Minimize the Maximum Duty in Multi-interface Networks

Gianlorenzo D'Angelo, Gabriele Di Stefano, Alfredo Navarra

Minimize the Maximum Duty in Multi-Interface Networks

Gianlorenzo D'Angelo ·
Gabriele Di Stefano · **Alfredo Navarra**

Abstract We consider devices equipped with multiple wired or wireless interfaces. By switching of various 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. In this paper, we consider two basic networking problems in the field of multi-interface networks. The first one, known as the Coverage problem, requires to establish the connections defined by a network. The second one, known as Connectivity problem, requires to guarantee a connecting path between any pair of nodes of a network. Both are subject to the constraint of keeping as low as possible the maximum cost set of active interfaces at each single node. We study the problems of minimizing the maximum cost set of active interfaces among the nodes of the network in order to cover all the edges in the first case, or to ensure connectivity in the second case. We prove that the Coverage problem is NP-hard for any fixed $\Delta \geq 5$ and $k \geq 16$, with Δ being the maximum degree, and k being the number of different interfaces among the network. We also show that, unless $P = NP$, the problem cannot be approximated within a factor of $\eta \ln \Delta$, for a certain constant η. We then provide a general approximation algorithm which guarantees a factor of $O((1 + b) \ln \Delta)$, with b being a parameter depending on the topology

G. D'Angelo and G. Di Stefano
Department of Electrical and Information Engineering,
University of L'Aquila, Via Gronchi 18, I–67100, L'Aquila – Italy.
E-mail: {gianlorenzo.dangelo, gabriele.distefano}@univaq.it

A. Navarra
Department of Mathematics and Computer Science,
University of Perugia, Via Vanvitelli 1, I–06123, Perugia – Italy.
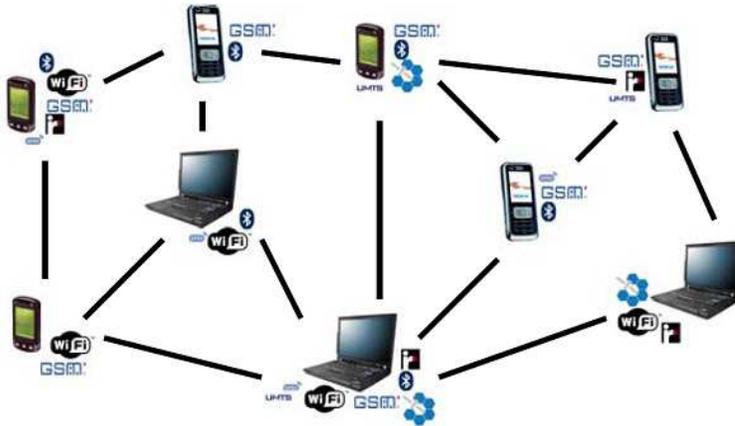E-mail: navarra@dmi.unipg.it

Fig. 1 The composed network according to available interfaces and proximities.

of the input graph. Interestingly, b can be bounded by a constant for many graph classes. Other approximation and exact algorithms for special cases are presented. Concerning the Connectivity problem, we prove that it is *NP*-hard for any fixed $\Delta \geq 3$ and $k \geq 10$. Also for this problem, the inapproximability result holds, that is, unless $P = NP$, the problem cannot be approximated within a factor of $\eta \ln \Delta$, for a certain constant η. We then provide approximation and exact algorithms for the general problem and for special cases, respectively.

1 Introduction

Wireless networks certainly provide intriguing problems for the scientific community due to the wide range of real-world applications. A very important issue recently addressed is constituted by the heterogeneity of the devices. Different computational power, energy consumption, radio interfaces, supported communication protocols, and other peculiarities can characterize the involved devices. In this paper, we are mainly interested in devices equipped with multiple interfaces. An example of a network instance is shown in Figure 1, where mobile phones, smart-phones and laptops can communicate by means of different interfaces and protocols such as IRdA, Bluetooth, WiFi, GSM, Edge, UMTS and Satellite. All the possible connections can be covered by means of at least one interface. Note that, some devices are not directly connected even though they share some interfaces. This can be due to many factors like for instance obstacles or distances.

A connection between two or more devices might be accomplished by means of different communication networks according to provided 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 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 a set of connections between devices, defined according to their proximity and the available interfaces that they may share. Each $v \in V$ is associated with a set of available interfaces $W(v)$. 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 *covered* when the endpoints of the corresponding edge share at least one active interface. If an interface i is activated at some node u, then u consumes some energy $c(i)$ for maintaining i as active. In this setting, we study two basic problems. The first one concerns the covering of all the edges of G by minimizing the maximum cost required at the single nodes. The second one concerns the covering of a connected spanning subgraph of G still by minimizing the maximum cost required at the single nodes. In other words, in the second problem, we look for the set of active interfaces among V, in such a way that for each pair of nodes $u, v \in V$ there exists a path of covered edges leading from u to v such that the maximum cost required for a single node is minimized. This implies that the cost provided by all the interfaces activated in the whole network to accomplish either the coverage or the connectivity tasks might not be the global minimum. Indeed, the chosen requirement is in favor of a uniform energy consumption among the devices, as it tries to maintain as low as possible the maximum cost spent by the single devices. This plays a central role in the context of wireless networks where the whole network survivability might depend on few devices. The two problems of coverage and connectivity coincide whenever the input graph is a tree and hence it is required to activate all the edges in order to accomplish to the connectivity issue.

1.1 Related work

Multi-interface wireless networks have been recently studied in a variety of contexts, usually focusing on the benefits of multiple radio devices of each node [10, 15, 16]. Many basic problems of standard wireless network optimization can be reconsidered in such a setting [4]. However, previous works have been mainly focused on the minimization of the costs among the whole network. In [9, 21], for instance, the problem of *Coverage* has been investigated, but with the goal of activating the minimum cost set of interfaces among all the network in such a way that all the edges of G are covered. *Connectivity* issues have been addressed in [3, 21, 22]. The goal is again to activate the minimum cost set of interfaces in G in order to guarantee a path of communication between every pair of nodes. Experimental evaluations of

connectivity properties obtained by exploiting multi-interface networks can be found in [6]. In [5, 22], 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. Recently, [13] addresses flow problems. Each interface is associated with a maximum bandwidth and the goal is to activate the minimum cost set of interfaces in order to activate a set of communication paths between a fixed pair of source and target nodes that guarantee a required bandwidth threshold.

1.2 Our results

In this paper, we study the coverage and the connectivity problems with respect to a new objective function. The coverage problem asks for establishing all the connections defined by G which minimize the maximum cost required at the single nodes. We call this problem the *Minimum Maximum-Cost Coverage problem in Multi-Interface Networks* (*MMCov* for short). The connectivity problem asks for establishing a connected spanning subgraph of G by minimizing the maximum cost required at the single nodes. We call this problem the *Minimum Maximum-Cost Connectivity problem in Multi-Interface Networks* (*MMCon* for short). The chosen requirements represent a first step toward distributed environments where the objective function refers to local properties rather than global costs.

We consider two variants of the above problems: the parameter k is either considered as part of the input (this is called the *unbounded case*), or k is a fixed constant (the *bounded case*). The case where the cost function is constant for each interface is called the *unit cost* case. In the paper, we always assume that $k \geq 2$, since all the problems arising for $k = 1$ admit the obvious solution provided by activating the unique interface at all the nodes.

For *MMCov*, we first prove that the problem is *NP*-hard, even for the unit cost case and even when the number of interfaces k and the maximum node degree Δ are fixed. In particular, we prove that the problem remains *NP*-hard for any fixed $\Delta \geq 5$ and $k \geq 16$. Then, we present efficient algorithms that optimally solve the problem in some relevant special cases. In detail, we focus on instances where the input graph is a tree, by giving a polynomial time algorithm for fixed k or fixed Δ. By using this algorithm we can derive efficient algorithms for $\Delta \leq 2$. Furthermore, for the unit cost case, we give a polynomial time algorithm for $k \leq 3$. For fixed k, $3 \leq k \leq 15$, and fixed Δ, $3 \leq \Delta \leq 4$, the complexity of *MMCov* remains open.

