Min-Max Coverage in Multi-Interface Networks

Gianlorenzo D'Angelo[1], Gabriele Di Stefano[1], and Alfredo Navarra[2]

[1] Department of Electrical and Information Engineering, University of L'Aquila, Italy. gianlorenzo.dangelo@univaq.it gabriele.distefano@univaq.it
[2] Department of Mathematics and Computer Science, University of Perugia, Italy. navarra@dmi.unipg.it

Abstract. We consider devices equipped with multiple wired or wireless interfaces. By switching among interfaces or by combining the available interfaces, each device might establish several connections. A connection is established when the devices at its endpoints share at least one active interface. Each interface is assumed to require an activation cost. In this paper, we consider the problem of establishing the connections defined by a network $G = (V, E)$ while keeping as low as possible the maximum cost set of active interfaces at the single nodes. Nodes V represent the devices, edges E represent the connections that must be established. We study the problem of minimizing the maximum cost set of active interfaces among the nodes of the network in order to cover all the edges. We prove that the 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 the problem cannot be approximated within $\Omega(\ln \Delta)$. 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 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.

1 Introduction

The heterogeneity of modern devices poses new challenges to the scientific community. The interest is also increased by the wide range of real-world applications inferred. For instance, the equipment of recent devices provides users with the opportunity to access to different networks by means of the selection of suitable interfaces. Classical problems related to wired and wireless networks can be reconsidered with respect to the new environment. Different computational power, energy consumption, radio interfaces, supported communication protocols, and other peculiarities can characterize the involved devices. In this paper, we are interested in multiple interfaces equipments where 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 the set of required connections according to proximity of devices 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 x is activated at some node u, then u consumes some energy $c(x)$ for maintaining x as active. In this setting, we study the problem of covering all the edges of G by minimizing the maximum cost required at the single nodes. This implies that the cost imposed by all the interfaces activated in the whole network to accomplish the coverage requirement 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.

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 [6, 9, 10]. Many basic problems of standard wireless network optimization can be reconsidered in such a setting [2]. However, previous works have been mainly focused on the minimization of the costs among the whole network. In [5, 14], for instance, the same problem of *Coverage* has been investigated, but with the goal of activating the minimum cost set of interfaces among all the nodes in the network in such a way that all the edges of G are covered. *Connectivity* issues have been addressed in [1, 8, 14, 15]. 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 particular, [8] considers the connectivity task under the same objective function of this paper, i.e., the minimization of the maximum cost spent by each single node. In [3, 15], 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.

1.2 Our results

In this paper, we study the problem of 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 (*MMCC* for short). The chosen requirement is intended as 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 problem: 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.

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. 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, we give a polynomial time algorithm for $k \leq 3$. For fixed k, $4 \leq k \leq 15$, and fixed Δ, $3 \leq \Delta \leq 4$, the complexity of *MMCC* remains open.

Concerning approximation results for *MMCC*, 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 $c_{max} = \max_{i \in \{1,\ldots k\}} c(i)$ 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. Another approximation factor which is directly implied by [8] is $\frac{k}{2}$. This clearly might be useful for small values of k.

1.3 Structure of the paper

In the next section, we formally define the problem of covering all the edges of the input graph by minimizing the maximum cost required at the single nodes and give some preliminary results. In Section 3, we study the complexity of *MMCC* 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 a polynomial time approximation algorithms for both the general case and particular cases. In Section 5, 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, according 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 *MMCC* optimization problem is formulated as follows.

MMCC: 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. $\min_{W_A} \max_{v \in V} \sum_{i \in W_A(v)} c(i)$.

We recall that two variants of the above problem 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). In both cases we assume $k \geq 2$, since the case $k = 1$ admits the obvious solution provided by activating the unique interface at all the nodes.

It is worth to mention that for paths and trees the *MMCC* problem coincides with the Connectivity problem studied in [8] where the aim is to allow a communication path between any pair of nodes. In fact, the following statement holds.

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

3 Complexity

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

Theorem 1. *MMCC 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 $MMCC_D$, 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. In detail, $MMCC_D$ is defined as follows.

$MMCC_D$	
***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 $c\colon \{1, 2, \ldots, k\} \to \mathbb{R}_+$, and a bound $B \in \mathbb{R}_+$.
***Question*:**	Is there 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 $v \in V$ and $\max_{v \in V} \sum_{i \in W_A(v)} c(i) \leq B$?

