in I} f'_1(s,v,i) \geq B$, resp.) then $\sum_{v \in V, i \in I} f_1(s,v,i) \geq B$ ($\sum_{v \in V'} f'_2(\tilde{s},v) \geq B$, resp.). This shows that the feasibility of f_2 (f'_1, resp.) implies the feasibility of f_1 (f'_2, resp.). To conclude the first part of the proof, note that, the cost of f_2 (f'_1, resp.) is equal to the cost of f_1 (f'_2, resp.) as the cost of arcs $((\overline{v},i),(\underline{v},i))$ 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 Θ 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., [21]).

We obtain an *IMCF* instance I_3 from I_2 by setting $H = G'$, $\Theta = B/M$ $\beta(a) = b'(a)/M$, and $\chi(a) = c'(a)/b'(a)$, for each $a \in A$. Given a feasible flow function f_3 for I_3, a flow function f_2 for I_2 is obtained by multiplying f_3 by M, that is $f_2(a) = M \cdot f_3(a)$, for each $a \in A$. The feasibility of f_3 for I_3 clearly implies the feasibility of f_2 for I_2.

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, $\text{OPT} = \sum_{a \in A^*} c'(a)$. As $f^*(a) \leq b'(a)$, it follows that

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

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

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

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, $b_{\max} \geq b'(a)$, for each $a \in A^{IMCF}$.

Therefore,

$$M \cdot \sum_{a \in A^{IMCF}} \frac{c'(a)}{b'(a)} \cdot f^{IMCF}(a) \geq \frac{M}{b_{\max}} \cdot \sum_{a \in A^{IMCF}} c'(a).$$

The computational time required by the algorithm defined in this proof is given by the cost of transforming I_1 into I_2, that of transforming I_2 into I_3, and that of solving I_3. As the graph defined for I_2 has $O(|V|k)$ nodes and $O(|V|k^2 + |E|)$ edges, the first and the second costs are $O(|V|k^2 + |E|)$ while the third one is $O(P_{\mathcal{A}}(|V|k^2 + |E|))$. ■

Corollary 5.3: Let $b \in \mathbb{Z}_0^+$. If $b(i) = b$ for each $i \in I$, *MCFMI* is solvable within $O(|V|k^2 + |E| + P_{\mathcal{A}}(|V|k^2 + |E|))$.

Proof: If $b = 1$, then the $\frac{b_{\max}}{M}$-approximation algorithm given in Theorem 5.2 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. The computational time required by this algorithm is equal to that required by the algorithm defined in Theorem 5.2 ■

B. Particular cases, $\Delta \leq 2$

We now analyze some special cases where the approximation bound can be improved. In the previous section, it has been shown that when $\Delta \leq 2$, *MCFMI* is *NP*-hard in the unbounded case and it is polynomially solvable in the bounded case. Theorem 5.1 shows that even for fixed $\Delta \geq 3$ *MCFMI* is not approximable within a constant approximation bound. Hence, we focus on the approximation for the unbounded case where $\Delta \leq 2$. We give two results which show that, in the unbounded case, if $\Delta = 1$ or the input graph is a path, then *MCFMI* admits a FPTAS, while in the case that the input graph is a cycle the approximability of *MCFMI* remains open.

The following corollary gives an FPTAS in the case that $\Delta = 1$ and it follows from Theorem 4.4.

Corollary 5.4: In the unbounded case with $\Delta = 1$, *MCFMI* admits a $(1 + \epsilon)$-approximation algorithm which requires $O(\frac{k^2}{\epsilon})$ time, for any $\epsilon > 0$.

Proof: It follows by applying the linear time algorithm $\mathcal{A}$ of Theorem 4.4 which requires $O(k)$ time, and the algorithm from [22] which provides a $(1+\epsilon)$-approximation for $MinKP$ in $O(\frac{k^2}{\epsilon})$ time. ■

The following theorem gives an FPTAS in the case that the input graph is a path.

Theorem 5.5: In the unbounded case, if the input graph is a path, *MCFMI* admits a $(2+\epsilon)$-approximation algorithm which requires $O(|V|\frac{k^2}{\epsilon})$ time, for any $\epsilon > 0$.