Concerning approximation results for *MMCov*, we show that the problem is not approximable within an $\eta \ln(\Delta)$ factor for a certain constant η, unless $P = NP$. This result holds even in the unit cost case and when the input graph is a tree but only when k or Δ are unbounded. We then provide an approximation algorithm that guarantees a factor of $\ln(\Delta) + 1 + b \cdot \min\{c_{max}, (\ln(\Delta) + 1)\}$, with $c_{max} = \max_{i \in \{1, \dots k\}} c(i)$ and b being a parameter depending on structural properties of the input graph. Such parameter can be bounded by a constant

in many graph classes (see Section 4). Note that, the obtained approximation guarantees a $1 + b$ factor from the best possible algorithm. We also provide two simple approximation algorithms which guarantee $1 + (k - 2)\frac{c_{max}}{2c_{min}}$ and $\frac{c_{max}}{c_{min}}\frac{\Delta}{2}$ factors of approximation, where c_{min} and c_{max} are the minimum and the maximum cost associated with an interface, respectively. For the unit cost case this leads to a $\frac{k}{2}$-approximation factor. This clearly might be useful for small values of k.

For *MMCon*, we first prove that the problem is *NP*-hard, even for the unit cost case and even when the number of interfaces k and the maximum node degree Δ are fixed. In particular, we prove that the problem remains *NP*-hard for any fixed $\Delta \geq 3$ and $k \geq 10$. Then, we present efficient algorithms that optimally solve the problem in some relevant special cases. In detail, results concerning instances where the input graph is a tree are solved by the algorithm proposed for *MMCov* when k or Δ are fixed parameters. By using this algorithm we can derive polynomial time algorithms for graphs with $\Delta \leq 2$ and for polynomially recognizable Hamiltonian graphs. Furthermore, we give a polynomial time algorithm for $k \leq 2$. In summary, *MMCon* is *NP*-hard for any fixed $\Delta \geq 3$, while it is polynomially solvable for $\Delta \leq 2$. Moreover, it is *NP*-hard for any fixed $k \geq 10$ while it is polynomially solvable for $k \leq 2$. For fixed k, $3 \leq k \leq 9$, the complexity of *MMCon* remains open.

Concerning approximation results, also for *MMCon* holds the inapproximability within an $\eta \ln(\Delta)$ factor as it was for *MMCov* since such a result has been obtained in the unit cost case when the input graph is a tree. Hence, we give some simple approximation algorithms which guarantee a factor of approximation of $1 + (k-2)\frac{c_{max}}{2c_{min}}$ or $\frac{c_{max}}{c_{min}}\frac{\Delta}{2}$. In the unit cost case, they guarantee factors of $\frac{k}{2}$- and $\frac{\Delta}{2}$-approximation factors. When $k = O(1)$ or $\Delta = O(1)$ these algorithms achieve an $O(1)$-approximation factor. Approximation results obtained for *MMCov* on trees hold for *MMCon* as well. In fact, as for trees $b = 1$, we can derive an $2(\ln(\Delta) + 1)$-approximation algorithm for the non-unit case and a $\ln(\Delta) + 2$-approximation algorithm for the unit cost case. Note that, the obtained approximation factor for non-unit cost trees is 2 times the best possible one, while in the unit cost case we only have an addend of 2 far from the best possible factor.

1.3 Structure of the paper

In the next section, we formally define the addressed problems. The section also provides some preliminary results. In Section 3, we study the complexity of *MMCov* by analyzing the cases where the problem is *NP*-hard, and when the problem can be optimally solved. In Section 4, we provide inapproximability results and we present some polynomial time approximation algorithms for both the general case and particular cases. In Section 5, we study the complexity of *MMCon* by analyzing the cases where the problem is *NP*-hard and the cases where it is polynomially solvable. Moreover, we study the approximability of *MMCon* by giving a lower bound to the best approximation factor

achievable and by giving polynomial time algorithms that match this bound in some cases. Finally, in Section 6 we outline some conclusion and possible future research.

2 Preliminaries and Notation

For a graph G, we denote by V its node set, by E its edge set. We denote the sizes of V and E by n and m, respectively. For any $v \in V$, let $N(v)$ be the set of its neighbors, and $deg(v) = |N(v)|$ be its degree in G. The maximum degree of G is denoted by $\Delta = \max_{v \in V} deg(v)$. Unless otherwise stated, the graph $G = (V, E)$ representing the network is always assumed to be simple (i.e., without multiple edges and loops), undirected and connected.

A global assignment of the interfaces to the nodes in V is given in terms of an appropriate interface assignment function W, fulfilling to the following definition.

Definition 1 A function $W\colon V \to 2^{\{1,2,\ldots,k\}}$ is said to *cover* graph G if for each $\{u, v\} \in E$ we have $W(u) \cap W(v) \neq \emptyset$.

The cost of activating an interface i is given by the cost function $c\colon \{1, 2, \ldots, k\} \to \mathbb{R}_+$ and it is denoted as $c(i)$. It follows that each node holding an interface i pays the same cost $c(i)$ by activating i. The considered *MMCov* optimization problem is formulated as follows.

MMCov: Minimum Maximum-Cost Coverage in Multi-Interface Networks

Input:	A graph $G = (V, E)$, an allocation of available interfaces $W\colon V \to 2^{\{1,2,\ldots,k\}}$ covering graph G, an interface cost function $c\colon \{1, 2, \ldots, k\} \to \mathbb{R}_+$.
Solution:	An allocation of active interfaces $W_A\colon V \to 2^{\{1,2,\ldots,k\}}$ covering G such that $W_A(v) \subseteq W(v)$ for all $v \in V$.
Goal:	Minimize the maximum cost of the active interfaces among all the nodes, i.e. $\min_{W_A} \max_{v \in V} \sum_{i \in W_A(v)} c(i)$.

Similarly, the considered *MMCon* optimization problem is formulated as follows.

MMCon: Minimum Maximum-Cost Connectivity in Multi-Interface Networks

Input:	A graph $G = (V, E)$, an allocation of available interfaces $W\colon V \to 2^{\{1,2,\ldots,k\}}$ covering graph G, an interface cost function $c\colon \{1, 2, \ldots, k\} \to \mathbb{R}_+$.
Solution:	An allocation of active interfaces $W_A\colon V \to 2^{\{1,2,\ldots,k\}}$ covering a connected subgraph $G' = (V, E')$ of G such that $W_A(v) \subseteq W(v)$ for all $v \in V$, and $E' \subseteq E$.
Goal:	Minimize the maximum cost of the active interfaces among all the nodes, i.e. $\min_{W_A} \max_{v \in V} \sum_{i \in W_A(v)} c(i)$.

We recall that two variants of the above problems are considered: when the parameter k is part of the input (i.e., the unbounded case), and when k is a fixed constant (i.e., the bounded case).

It is worth to mention that for trees the *MMCon* problem coincides with *MMCov* since in such cases the connectivity requirement implies the coverage of the whole input graph. Hence, the following statement holds.

Proposition 1 *When the input graph is a tree, any solution for MMCon is also a solution for MMCov at the same cost.*

3 Complexity of *MMCov*

In this section, we study the complexity of *MMCov*. First, we prove that the problem is *NP*-hard and then we identify special cases where it is polynomially solvable.

Theorem 1 *MMCov is NP-hard even when restricted to the bounded unit cost case, for any fixed $\Delta \geq 5$ and $k \geq 16$.*

Proof We prove that the underlying decisional problem, denoted by $MMCov_D$, is in general *NP*-complete. We need to add one bound $B \in \mathbb{R}_+$ such that the problem will be to ask whether there exists an activation function which induces a maximum cost of the active interfaces per node of at most B.

The problem is in *NP* as, given an allocation function of active interfaces for an instance of $MMCov_D$, to check whether it covers the input graph G with a maximum cost of active interfaces per node of at most B is linear in the size of the instance.

The proof then proceeds by a polynomial reduction from the well-known *Satisfiability* problem. The problem is known to be *NP*-complete [19] and it can be stated as follows.

SAT: Satisfiability
Input: Set U of variables and collection C of clauses over U.   **Question**: Is there a satisfying truth assignment for C?

SAT remains *NP*-complete even if there are at most three literals for each clause and a variable appears, negate or not, in at most three clauses [19]. Moreover, the problem remains *NP*-complete even if we assume that there are no clauses with a single literal. Then, in what follows, we assume that each clause has two or three literals and each variable belongs to at most three clauses.

Given an instance of *SAT*, we can build an instance of $MMCov_D$ in polynomial time as follows (see Fig. 3 for an example of such instance). Let $B = 3$. The graph $G = (V, E)$ of $MMCov_D$ has, for each variable $u \in U$, three nodes a_u, b_u, c_u in V and two edges $\{a_u, b_u\}$ and $\{a_u, c_u\}$. For each clause $q \in C$, G has two nodes d_q, e_q in V and an edge $\{d_q, e_q\}$. Let $D = \{d_q \in V \mid q \in C\}$. If clause q has two literals, we add a new node f_q and the edge $\{d_q, f_q\}$. Finally,

for each variable u and each clause q containing u, the graph G has an edge $\{a_u, d_q\}$.
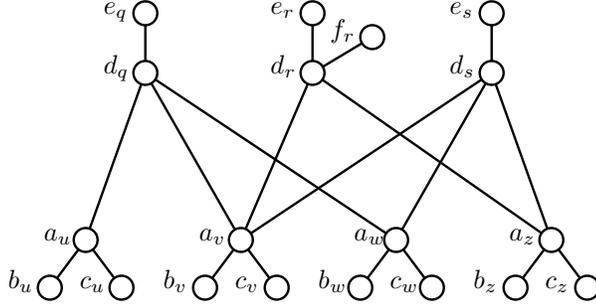

Fig. 2 Graph G in Theorem 1 corresponding to clauses $q = (\overline{u} \vee v \vee w)$, $r = (v \vee \overline{z})$ and $s = (v \vee \overline{w} \vee z)$, where $W(e_q) = \{F_u, T_v, T_w\}$, $W(e_r) = \{T_v, F_z\}$, and $W(e_s) = \{T_v, F_w, T_z\}$.

Note that nodes in G have degree at most five, then $\Delta = 5$.

There are three interfaces I_b, I_c, I_d and, for each variable $u \in U$, two further interfaces: T_u and F_u.

For each $u \in U$, node a_u has four interfaces: T_u, F_u, I_b, I_c, node b_u has interface I_b and node c_u has interface I_c. For each clause $q \in C$, node d_q has interfaces T_w, F_w for each variable $w \in U$ that appears in q; d_q has an additional interface I_d if q has only two literals; e_q has either interface F_w or T_w, according to whether the occurrence of the literal w in q is negate or not, respectively, for each variable $w \in U$ that appears in q. Nodes f_q have only interface I_d for each $q \in C$ having only two literals.

It is easy to see that the above instance of SAT can be computed in polynomial time.

Let us assume that SAT admits a satisfying truth assignment for its variables. For each variable $u \in U$, we activate interfaces I_b and I_c in a_u, b_u, c_u, and if u has true (false, resp.) assignment, we activate interface T_u (F_u, resp.) in a_u.

For each clause $q \in C$, and for each variable w in q, we activate on nodes d_q, e_q interfaces T_w if the corresponding literal has a true value, F_w otherwise. Moreover, if q has only two literals, we activate interfaces I_d on nodes d_q and f_q.

Now, the number of active interfaces for each node is at most $B = 3$ and each edge is covered. In fact edges $\{a_u, b_u\}$ and $\{a_u, c_u\}$ are covered by interfaces I_b and I_c, respectively, for each $u \in U$. Each edge $\{d_q, f_q\}$ is covered by interface I_d for each clause $q \in C$ with two literals. As there exists at least one true literal for each clause q, then edge $\{d_q, e_q\}$ is covered by the corresponding interface F_w or T_w according to whether the occurrence of the literal w is negate or not, respectively. Finally, for each clause $q \in C$ and for

Algorithm 1
1.   **if** $\exists\, i \in \bigcap_{v \in V} W(v)$ **then**
2. $W_A(v) := \{i\}$ for each $v \in V$
3.   **else**
4.   **for each** $v \in V$ s.t. $|W(v)| \leq 2$
5. $W_A(v) := W(v)$
6.   **for each** $v \in V$ s.t. $|W(v)| = 3$
7.   **if** $\exists$ a set $S \subset W(v)$, $|S| = 2$ s.t. $\forall\, u \in N(v)$, $S \cap W(u) \neq \emptyset$ **then**
8. $W_A(v) := S$
9.   **else**
10. $W_A(v) := W(v)$

Fig. 3

each variable u in c, each edge $\{a_u, d_q\}$ is covered by the interface T_u or F_u according to whether u is true or false, respectively.

On the contrary, let us assume that $MMCov_D$ has a positive answer. Then both interfaces I_b and I_c are active on each node a_u, $u \in U$, to cover edges $\{a_u, b_u\}$ and $\{a_u, c_u\}$. Being $B = 3$, each a_u can activate either T_u or F_u to cover edges connecting it to nodes in D. For each $u \in U$, if a_u activates T_u we assign true to u, otherwise, if a_u activates F_u we assign false to u.

Now, each node $d_q \in D$, where q is a clause in C, activates either interface T_u or interface F_u for each variable u in q (and interface I_d, if it has only two variables, to cover edge $\{d_q, f_q\}$). Being $B = 3$, one of these interfaces is also used to cover edge $\{d_q, e_q\}$, corresponding to a true value for one literal in q. Then q is satisfied.

This shows that $MMCov_D$ is *NP*-complete. To show that the problem remains *NP*-complete even if k is bounded, note that it is not necessary to use all the interfaces T_u and F_u for each variable $u \in U$. In fact it is sufficient that each node $d \in D$ has a set of distinct interfaces, two for each variable in the corresponding clause. Then, provided that two variables x and y never appear at the same time into a single clause, the pair of interfaces T_x and F_x associated to x can be reused for y.

To assign interfaces to variables, and in particular to nodes in a_u, $u \in U$, we proceed as follows. We build the conflict graph $H = (U, E_H)$, where there is an edge $\{u, v\}$ in E_H between two variables u, v in U if and only if there exists a clause $q \in C$ and u, v are in q. We find a coloring of H and, if $\chi(u)$ is the color assigned to a variable $u \in U$ we replace the pair of interfaces T_u and F_u with the pair $T_{\chi(u)}$ and $F_{\chi(u)}$ in each node of G. As H has maximum degree 6, it is possible to color it with at most 7 colors in polynomial time. In conclusion, 14 interfaces (two for each color) are sufficient. Concerning I_d, at each pair of connected nodes $d_q \in D$ and f_q, interface I_q can be substituted by any interface among the 14 used by the previous coloring which has not been already assigned to d_q. Other two interfaces I_b and I_c completes the set. Hence, we require a total of 16 interfaces. □

We now give some special cases where *MMCov* is optimally solvable in polynomial time.

Theorem 2 *In the unit cost case with $k \leq 3$, MMCov is optimally solvable in $O(n)$ time.*

Proof The proof is based on the analysis of Algorithm 1. When $k = 2$, either there exists one common interface for all the nodes (code lines 1–2 of Algorithm 1), or the optimal solution costs 2 which implies to activate all the available interfaces at all the nodes (code lines 4–5). Note that, in this case code lines 6–10 are not executed as no node holds more than 2 interfaces. When $k = 3$, if there exists a solution of cost 1 (code lines 1–2), again it is easily verifiable by checking whether all the nodes hold one common interface. If not, in order to check whether there exists a solution of cost 2, the algorithm proceeds as follows. At code lines 4–5, all the interfaces at the nodes are activated. For each node v holding 3 interfaces, it is possible to check whether at most 2 interfaces among the available 3 are enough to connect v to all its neighbors holding less than 3 interfaces. If not, then the optimal solution costs 3 and all the nodes can activate all their interfaces to accomplish the coverage task (code lines 9–10). If yes, then v activates the 2 interfaces induced by its neighborhood (code lines 7–8). In this way, all the edges connecting two nodes holding at most 2 interfaces and all the edges connecting a node holding 3 interfaces with a node holding at most 2 interfaces are covered. In order to conclude the proof, we need to show that all the edges between nodes holding 3 interfaces are covered by the designed activation function. Indeed, since each node holding 3 interfaces activates 2 interfaces, every two of such neighbors must share at least one common interface, and the claim holds.   □

We now show that when the input graph is a tree and we further assume that $k = O(1)$ or $\Delta = O(1)$, then *MMCov* is polynomially solvable in $O(n)$ or $O(k^{2\Delta}n)$, respectively, by a dynamic programming algorithm.