Proof: Let us denote the input path as a sequence of n nodes: $s \equiv x_0, x_1, \ldots, x_{n-1} \equiv t$. We define an algorithm $\mathcal{C}$ as follows. It defines $n - 1$ $MinKP$ problems, each one arising from one different edge $e_i = \{x_{i-1}, x_i\}$ of the path, $1 \leq i \leq n - 1$, by using the linear time algorithm $\mathcal{A}$ of Theorem 4.4. From Corollary 5.4, this implies that for each e_i and for any $\epsilon > 0$, a $(1 + \epsilon)$-approximation for $MinKP$ can be guaranteed. Algorithm $\mathcal{C}$ chooses, for each $1 \leq i \leq n - 1$, interfaces W_i arising from the approximate solution of the related knapsack problem on edge e_i, that is interfaces W_i are activated on nodes x_{i-1} and x_i.

For each $1 \leq i \leq n-1$, let us denote as $W_i^* \subseteq X(x_{i-1}, x_i)$, the sets of active interfaces in nodes x_{i-1} and x_i in an optimal solution of *MCFMI* for the input path; and let $W_i^{MK} \subseteq X(x_{i-1}, x_i)$ the sets of active interfaces in nodes x_{i-1} and x_i in an optimal solution of the $MinKP$ problem obtained by $\mathcal{C}$ for the input path.

Note that, for some i, the set $W_i \cap W_{i+1}$ is not necessarily empty, which means that node x_i uses a set of interfaces for communicating with both x_{i-1} and x_{i+1}. Thus, in this case, the cost paid for activating the interfaces used by x_i is less than $c(W_i) + c(W_{i+1})$ and the same holds for solutions W_i^* and W_i^{MK}. It follows that, for each $1 \leq i \leq n - 1$ the cost paid for activating interfaces in W_i in nodes x_i and x_{i-1} is at most $2c(W_i)$ and the overall cost of the solution provided by $\mathcal{C}$ is less than or equal to $2\sum_{i=1}^{n-1} c(W_i)$. As from Corollary 5.4 we are using in each edge a $(1 + \epsilon)$-approximation algorithm for the knapsack problem, it follows that: $2\sum_{i=1}^{n-1} c(W_i) \leq 2\sum_{i=1}^{n-1}(1 + \epsilon)c(W_i^{MK})$. As W_i^{MK} is an optimal solution for $MinKP$ on edge e_i which guarantees a bandwidth of B, $c(W_i^{MK}) \leq c(W_i^*)$, for each $1 \leq i \leq n - 1$, and hence: $2(1 + \epsilon)\sum_{i=1}^{n-1} c(W_i^{MK}) \leq 2(1 + \epsilon)\sum_{i=1}^{n-1} c(W_i^*) \leq 2(1+\epsilon)\left(\sum_{i=1}^{n-2} c(W_i^* \cup W_{i+1}^*) + c(W_{n-1}^*)\right) \leq 2(1+\epsilon)\,\text{OPT}$, where the two last inequalities follow from the fact that in an optimal solution the cost of activating interfaces for each node x_i is $c(W_i^* \cup W_{i+1}^*) \geq c(W_i^*)$ and the overall cost is $\text{OPT} = c(W_1^*) + \sum_{i=1}^{n-2} c(W_i^* \cup W_{i+1}^*) + c(W_{n-1}^*)$.

The complexity of $\mathcal{C}$ is $O(n\frac{k^2}{\epsilon})$ as it is composed of $n - 1$ executions of algorithm $\mathcal{A}$ of Theorem 4.4 which requires $O(k)$ time, and $n - 1$ executions of algorithm from [22] which requires $O(\frac{k^2}{\epsilon})$ time. By defining $\epsilon' = 2\epsilon$, Algorithm $\mathcal{C}$ provides a $(2 + \epsilon')$-approximated solution and requires $O(|V|\frac{k^2}{\epsilon'})$ time. ■

The FPTAS provided directly implies the existence of a pseudo-polynomial-time algorithm for the case where the input graph is a path. This implies that, in this case, the problem is not *NP*-hard in the strong sense.

VI. Experimental Analysis

In this Section, we report the results of our experimental study on the approximation algorithm given in Theorem 5.2 which is denoted by ALG in the remainder of the 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 by using the LEMON Graph Library [23] framework. In order to solve the *IMCF* instances required by ALG we used the Network Simplex algorithm [24] provided by LEMON as it is the most experimentally efficient in general cases.