Let us consider a node $v \in V$, we introduce the following notation: T^v is the rooted undirected tree obtained from G by using v as a root; for each $u \in V$, $T^v(u)$ is the subtree of T^v rooted in u and $N^v(u)$ is the set of neighbors of u which belong to $T^v(u)$. Given a set of interfaces S, the cost of activating all the interfaces in S is denoted by $c(S) = \sum_{i \in S} c(i)$.

Note that, for any optimal solution $W_A \colon V \to 2^{\{1,2,\ldots,k\}}$, then $|W_A(u)| \leq \deg(u)$, for each $u \in V$. Therefore, for each node $u \in V$, we define the set $\mathcal{W}(u)$ of subsets of $W(u)$ covering $N(u)$ whose size is at most $\deg(u)$, formally:

$$\mathcal{W}(u) = \{S \subseteq W(u) \mid \forall z \in N(u), W(z) \cap S \neq \emptyset \text{ and } |S| \leq \deg(u)\}.$$

Given a rooted tree T^v and a node $u \in T^v$, for each set of interfaces $S \in \mathcal{W}(u)$, we introduce a data structure $C^v[u, S]$ which stores the minimal cost that has to be paid to cover the subtree $T^v(u)$ if we choose S to cover $N(u)$. Intuitively, $C^v[u, S]$ is given by the maximum among $c(S)$ and, for each $z \in N^v(u)$, the minimal cost that has to be paid to cover $T^v(z)$ by activating in z a set of interfaces $S_z \in \mathcal{W}(z)$ which shares at least one interface with S (i.e. $S_z \cap S \neq \emptyset$). Formally, $C^v[u, S]$ is defined as,

$$C^v[u, S] = \max_{z \in N^v(u)} \left\{ c(S), \min_{S_z \in \mathcal{W}(z), S_z \cap S \neq \emptyset} \{C^v[z, S_z]\} \right\}. \qquad (1)$$

Note that, in the above expression, if u is a leaf of T_v, then $N^v(u) = \emptyset$. It follows that, in this case, $C^v[u, S] = c(S)$.

The following lemma allows us to compute an optimal solution by recursively compute $C^v[u, S]$, starting from the leaves of T^v and going upwards until v is reached.

Lemma 1 *If the input graph is a tree, then the optimal value of MMCov is given by* $\min_{S \in \mathcal{W}(v)} C^v[v, S]$.

Proof Given a node u, and a set of interfaces $S \in \mathcal{W}(u)$, let us define as $\mathrm{OPT}^v(u, S)$ the optimal value of the subproblem consisting of tree $T^v(u)$, assuming that u activates the set of interfaces in S. The optimal value OPT of the original instance is then given by $\mathrm{OPT} = \min_{S \in \mathcal{W}(v)} \mathrm{OPT}^v(v, S)$. Hence, it is sufficient to show that $\mathrm{OPT}^v(u, S) = C^v[u, S]$, for each $u \in V$ and $S \in \mathcal{W}(u)$. The proof is by induction on the height h of the tree $T^v(u)$.

If $h = 1$, then the subproblem considered is made only of node u and its neighbors $N(u)$. Hence $\mathrm{OPT}^v(u, S) = c(S)$. By Equation 1, $C^v[u, S] = \max_{z \in N^v(u)} \{c(S), \min_{S_z \in \mathcal{W}(z), S_z \cap S \neq \emptyset} \{C^v[z, S_z]\}\}$. Since nodes z are leaves of T^v, then $S_z \subseteq S$, for each $S_z \in \mathcal{W}(z)$. Hence, $C^v[z, S_z] = c(S_z) \leq c(S)$ and then $C^v[u, S] = c(S)$.

If $h > 1$, by inductive hypothesis, let us assume that $\mathrm{OPT}^v(z, S_z) = C^v[z, S_z]$, for each $z \in N^v(u)$ and $S_z \in \mathcal{W}(z)$. By cut-and-paste arguments we can show that:

$$\mathrm{OPT}^v(u, S) = \max_{z \in N^v(u)} \left\{ c(S), \min_{S_z \in \mathcal{W}(z), S_z \cap S \neq \emptyset} \{\mathrm{OPT}^v(z, S_z)\} \right\}. \qquad (2)$$

In fact, let us consider an assignment of interfaces $\hat{W} : V \to 2^{\{1,\dots,k\}}$ such that $\hat{W}(u) = S$ and $\sum_{x \in T^v(u)} c(\hat{W}(x)) = \mathrm{OPT}^v(u, S)$ and let us suppose that, in $\hat{W}$, z is any node of $N^v(u)$ that maximizes Equation 2 and that $\hat{W}$ does not induce a cost $\mathrm{OPT}^v(z, S_z)$ over $T^v(z)$, where $S_z = \hat{W}(z)$, that is $\sum_{x \in T^v(z)} c(\hat{W}(x)) > \mathrm{OPT}^v(z, S_z)$. Then, we can cut out from $\hat{W}$ the part defined for $T^v(z)$ and paste in an optimal assignment for $T^v(z)$ of cost $\mathrm{OPT}^v(z, S_z)$, hence obtaining a value that is smaller than $\mathrm{OPT}^v(u, S)$, a contradiction.

By inductive hypothesis and equation 1,

$$\mathrm{OPT}^v(u, S) = \max_{z \in N^v(u)} \left\{ c(S), \min_{S_z \in \mathcal{W}(z), S_z \cap S \neq \emptyset} \{C^v[z, S_z]\} \right\} = C^v[u, S].$$

$\qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \square$

Theorem 3 *When the input graph is a tree then MMCov can be optimally solved in $O(n)$ time if $k = O(1)$, and it can be optimally solved in $O(k^{2\Delta} n)$ time if $\Delta = O(1)$.*

Proof By Lemma 1, it is sufficient to compute $C^v[u, S]$, for each $u \in V$ and $S \in \mathcal{W}(u)$. Hence a straightforward algorithm is given by the definition of $C^v[u, S]$, that is $C^v[u, S]$ is computed recursively, starting from the leaves of

T^v and going upwards to v. Moreover, while computing $C^v[u, S]$, we need to store, for each $z \in N^v(u)$, the sets $S_z \in \mathcal{W}(z)$ which give the minimum value in Equation 1.

Note that, for each $u \in V$, $|W(u)| \leq k$, and hence $|\mathcal{W}(u)| \leq 2^{|W(u)|} \leq 2^k$. Therefore, if $k = O(1)$, then $|\mathcal{W}(u)| = O(1)$. It follows that, for each node $u \in V$ and $S \in \mathcal{W}(u)$, computing $C^v[u, S]$ by using Equation 1 requires

$$O\left(\sum_{z \in N^v(u)} |\mathcal{W}(z)|\right) = O(\deg(u)).$$

Since $|\mathcal{W}(u)| = O(1)$, computing $C^v[u, S]$ for all the possible S requires $O(\deg(u))$ for each node $u \in V$. Therefore, the overall computational time is $\sum_{u \in V} O(deg(u)) = O(\sum_{u \in V} deg(u)) = O(n)$.

Moreover, for each $u \in V$, $\deg(u) \leq \Delta$. Hence $|\mathcal{W}(u)| \leq |W(u)|^{\deg(u)} \leq k^\Delta$, implying that, if $\Delta = O(1)$, then $|\mathcal{W}(u)| = O(k^\Delta)$. It follows that, for each node $u \in V$ and $S \in \mathcal{W}(u)$, computing $C^v[u, S]$ by using Equation 1 requires

$$O\left(\sum_{z \in N^v(u)} |\mathcal{W}(z)|\right) = O(k^\Delta \deg(u)).$$

Since $|\mathcal{W}(u)| = O(k^\Delta)$, computing $C^v[u, S]$ for all the possible S requires $O(k^{2\Delta} \deg(u))$ for each node $u \in V$. Therefore, the overall computational time is $\sum_{u \in V} O(k^{2\Delta} deg(u)) = O(\sum_{u \in V} k^{2\Delta} deg(u)) = O(k^{2\Delta} n)$. $\square$

Corollary 1 *If the input graph is a path, MMCov can be optimally solved in* $O(k^4 n)$ *time.*

Proof It is enough to note that a path is a tree with $\Delta = 2$. $\square$

The case of cycles requires some more insights.

Theorem 4 *If the input graph G is a cycle, MMCov can be optimally solved in* $O(k^6 n)$ *time.*

Proof Let OPT be the optimal solution over G, and x be a generic node. In OPT, x makes use of interface i to establish the connection with one of its neighbors, say y, and interface j for the other neighbor z. Possibly, $i \equiv j$. If we consider the path obtained by removing $\{x, y\}$ from G and by adding a new neighbor x' to y, with x holding only interface j and x' holding only interface i, then the solution OPT is also an optimal solution with respect to the obtained path. In fact, if there exists a better solution OPT$'$ for the obtained path, it must activate the only available interfaces i and j at nodes x' and x to communicate with y and z, respectively, and then it saves something with respect to OPT on the other connections. This would imply that by activating at node x interfaces i and j in the original cycle and by following the solution provided by OPT$'$ for the other nodes, we should obtain a better solution for G with respect to OPT, despite its optimality.

The aforementioned property suggests a way to compute an optimal solution for cycles by means of an algorithm for paths. In order to find the optimal solution for G, we may consider all the path instances obtainable as previously described by associating to x and x' only one interface, possibly the same one, among the original set of interfaces associated with x in G. Such paths are at most $\binom{k}{2} + k = O(k^2)$, and we choose the solution which minimizes the cost in the original cycle G. Hence, by applying the algorithm from Theorem 3 for the case of $\Delta = 2$ for all the obtained path instances, we can find the optimal solution in $O(k^2 \cdot k^4 n)$. □

4 Approximation results for *MMCov*

In this section, we study the approximability properties of *MMCov*. We first show that the problem is not approximable within a factor of $\eta \ln(\Delta)$, for a certain constant η, and then we devise a polynomial time algorithm which guarantees an approximation factor of $O((1+b)\ln(\Delta))$ with b being a parameter depending on structural properties of the input graph. We remind that such a parameter can be bounded by a constant in many graph classes. Moreover, we give two simple approximation algorithms which guarantee factors of approximation of $1 + (k-2)\frac{c_{max}}{2c_{min}}$ and $\frac{c_{max}}{c_{min}}\frac{\Delta}{2}$, where c_{min} and c_{max} are the minimum and the maximum cost associated with an interface, respectively. For the unit cost case this result lead to $\frac{k}{2}$- and $\frac{\Delta}{2}$-approximation algorithm.

Theorem 5 *Unless $P = NP$, MMCov in the unit cost unbounded case cannot be approximated within an $\eta \ln(\Delta)$ factor for a certain constant η, even when the input graph is a tree.*

Proof The proof provides a polynomial time algorithm that transforms any instance I_1 of *Set Cover* (*SC*) into an instance I_2 of *MMCov* with unit costs such that the optimum value SOL^*_{SC} on I_1 for the problem SC is equal to the optimum value SOL^*_{MMCov} on I_2 for the problem *MMCov*.

SC: Set Cover	
Input:	A set U with n elements and a collection $S = \{S_1, S_2, \ldots, S_q\}$ of subsets of U.
Solution:	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'.
Goal:	Minimize $|S'|$.

The graph G is a star of $n+1$ nodes, that is, one for each element of U and a node connected to all the other ones constituting the center. There are $k = q$ interfaces of unitary cost, one for each subset in S. Each node corresponding to an element belonging to a subset S_i holds interface i.

Let $SOL_{SC}(I_1, \sigma_1)$ be the cost of a solution σ_1 for the instance I_1 of SC, and let $SOL^*_{SC}(I_1)$ be the optimal cost for instance I_1. Moreover, let $SOL_{MMCov}(I_2, \sigma_2)$ be the cost of a solution σ_2 for the instance I_2 of *MMCov*, and let $SOL^*_{MMCov}(I_2)$ be the optimal cost for *MMCov* 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 available interfaces $i_1, i_2, \ldots, i_m$ in G, and in particular at the central node, we obtain a feasible solution σ for I_2 such that $SOL(I_2, \sigma) = SOL_{SC}^*(I_1)$, hence: $SOL_{MMCov}^*(I_2) \leq SOL_{SC}^*(I_1)$.

Now we show that it is possible to transform in polynomial time any solution σ_2 for the instance I_2 of $MMCov$ into a solution σ_1 for the instance I_1 of SC such that $SOL_{SC}(I_1, \sigma_1) = SOL_{MMCov}(I_2, \sigma_2)$. A solution σ_2 consists in activating a set of interfaces $\{i_1, i_2, \ldots i_m\}$ at the central node in such a way that each leaf shares at least one active interface with the center. We obtain a covering of all the elements of U by means of subsets $S_{i_1}, S_{i_2}, \ldots, S_{i_m}$ corresponding to the subsets of nodes holding interfaces $\{i_1, i_2, \ldots i_m\}$. As a consequence, $SOL_{SC}(I_1, \sigma_1) = SOL_{MMCov}(I_2, \sigma_2)$.

If there exists an α factor approximation algorithm A for $MMCov$, 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 by using the above transformation in an instance I_2 of $MMCov$ and applying A to find an α-approximate solution σ_2. Hence $SOL_{SC}(I_1, \sigma_1) = SOL_{MMCov}(I_2, \sigma_2) \leq \alpha SOL_{MMCov}^*(I_2) \leq \alpha SOL_{SC}^*(I_1)$. In [1], the authors show that no approximation algorithm for SC exists with an approximation factor less than $\eta \ln |U|$, for a certain constant η. Then there is no algorithm for $MMCov$ with an approximation factor less than $\eta \ln |U| = \eta \ln(n-1) = \eta \ln(\Delta)$.   □

In order to devise an approximation algorithm, we provide a characterization of the graph according to the existence of a *b-bounded ownership function* [11, 17]. Given a graph $G = (V, E)$, an ownership function $Own : E \to V$ is a function that assigns each edge $\{u, v\}$ to an *owner* node between u or v. The set of nodes connected to node u by the edges owned by u is denoted as $Own'(u)$, i.e., $Own'(u) = \{v \mid Own(\{u, v\}) = u\}$. Function Own is said to be b-bounded if the maximum number of edges owned by a node is less than or equal to b, that is $|Own'(u)| \leq b$ for each $u \in V$.

b-bounded ownership function and parameter b can be computed in polynomial time by using structural properties of the graph. For example b is easily bounded by the *maximum degree*, the *treewidth*, and the *arboricity* of G. Hence, for graphs with bounded degree, treewidth or arboricity, a b-bounded accounting function can by computed in polynomial time [7, 18]. In [11], the authors provide a linear time algorithm to find a 3-bounded ownership function for *planar graph*. In [17] it has been observed that for a graph with *pagenumber* p, $b \leq p$ and that, as for graphs with *genus* g, $p = O(\sqrt{g})$ [23], then $b = O(1 + \sqrt{g})$. Moreover, for any graph $g \leq m$, and then for general graphs $b = O(\sqrt{m})$. Finally, in [8] it has been observed that for general graphs a b-bounded accounting function, where $b = O\left(\frac{m}{n}\right)$, can be computed in polynomial time. All these bounds are summarized in Table 1.