A. Input data and executed tests.

Instances of *MCFMI* have been randomly generated by using two different models: The *balls-into-bins* [25], [26] and the *Barabási-Albert* power-law [27] models.

The balls-into-bins model is used to simulate devices thrown at random in a two-dimensional space [25]. In this model, each instance of *MCFMI* is made of a graph $G_{BIB} = (V_{BIB}, E_{BIB})$, a set of interfaces $I_{BIB} = \{1, 2, \ldots, k\}$ along with cost and bandwidth functions c_{BIB}, and b_{BIB}, and two allocation functions $W_{BIB} : V_{BIB} \to 2^{I_{BIB}}$ and $X_{BIB} : V_{BIB} \times V_{BIB} \to 2^{I_{BIB}}$. First, nodes in V_{BIB} are generated and a uniformly random position in a unit size square is associated to each of them. From the "balls-into-bins" theory [26], 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.

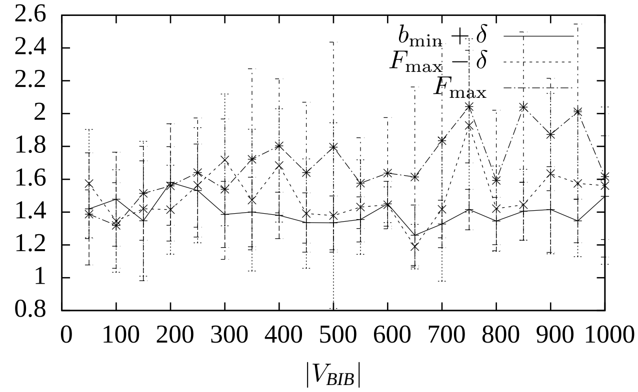
Barabási–Albert networks have been proven to model many real-world networks such as the Internet, the World Wide Web, citation graphs, and some social networks [28]. A Barabási–Albert topology is generated by iteratively adding one node at a time, starting from a given connected graph with at least two 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 $I_{BA} = \{1, 2, \ldots, k\}$ along with cost and bandwidth functions c_{BA}, and b_{BA}, and two allocation functions $W_{BA} : V_{BA} \to 2^{I_{BA}}$ and $X_{BA} : V_{BA} \times V_{BA} \to 2^{I_{BA}}$. 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.

TABLE II
SIZE OF THE INPUT DATA.

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

Fig. 5. Graphs G_{BIB}: average upper bounds on the approximation ratios for $|V_{BIB}| \in \{50, 100, \ldots, 1000\}$, $k = 9$ and three values of required flow.



For each edge $\{u, v\} \in E_{BA}$, interface $i \in I_{BA}$ is added to $X_{BA}(u, v)$ with probability 0.5. For each node v, $W_{BA}(v)$ is induced by the interfaces associated in $X_{BA}(u, v)$ for each edge $\{u, v\}$ 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 Theorem 4.1. 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 cheapest path (see [11]) connecting source and destination. When the considered instance is clear by the context, we denote $\frac{F_{\max} - b_{\min}}{3}$ simply by δ.

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*. 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 5.3.

Fig. 6. Graphs G_{BIB}: average upper bounds on the approximation ratios for $|V_{BIB}| \in \{50, 100, \ldots, 1000\}$, $k = 3$ and three values of required flow.

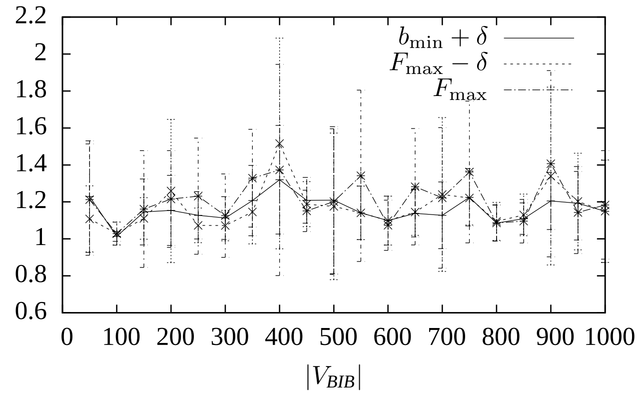


Fig. 7. Graphs G_{BIB}: average upper bounds on the approximation ratios for $|V_{BIB}| = 10000$, $k \in \{2, 4, \ldots, 16\}$ and three values of required flow.