The approximation algorithm is given in Figure 4. It is based on suitable instances of Set Cover, we recall that the definition of such problem is that given in the proof of Theorem 5.

General graphs	$b = O(\sqrt{m})$, $b = O\left(\frac{m}{n}\right)$
Planar graphs	$b \leq 3$
Graph with genus g	$b = O(1 + \sqrt{g})$
Graphs with arboricity a	$b \leq a$
Graphs with maximum degree Δ	$b \leq \Delta$
Graphs with pagenumber p	$b \leq p$
Graphs with treewidth t	$b \leq t$

Table 1 Known bounds on ownership functions for some graph classes.

Algorithm 2

1. Compute a b-bounded accounting function Own for G
2. **for each** node $u \in V$
3.   Define an instance $I_{SC}(u) = \{U, S = \{S_1, S_2, \ldots, S_k\}\}$ of SC as follows
4. $U := N(u) \setminus Own'(u)$
5.   **for each** $i \in W(u)$
6. $S_i := \{v \in N(u) \setminus Own'(u) \mid i \in W(v)\}$
7.   Solve $I_{SC}(u)$ by using the best known polynomial approximation algorithm for SC, let S' be the obtained solution, and activate at u the set of interfaces corresponding to S'
8.   Activate at each $v \in N(u) \setminus Own'(u)$ the interface of minimal cost in $S' \cap W(v)$

Fig. 4

Algorithm 2 activates a coverage of the graph. In fact, for each node u, it covers all the edges $\{u, v\}$, $v \in N(u) \setminus Own'(u)$ at code lines 7–8 by activating the interfaces corresponding to a solution of $I_{SC}(u)$. The remainder edges $\{u, v\}$, $v \in Own'(u)$ are covered during the iteration related to node v of the cycle at line 2–8. In fact, by definition of accounting function, $u \in N(v) \setminus Own'(v)$.

It is easy to see that Algorithm 2 is polynomial and its computational time is given by the algorithms used to compute function Own at code line 1 and to solve $I_{SC}(u)$ at code line 7. The following theorem gives us the approximation bound for Algorithm 2.

Theorem 6 *Let I be an instance of MMCov where the input graph admits a polynomial-time computable b-bounded ownership function, the solution provided by Algorithm 2 guarantees a $(\ln(\Delta) + 1 + b \cdot \min\{\ln(\Delta) + 1, c_{max}\})$-approximation factor, with $c_{max} = \max_{i \in \{1, \ldots k\}} c(i)$.*

Proof Let OPT denote the cost of an optimal solution for I, we show that the solution provided by Algorithm 2 has a cost C such that $C \leq (\ln(\Delta) + 1 + b \cdot \min\{\ln(\Delta) + 1, c_{max}\}) \cdot$ OPT. Given a node $u \in V$, let us denote as OPT$_{SC}(u)$ and $C_{SC}(u)$ the cost of an optimal solution for instance $I_{SC}(u)$ of $SC(u)$ defined at code lines 3–6 and the cost of the solution for $I_{SC}(u)$ computed at code line 7, respectively. Moreover, let OPT$_{SC} = \max_{u \in V}\{$OPT$_{SC}(u)\}$ and $C_{SC} = \max_{u \in V}\{C_{SC}(u)\}$.

Node u will activate a set of interfaces corresponding to the solution computed at code line 7 at the cost of $C_{SC}(u) \leq C_{SC}$ plus the cost of $|Own'(u)|$ interfaces for the connection to nodes in $Own'(u)$ activated at code line 8, in the iteration related to such nodes. Note that, the cost of each interface

induced on u by nodes $v \in Own'(u)$ cannot be bigger than both c_{max} and $C_{SC}(v) \leq C_{SC}$. Moreover, as $|Own'(u)| \leq b$ we obtain,

$$C \leq C_{SC} + |Own'(u)| \cdot \min\{C_{SC}, c_{max}\} \leq C_{SC} + b \cdot \min\{C_{SC}, c_{max}\}.$$

Let us denote as $\text{OPT}(u)$ the cost at u induced by an optimal solution. By definition, for any optimal solution $\text{OPT}(u) \leq \text{OPT}$. Moreover, as an optimal solution has to cover all the edges incident to u,

$$\text{OPT}_{SC}(u) \leq \text{OPT}(u).$$

From [20], there exists a $(\ln |U| + 1)$-approximation algorithm for weighted SC that can be applied at code line 7. Therefore, since $|U| \leq \Delta$,

$$C_{SC}(u) \leq (\ln |U| + 1) \cdot \text{OPT}_{SC}(u) \leq (\ln(\Delta) + 1) \cdot \text{OPT}(u).$$

As the above inequalities hold for any $u \in V$, it follows that

$$C_{SC} \leq (\ln(\Delta) + 1) \cdot \text{OPT},$$

and hence,

$$C \leq (\ln(\Delta) + 1) \cdot \text{OPT} + b \cdot \min\{(\ln(\Delta) + 1) \cdot \text{OPT}, c_{max}\}.$$

$\square$

As simple approximation algorithms, we can easily guarantee a factor of approximation of $\frac{c_{max}}{c_{min}} k$ or $\frac{c_{max}}{c_{min}} \Delta$ by activating all the interfaces available at all nodes or by activating the cheapest interface for each edge, respectively. The following theorems slightly improve these bounds.

Theorem 7 *MMCov is $(1 + (k-2)\frac{c_{max}}{2c_{min}})$-approximable in $O(n)$ time.*

Proof We show that Algorithm 3 provides a $(1 + (k-2)\frac{c_{max}}{2c_{min}})$-approximation for *MMCov*. Let us assume that an optimal solution activates a maximal-cost set of interfaces I^* at a certain node. If the test at line 1 of Algorithm 3 returns true, then the obtained solution activates a certain interface i in each node of G. If $|I^*| = 1$, then i is an optimal solution, otherwise, the approximation ratio is:

$$\frac{c(i)}{\sum_{j \in I^*} c(j)} \leq \frac{c_{max}}{|I^*| c_{min}} \leq \frac{c_{max}}{2c_{min}}.$$

If the test at line 1 of Algorithm 3 returns false, then $|I^*| \geq 2$ and Algorithm 3 activates all the interfaces available at each node. It follows that the approximation ratio is at most:

$$\frac{\sum_{i \in I} c(i)}{\sum_{j \in I^*} c(j)} = 1 + \frac{\sum_{i \in I \setminus I^*} c(i)}{\sum_{j \in I^*} c(j)} \leq 1 + \frac{|I \setminus I^*| c_{max}}{|I^*| c_{min}} \leq 1 + \frac{(k-2)c_{max}}{2c_{min}},$$

which is the worst of the above two cases. $\square$

Algorithm 3
1.   **if** $\bigcap_{v \in V} W(v) \not\equiv \emptyset$ **then**
2. $W_A(v) = \mathrm{argmin}\{c(i) \mid i \in \bigcap_{u \in V} W(u)\}$, for each $v \in V$
3.   **else**
4. $W_A(v) \equiv W(v)$, for each $v \in V$
Fig. 5

Algorithm 4
1.   **if** $\bigcap_{v \in V} W(v) \not\equiv \emptyset$ **then**
2. $W_A(v) = \mathrm{argmin}\{c(i) \mid i \in \bigcap_{u \in V} W(u)\}$, for each $v \in V$
3.   **else**
4.   **for each** $e = \{u, v\} \in E$
5.   choose an interface $i \in W(v) \cap W(u)$
6. $W_A(v) := W_A(v) \cup \{i\}$
7. $W_A(u) := W_A(u) \cup \{i\}$
8. $E := E \setminus \{e\}$
Fig. 6

Corollary 2 *In the unit cost case MMCov is $\frac{k}{2}$-approximable in $O(n)$ time.*

Theorem 8 *MMCov is $\frac{c_{max}}{c_{min}} \frac{\Delta}{2}$-approximable in $O(n + m)$ time.*

Proof We show that Algorithm 4 provides a $\frac{c_{max}}{c_{min}} \frac{\Delta}{2}$-approximation for *MMCov*. If exists an optimal solution which activates only one interface for each node, then Algorithm 4 finds it at lines 1–2. If the optimal solution activates at least two interfaces for each node, then Algorithm 4 activates for each node, at most one interface for each neighbor. Hence, the maximum number of interfaces that a single node can activate is Δ. As any optimal solution costs at least $2c_{min}$, the approximation bound is at most $\frac{c_{max}}{c_{min}} \frac{\Delta}{2}$.   □

Corollary 3 *In the unit cost case MMCov is $\frac{\Delta}{2}$-approximable in $O(n)$ time.*

5 Complexity and approximation results for *MMCon*

In this section we study the complexity and the approximation bounds of *MMCon*. First, we prove that the problem is *NP*-hard, even for the unit cost case and even when the number of interfaces k and the maximum node degree Δ are fixed. Then, we present efficient algorithms that optimally solve the problem in some relevant special cases. Then, we show that the problem is not approximable within an $\eta \ln(\Delta)$ factor for a certain constant η, unless $P = NP$. Hence, we give some simple approximation algorithms. As the inapproximability result holds when the input graphs is restricted to trees, we provide a $(\ln(\Delta) + 1 + \min\{\ln(\Delta) + 1, c_{max}\})$-approximation algorithm for this special case. Note that, this bound is $\ln(\Delta) + 2$ for the unit cost case. Some of the results stated are derived from results given for *MMCov*.

Theorem 9 *MMCon is NP-hard even when restricted to the bounded unit cost case, for any fixed $\Delta \geq 3$ and $k \geq 10$.*

Proof We prove that the underlying decisional problem, denoted by $MMCon_D$, is in general NP-complete. We need to add one bound $B \in \mathbb{R}_+$ such that the problem will be to ask whether there exists an activation function which induces a maximum cost of the active interfaces per node of at most B.

The problem is in NP as, given an allocation function of active interfaces for an instance of $MMCon_D$, to check whether it covers a connected spanning subgraph of G with a maximum cost of active interfaces per node of at most B is linear in the size of the instance.

The proof then proceeds by a polynomial reduction from the well-known *Hamiltonian Path* problem. The problem is known to be NP-complete [19] and it can be stated as follows:

HP: Hamiltonian Path

Input:	Graph $H = (V_H, E_H)$
Question: Does H contain a Hamiltonian path?	

Given an instance of *HP*, we can build an instance of $MMCon_D$ in polynomial time as follows. Let $B = 2$. The graph G of $MMCon_D$ is the input graph H of *HP*. Regarding the interfaces, we associate a distinct interface to each edge in G. That is, the set of interfaces $W(v)$ of each node v in G is given by the interfaces associated to the edges incident on v. Then, if $e = \{u, v\}$ is an edge of G, we define $W(e) = W(u) \cap W(v)$. By construction, for each edge e in G we have:

1. $|W(e)| = 1$;
2. $W(e) \neq W(e')$, for each pair of edges e, e' such that $e \neq e'$.

Let us assume that H admits an Hamiltonian path P. Then the connected subgraph of G to be covered is $G' \equiv P$ and, for each node $v \in G'$, the set of active interface $W_A(v)$ is given by the interfaces associated to the edges in G' incident on v. As G' is a path, there are at most two active interfaces for each node, and then, being $B = 2$, $MMCon_D$ has a positive answer.

On the contrary, let us assume that $MMCon_D$ has a positive answer. Let G' be the covered spanning subgraph and let $W_A(v)$, $v \in V$, be such that $|W_A(v)| \leq B = 2$. As a consequence of the above properties 1 and 2, the degree of v in G' is less than or equal to 2. Since G' is connected, then G' must be either a path or a cycle. This implies that H admits a Hamiltonian path.

Now, let us show that the problem is NP-complete even if k and Δ are bounded. We assume here that each vertex of H has degree 3, in fact *HP* remains NP-complete even with this restriction [19].

Clearly, the above proof remains valid if we assign the interfaces in such a way that for each node v, each interface in $W(v)$ covers only one edge incident on v. Formally, we have to find an assignment of the interfaces to the nodes such that

1. $|W(e)| = 1$, for each e in G;
2. $W(u) \cap W(v) \cap W(w) = \emptyset$, for each pair of distinct edges $\{u, v\}$, $\{v, w\}$ in G.

To this end, we consider the strong edge-coloring of graph G, that is, an edge-coloring in which every color class is an induced matching. In other words, any two vertices belonging to distinct edges with the same color are not adjacent. It is known [2] that cubic graphs, that are those having degree 3 for each node, admit a strong edge-coloring by means of 10 colors.

Now, let us associate to each edge e in G the interface corresponding to the color assigned to e in the above coloring. Then $W(v)$, for each $v \in G$, is given by the interfaces associated to the edges incident on v, as above. It remains to show that this interface assignment fulfills property number 2, as by construction $|W(e)| = 1$, for each e in G.

We write e_1^x, e_2^x, and e_3^x to denote the three edges of each node $x \in V$. Then $W(x) = W(e_1^x) \cup W(e_2^x) \cup W(e_3^x)$. By contradiction, let us assume that there are two edges $\{u, v\}$, $\{v, w\}$ in G such that $W(u) \cap W(v) \cap W(w)$ is not empty. This means that there are three edges e_j^u, e_k^v and e_l^w, for suitable values j, k, and l, such that $W(e_j^u) = W(e_k^v) = W(e_l^w)$. As at least two edges among e_j^u, e_k^v and e_l^w are distinct, then the associated interfaces are different, a contradiction.   □

In the case of bounded number of interfaces or bounded degree, we are able to solve *MMCon* in polynomial time when the input graph is a tree. In fact, by Proposition 1 and Theorem 3, we can state the following theorem.

Theorem 10 *When the input graph is a tree then MMCon can be optimally solved in $O(n)$ time if $k = O(1)$, and it can be optimally solved in $O(k^{2\Delta}n)$ time if $\Delta = O(1)$.*

Moreover, we can derive a simple polynomial time algorithm for fixed $\Delta \leq 2$ that is, for paths and cycles.

Corollary 4 *If the input graph is a path, MMCon can be optimally solved in $O(k^4 n)$ time.*

Corollary 5 *If the input graph is a cycle, MMCon can be optimally solved in $O(k^4 n^2)$ time.*

Proof It suffices to choose the cheapest solutions among the ones obtained by solving the n possible paths obtained from the input cycle by excluding one different edge at time.   □

The next theorems allow us to optimally solve the problem in two special cases: when the input graph is a polynomially recognizable Hamiltonian graph in the unit cost case; and when $k \leq 2$.

Theorem 11 *In the unit cost case, if the input graph is a polynomially recognizable Hamiltonian graph, MMCon can be optimally solved in $O(k^4 n)$ time.*

Proof If all the nodes hold one common interface, then an optimal solution is given by activating such interface and the minimum cost is 1. Otherwise, an optimal solution costs at least 2. Hence, by solving the problem restricted to an Hamiltonian path of the input graph we obtain solution of cost 2 which is optimal.   □

Theorem 12 *MMCon is polynomially solvable when $k \leq 2$.*

Proof If an optimal solution uses only one interface, then it can be easily found by checking whether all the nodes hold the same interface. Otherwise any optimal solution needs to activate both the available interfaces at some node, regardless the cost of the interfaces.   □

In Section 4, it has been shown that the *MMCov* problem is not approximable within $\eta \ln(\Delta)$ for a certain constant η, by an approximation factor preserving reduction from Set Cover. As such a reduction is based on a star topology, from Proposition 1 the following theorem holds.

Theorem 13 *MMCon in the unit cost unbounded case cannot be approximated within an $\eta \ln(\Delta)$ factor for a certain constant η, unless $P = NP$.*

The simple approximation algorithms provided for *MMCov* can be also applied for *MMCon*, obtaining the same approximation factors. This is stated in the next theorems, whose proof is similar to that of Theorems 7 and 8.