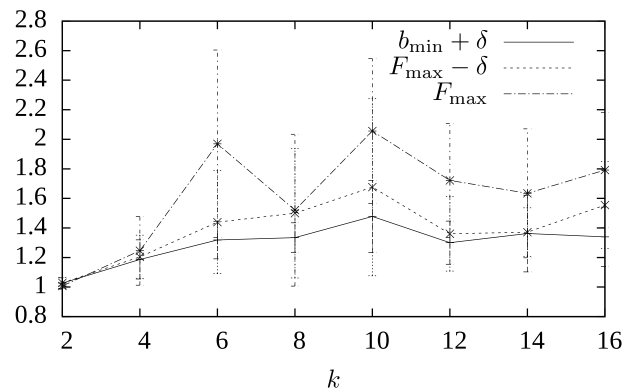


Table II reports the size of the input data used in the experiments. We perform two kind of experiments: we fix k to 3, 6, and 9 and we let vary the number of nodes in the graphs from 50 to 1000; we fix the number of nodes in a graph to 10, 100, 1000, and 10000 and let k vary from 2 to 16. In each setting,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

B. Analysis of experimental results

The results of our experiments are reported in Fig.s 5–11 and in Table III. For a better visualization, all the obtained values are normalized to $|V_{BIB}|$ for the experiments referring to graphs G_{BIB}, and to $|V_{BA}|$ for the experiments referring to graphs G_{BA}. This is equivalent to consider each instance graph $G = (V, E)$ inside a $|V| \times |V|$ 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 $\{b_{min}, \ldots, F_{\max}\}$. Namely, the curves refer to $b_{\min} + \delta$, $F_{\max} - \delta$ and $F_{\max}$, as for $b_{\min}$ we can compute the optimal value.

Fig. 5 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 3.12, achieved by an instance of 350 nodes and 5229 edges, when the required flow is $F_{\max}$. However, there are very few instances with an upper bound on the approximation ratio in $[3, 4)$. In detail, for 3 instances it is in $[3, 4)$, for 71 instances it is in $[2, 3)$, for 507 instances it is in $(1, 2)$ and for all the other 19 instances it ensures the optimal value. On average, the upper bound is always smaller than 2.04. Moreover, we remind that these are only upper bounds to the real ratio. 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 upper bound on 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 $\frac{b_{\max}}{M}$ guaranteed by the theoretical analysis of Theorem 5.2. The value $\frac{b_{\max}}{M}$ can be in fact very much higher than the experimented results. For instance, networks providing Fig. 5 shows an average value for $b_{\max}$ larger than 10.000. This confirms the interest in studying the algorithm for practical instances in order to better understand its real performances. Fig. 6 shows the three curves when $k = 3$ and the other parameters are in the same setting as Fig. 5. As expected, the upper bound on the approximation ratio is improved here. This is due to the fact that reducing the number of interfaces implies that the possible overhead at each node is also reduced. In detail, the maximum upper bound on the approximation ratio obtained is 2.71, achieved by an instance of 400 nodes and 5311 edges, when the required flow is $F_{\max}$. The upper bound to the approximation ratio is in $[2, 3)$ for 16 instances, in $(1, 2)$ for 382 instances and the algorithm finds the optimum for the remaining 202 instances.

Fig. 7 refers to the case where $|V_{BIB}|$ is fixed to 10000, and the number of interfaces k ranges from 2 to 16. Also in this case, the upper bound on the approximation ratio is very small. In detail, in the worst case it achieves 3.03. 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, the upper bound on the approximation ratio slightly increases with the required flow.

We can conclude that, in graphs G_{BIB}, the approximation ratio is always very small and it depends neither on the number of nodes nor on the number of interfaces, while there is a small dependency from the required flow.

Fig.s 8 and 9 show the experimental results in the same settings as Fig.s 5 and 7 for graphs G_{BA}. Also in these cases, the properties inferred for G_{BIB} hold. In fact, the upper bound on 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, Fig. 8 shows a case where the average

Fig. 8. Graphs G_{BA}: average upper bounds on the approximation ratios for $|V_{BA}| \in \{50, 100, \ldots, 1000\}$, $k = 9$ and three values of required flow.