Theorem 14 *MMCon is $(1 + (k-2)\frac{c_{max}}{2c_{min}})$-approximable in $O(n)$ time.*

Corollary 6 *In the unit cost case MMCon is $\frac{k}{2}$-approximable in $O(n)$ time*

Theorem 15 *MMCon is $\frac{c_{max}}{c_{min}}\frac{\Delta}{2}$-approximable in $O(n+m)$ time.*

Corollary 7 *In the unit cost case MMCon is $\frac{\Delta}{2}$-approximable in $O(n)$ time.*

From Theorem 6, and considering that on trees $b = 1$, the following theorem and corollary can be stated.

Theorem 16 *When the input graph is a tree, there exists a polynomial algorithm for MMCon which provides a $(\ln(\Delta) + 1 + \min\{\ln(\Delta) + 1, c_{max}\})$-approximation, with $c_{max} = \max_{i \in \{1,\ldots k\}} c(i)$.*

Corollary 8 *In the unit cost case, when the input graph is a tree, there exists an algorithm for MMCon which provides a $(\ln(\Delta) + 2)$-approximation.*

6 Conclusion

We have considered the Coverage (*MMCov*) and the Connectivity (*MMCon*) problems in Multi-Interface Networks. The new objective function with respect to previous works in this area considers the minimization of the maximum cost required by the single nodes of the network. We focused on problems hardness and approximation factors in general and more specific settings. We have shown that *MMCov* is *NP*-hard for any fixed $\Delta \geq 5$, while it is polynomially solvable for $\Delta \leq 2$. Moreover, it is *NP*-hard for any fixed $k \geq 16$ while it is polynomially solvable for $k \leq 3$. For fixed k, $4 \leq k \leq 15$ and for fixed Δ, $3 \leq \Delta \leq 4$, the complexity of *MMCov* remains open. Concerning approximation results for *MMCov*, we have shown that the problem is not approximable

within a factor of $\eta \ln(\Delta)$ for a certain constant η, unless $P = NP$. This result holds even in the unit cost case and when the input graph is a tree, but only when k and Δ are unbounded. If either $k = O(1)$ or $\Delta = O(1)$, *MMCov* can be optimally solved in $O(n)$ or $O(k^{2\Delta}n)$ time, respectively. Then, we have provided a general approximation algorithm for trees that guarantees a factor of $\ln(\Delta) + 1 + b \cdot \min\{c_{max}, (\ln(\Delta) + 1)\}$, with $c_{max} = \max_{i \in \{1,...k\}} c(i)$ and b being a parameter depending on structural properties of the input graph. Simple approximation algorithms guarantee $1 + (k - 2)\frac{c_{max}}{2c_{min}}$ and $\frac{c_{max}}{c_{min}}\frac{\Delta}{2}$ factors of approximation, where c_{min} and c_{max} are the minimum and the maximum cost associated with an interface, respectively.

Concerning *MMCon*, we have shown that it is *NP*-hard for any fixed $\Delta \geq 3$, while it is polynomially solvable for $\Delta \leq 2$. Moreover, it is *NP*-hard for any fixed $k \geq 10$ while it is polynomially solvable for $k \leq 2$. For fixed k, $3 \leq k \leq 9$, the complexity of *MMCon* remains open. Since *MMCon* coincides with *MMCov* when the input graph is a tree, it follows that also *MMCon* is not approximable within $\eta \ln(\Delta)$ when k and Δ are both unbounded and even in the unit cost case. When at least one parameter among k or Δ is bounded, the problem can be optimally solved in $O(n)$ or $O(k^{2\Delta}n)$ time, respectively. We give some simple approximation algorithms for the general case which guarantee a factor of approximation of $1 + (k - 2)\frac{c_{max}}{2c_{min}}$ or $\frac{c_{max}}{c_{min}}\frac{\Delta}{2}$. Moreover, for trees, a general $(\ln(\Delta) + 1 + \min\{\ln(\Delta) + 1, c_{max}\})$-approximation algorithm can be applied. This guarantees a $(\ln(\Delta) + 2)$-approximation factor for the unit cost case.

This paper represents a first step towards distributed approaches as the objective function refers to local parameters rather than global ones. Further investigations on experimental results and modifications to the proposed model are of main interest.

References

1. N. Alon, D. Moshkovitz, and S Safra. Algorithmic construction of sets for k-restrictions. *ACM Transactions on Algorithms*, 2(2):153–177, 2006.
2. L.D. Andersen. The strong chromatic index of a cubic graph is at most 10. *Discrete Mathematics*, 108(1–3):231–252, 1992.
3. S. Athanassopoulos, I. Caragiannis, C. Kaklamanis, and E. Papaioannou. Energy-efficient communication in multi-interface wireless networks. In *Proc. of the 34th Int. Symp. on Mathematical Foundations of Computer Science (MFCS)*, volume 5743 of *Lecture Notes in Computer Science*, pages 102–111. Springer, 2009.
4. P. Bahl, A. Adya, J. Padhye, and A. Walman. Reconsidering wireless systems with multiple radios. *SIGCOMM Comput. Commun. Rev.*, 34(5):39–46, 2004.
5. F. Barsi, A. Navarra, and M.C. Pinotti. Cheapest paths in multi-interface networks. In *Proc. of the 10th Int. Conf. on Distributed Computing and Networking (ICDCN)*, volume 5408 of *Lecture Notes in Computer Science*, pages 37–42. Springer, 2009.

6. S. Basagni, F. Nidito, and A. Farago. The multi-radio advantage. In *Proc. of the IEEE Radio and Wireless Symposium (RWS)*, pages 478–481. IEEE, 2009.

7. H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

8. F. Bruera, S. Cicerone, G. D'Angelo, G. Di Stefano, and D. Frigioni. Dynamic multi-level overlay graphs for shortest paths. *Mathematics in Computer Science*, 1(4):709–736, 2008.

9. 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. Workshop on Software and Performance (WOSP)*, pages 42–45. ACM Press, 2007.

10. 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.

11. M. Chrobak and D. Eppstein. Planar orientations with low out-degree and compaction of adjacency matrices. *Theoretical Computer Science*, 86(2):243 – 266, 1991.

12. G. D'Angelo, G. Di Stefano, and A. Navarra. Minimizing the Maximum Duty for Connectivity in Multi-Interface Networks. In *Proc. of the 4th Annual International Conference on Combinatorial Optimization and Applications (COCOA)*, volume 6509, Part II of *Lecture Notes in Computer Science*, pages 254–267. Springer, 2010.

13. G. D'Angelo, G. Di Stefano, and A. Navarra. Bandwidth Constrained Multi-Interface Networks. In *Proc. of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 6543 of *Lecture Notes in Computer Science*, pages 202–213. Springer, 2011.

14. G. D'Angelo, G. Di Stefano, and A. Navarra. Min-Max Coverage in Multi-Interface Networks. In *Proc. of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 6543 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 2011.

15. R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of the 10th annual international conference on Mobile computing and networking (MobiCom)*, pages 114–128. ACM, 2004.

16. A. Faragó and S. Basagni. The effect of multi-radio nodes on network connectivity—a graph theoretic analysis. In *Proc. of the IEEE Int. Workshop on Wireless Distributed Networks (WDM)*. IEEE, 2008.

17. D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni. Semidynamic algorithms for maintaining single-source shortest path trees. *Algorithmica*, 22(3):250–274, 1998.

18. H. Gabow and H. Westermann. Forests, frames, and games: Algorithms for matroid sums and applications. *Algorithmica*, 7:465–497, 1992.

19. 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.
20. D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and Sysntem Sciences*, 9:256–278, 1974.
21. 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.
22. A. Kosowski, A. Navarra, and M.C. Pinotti. Exploiting Multi-Interface Networks: Connectivity and Cheapest Paths. *Wireless Networks*, 16(4):1063–1073, 2010.
23. S.M. Malitz. Genus g graphs have pagenumber $O(\sqrt{g})$. *Journal of Algorithms*, 17(1):85–109, 1994.