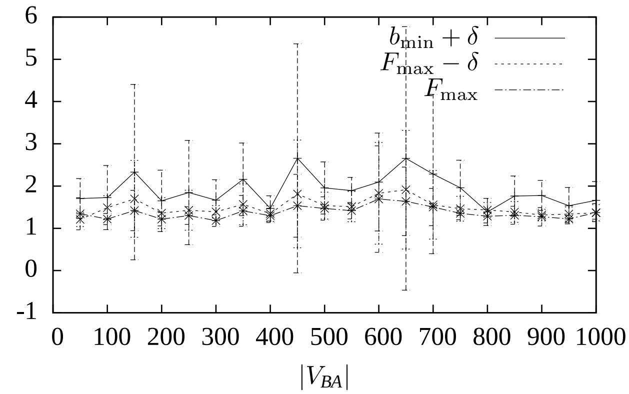


Fig. 9. Graphs G_{BA}: average upper bounds on the approximation ratios for $|V_{BA}| = 10000$, $k \in \{2, 4, \ldots, 16\}$ and three values of required flow.

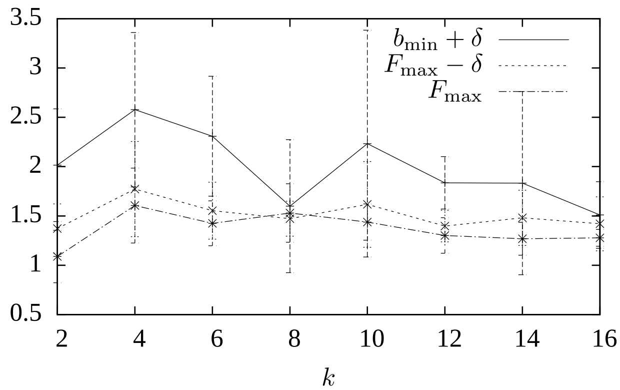


Fig. 10. Graphs G_{BA} when $\frac{b_{max}}{b_{min}}$ is bounded: average upper bounds on the approximation ratios in the same setting as Fig. 8.

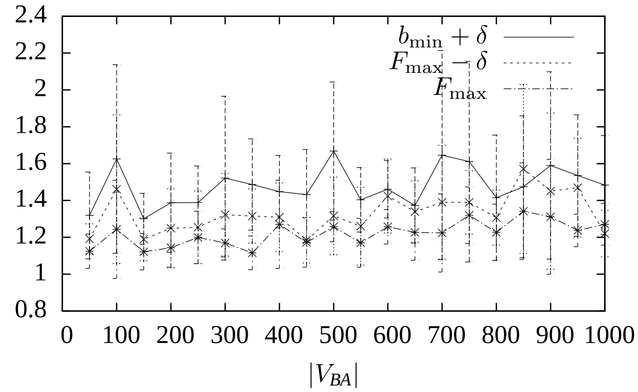
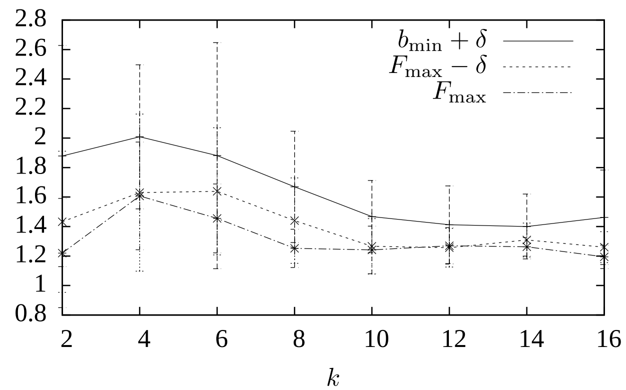


Fig. 11. Graphs G_{BA} when $\frac{b_{max}}{b_{min}}$ is bounded: average upper bounds on the approximation ratios in the same setting as Fig. 9.



value is 2.65 and the standard deviation is 3.12 which is due to an instance where the upper bound on the approximation ratio is 11.80. Similar instances also appear in the other experiments on G_{BA}. 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 to the behavior of ALG. In fact, it is known that graphs G_{BA} have a small diameter [29] and hence both relaxations used for obtaining the lower bounds of the optimal value give a rather small value. In order to obtain a better estimation of the lower bound in graphs G_{BA}, we performed a new set of tests with the same parameters as those of Fig.s 8–9 but in instances where the ratio between the maximal and the minimal bandwidth of the involved interfaces is upper bounded. These particular instances have a practical relevance because, in real cases, it is reasonable that the mentioned ratio is upper bounded. Such instances allow us to better estimate the optimal value because in such cases the two relaxations used can give a tight lower bound. Results for the case where the ratio between the maximal and the minimal bandwidth of an interface is at most 10 are given in Fig.s 10–11. Note that the values are similar to those of G_{BIB} graphs.

Concerning the execution time of the algorithm, it goes from few microseconds in the smaller instances to some seconds in large instances made of 10000 nodes and 16 interfaces (see Table III). Hence, the algorithm is fast enough to be used in large scale networks.

VII. 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 establish the maximal bandwidth that can be guaranteed between two given nodes of the input network. 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. Polynomial exact and approximation algorithms for the general case and for special cases of *MCFMI* have been provided. Moreover, we experimentally analyzed algorithm ALG for *MCFMI*, showing that in practical cases it guarantees a low approximation ratio which allows us to use it in real-world.

References

[1] G. D'Angelo, G. Di Stefano, and A. Navarra, "Bandwidth Constrained Multi-Interface Networks," in *Proc. of the 37th Int.l Conf. on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, ser. Lecture Notes in Computer Science, vol. 6543. Springer, 2011, pp. 202–213.

TABLE III
AVERAGE COMPUTATIONAL TIME.

Graph	k	$\|V\|$	time (ms)
Balls-into-bins	2	10	0.05
		10000	1501.69
	16	10	0.37
		10000	26105.70
Barabási–Albert	2	10	0.04
		10000	68.15
	16	10	0.53
		10000	5278.01

[2] ——, “Maximum Flow and Minimum-Cost Flow in Multi-Interface Networks,” in *Proc. of the 5th ACM Int.l Conf. on Ubiquitous Information Management and Communication (ICUIMC)*. ACM, 2011, pp. 19:1–19:9.

[3] P. Bahl, A. Adya, J. Padhye, and A. Walman, “Reconsidering wireless systems with multiple radios,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 5, pp. 39–46, 2004.

[4] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *10th Int.l Conference on Mobile computing and networking (MobiCom)*. ACM, 2004, pp. 114–128.

[5] G. DAngelo, G. Di Stefano, and A. Navarra, “Minimize the maximum duty in multi-interface networks,” *Algorithmica*, vol. 63, no. 1-2, pp. 274–295, 2012.

[6] D. Cavalcanti, H. Gossain, and D. Agrawal, “Connectivity in multi-radio, multi-channel heterogeneous ad hoc networks,” in *IEEE 16th Int.l Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2005, pp. 1322–1326.

[7] A. Faragó and S. Basagni, “The effect of multi-radio nodes on network connectivity—a graph theoretic analysis,” in *IEEE Int.l Workshop on Wireless Distributed Networks (WDM)*. IEEE, 2008.

[8] M. Caporuscio, D. Charlet, V. Issarny, and A. Navarra, “Energetic Performance of Service-oriented Multi-radio Networks: Issues and Perspectives.” in *Proc. of the 6th Int.l Workshop on Software and Performance (WOSP)*. ACM Press, 2007, pp. 42–45.

[9] R. Klasing, A. Kosowski, and A. Navarra, “Cost Minimization in Wireless Networks with a Bounded and Unbounded Number of Interfaces,” *Networks*, vol. 53, no. 3, pp. 266–275, 2009.

[10] S. Athanassopoulos, I. Caragiannis, C. Kaklamanis, and E. Papaioannou, “Energy-efficient communication in multi-interface wireless networks,” in *34th Int.l Symp. on Mathematical Foundations of Computer Science (MFCS)*, ser. LNCS, vol. 5743. Springer, 2009, pp. 102–111.

[11] A. Kosowski, A. Navarra, and M. Pinotti, “Exploiting Multi-Interface Networks: Connectivity and Cheapest Paths,” *Wireless Networks*, vol. 16, no. 4, pp. 1063–1073, 2010.

[12] V. Bhandari and N. H. Vaidya, “Scheduling in Multi-Channel Wireless Networks,” in *Proc. of the 11th Int.l Conf. on Distributed Computing and Networking (ICDCN)*, ser. Lecture Notes in Computer Science, vol. 5935. Springer, 2010, pp. 6–17.

[13] X. Lin and S. B. Rasool, “Distributed and provably efficient algorithms for joint channel-assignment, scheduling, and routing in multichannel ad hoc wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1874–1887, 2009.

[14] P.-J. Wan, Y. Cheng, Z. Wang, and F. Yao, “Multiflows in multi-channel multi-radio multihop wireless networks,” in *Proc. of the 30th IEEE Int.l Conf. on Computer Communications (INFOCOM)*. IEEE, 2011, pp. 846 – 854.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman and Company, 1979.

[16] S. Görtz and A. Klose, “Analysis of some greedy algorithms for the single-sink fixed-charge transportation problem,” *Journal of Heuristics*, vol. 15, no. 4, pp. 331–349, 2009.

[17] M. Güntzer and D. Jungnickel, “Approximate minimization algorithms for the 0/1 knapsack and subset-sum problem,” *Operations Research Letters*, vol. 26, no. 2, pp. 55–66, 2000.

[18] V. V. Vazirani, *Approximation Algorithms*. Springer, 2001.

[19] N. Alon, D. Moshkovitz, and S. Safra, “Algorithmic construction of sets for k-restrictions,” *ACM Transactions on Algorithms*, vol. 2, no. 2, pp. 153–177, 2006.

[20] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.

[21] E. Lawer, *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, and Winston, 1976.

[22] G. Gens and E. Levner, “Computational complexity of approximation algorithms for combinatorial problems,” in *8th Int.l Symp. on Mathematical Foundations of Computer Science (MFCS)*, ser. LNCS, vol. 74. Springer, 1979, pp. 292–300.

[23] “LEMON graph library,” `http://lemon.cs.elte.hu/trac/lemon/`.

[24] G. B. Dantzig, *Linear Programming and Extensions*. Princeton University, 1963.

[25] A. Papadopoulos, J. McCann, and A. Navarra, “Connectionless probabilistic (cop) routing: an efficient protocol for mobile wireless ad-hoc sensor networks,” in *24th IEEE Int.l Performance, Computing, and Communications Conference, IPCCC*. IEEE, 2005, pp. 73–77.

[26] M. Raab and A. Steger, ““balls into bins” – a simple and tight analysis,” in *2nd Int.l Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM’98*, ser. LNCS, vol. 1518. Springer, 1998, pp. 159–170.

[27] R. Albert and A.-L. Barabási, “Emergence of scaling in random networks,” *Science*, vol. 286, pp. 509–512, 1999.

[28] ——, “Statistical mechanics of complex network,” *Reviews of Modern Physics*, vol. 74, pp. 47–97, 2002.

[29] R. Cohen and S. Havlin, “Scale-free networks are ultrasmall,” *Physical Review Letters*, vol. 90, no. 5, p. 058701, 2003.

Gianlorenzo D’Angelo Dr. Gianlorenzo D’Angelo obtained his B.Sc., M.Sc. and Ph.D. from the University of L’Aquila, Italy in 2004, 2006 and 2010, respectively. He is currently a Post-doc at the MASCOTTE project of the INRIA institute of Sophia Antipolis, France. His research interests are in the area of combinatorial optimization, robust optimization, networking and distributed computing.

Gabriele Di Stefano Prof. Gabriele Di Stefano obtained his Ph.D. at University “La Sapienza” of Rome in 1992. Currently he is associate professor for computer science at the University of L’Aquila; his current research interests include network algorithms, combinatorial optimization, algorithmic graph theory; he is (co-)author of more than 60 publications in journals and international conferences. He had key-participation in several EU funded projects. Among them: MILORD (AIM 2024), COLUMBUS (IST 2001-38314), AMORE (HPRN-CT-1999-00104), and ARRIVAL (IST FP6-021235-2).

Alfredo Navarra Dr. Alfredo Navarra is Assistant Professor at the University of Perugia, Italy. He received his Ph.D. in Computer Science at “Sapienza” University of Rome in 2004. From 2003 to 2007 he as moved among various international research centers like: the INRIA institute of Sophia Antipolis, France; the Dept of Computer Science at the Univ. of L’Aquila, Italy; the LaBRI, Univ. of Bordeaux, France; the Dept of Electrical and Information Engineering at Univ. of L’Aquila, Italy. His research interests include algorithms, computational complexity, networking and distributed computing.