The problem is in NP as, given an allocation function of active interfaces for an instance of $MMCC_D$, 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 [12] 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, negated or not, in at most three clauses [12]. Moreover, the problem remains NP-complete even if we assume that there are no clauses with a single literal. Then, in the following reduction, 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 $MMCC_D$ in polynomial time as follows. Let $B = 3$. The graph $G = (V, E)$ of $MMCC_D$ 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 $\{a_u, d_q\}$.

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.

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 $u \in U$. 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 w is negated in q or not, 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.

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 T_w or F_w according to the literal is w or $\overline{w}$. Finally, for each clause $q \in C$ and for each variable u in q, 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 $MMCC_D$ 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 interface 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 $MMCC_D$ 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 there exists a node $d \in D$ and two edges $\{a_u, d\}$, $\{a_v, y\}$ in G. We find a minimum 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 conclusion, 14 interfaces 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. □

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

Algorithm 1.

```
1.  V' := ∅
2.  if ∃ i ∈ ⋂_{v∈V} W(v) then
3.    W_A(v) := {i} for each v ∈ V
4.  else
5.    for each v ∈ V s.t. |W(v)| ≤ 2
6.      V' := V' ⋃ {v}
7.      W_A(v) := W(v)
8.    for each v ∈ V s.t. |W(v)| = 3
9.      if ∃ a set S ⊂ W(v), |S| ≤ 2, s.t.
          ∀ u ∈ N(v) ∩ V', S ∩ W_A(u) ≠ ∅, then
10.       W_A(v) := S
11.       if |S| = 1 then activate one further arbitrary interface at v
12.     else
13.       if ∀ u ∈ N(v), |W(u)| = 3 then
14.         activate two arbitrary interfaces at v
15.       else
16.         W_A(v) := W(v)
```

Proof. The proof is based on the analysis of Algorithm 1. The case $k = 1$ is trivial and it is solved by code lines 2–3 of the algorithm. When $k = 2$, either there exists one common interface for all the nodes (again code lines 2–3 of Algorithm 1), or the optimal solution costs 2 which equals to activate all the available interfaces at all the nodes (code lines 5–7). Note that in this case code lines 8–16 are not executed as no node holds more than 2 interfaces. When $k = 3$, if there exists a solution of cost 1 (code lines 2–3), again it is easily verifiable by checking whether all the nodes hold one same interface. If not, in order to check whether there exists a solution of cost 2, it is possible to activate all the interfaces at the nodes holding less than 3 interfaces. This can be realized as at code lines 8–16. 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 15–16). If yes, then v activates the 2 interfaces induces by its neighborhood (code lines 9–10); if only 1 or 0 interfaces are induced by the neighborhood then v activates one further (code line 11) or two interfaces (code lines 13–14), respectively, chosen arbitrarily. In this way, all the edges connecting nodes holding at most 2 interfaces and all the edges connecting nodes holding 3 interfaces with nodes 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. The above algorithm requires $O(m)$ time, as the execution of code lines 9–11 might refer to all the edges of the input graph. □

In [8], it has been shown that when the input graph is a tree and $k = O(1)$ or $\Delta = O(1)$, then Connectivity is polynomially solvable in $O(n)$ or $O(k^{2\Delta}n)$, respectively. Hence, by Proposition 1, we can state the following theorem.

Theorem 3. *If the input graph is a tree and $k = O(1)$ or $\Delta = O(1)$, MMCC can be optimally solved in $O(n)$ or $O(k^{2\Delta}n)$ time, respectively.*

Theorem 3 implies that if the input graph is a path, *MMCC* can be optimally solved in $O(k^4n)$ time. The case of cycles requires some more insights.

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

In this section, we study the approximability properties of *MMCC*. We first show that the problem is not approximable within $\Omega(\ln(\Delta))$, 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 the reader that such a parameter can be bounded by a constant in many graph classes.

In [8], it has been shown that the Connectivity problem is not approximable within $\eta \ln(\Delta)$ for a certain constant η, by an approximation factor preserving reduction from Set Cover (SC). As such a reduction is based on a star topology, from Proposition 1 the following theorem holds.

Theorem 5. *Unless $P = NP$, MMCC 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.*

In order to devise an approximation algorithm, we provide a characterization of the graph according to the existence of a *b-bounded ownership function* [7, 11]. 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', i.e., $Own'(u) = \{v | Own(\{u, v\}) = u\}$. 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 $u \in V$.

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. In [7], the authors provide a linear time algorithm to find a 3-bounded ownership function for *planar graph*. In [11] it has been observed that for a graph with *pagenumber* p, $b \leq p$ and that, as for graphs with *genus* g, $p = O(\sqrt{g})$ [16], then $b = O(1 + \sqrt{g})$. Moreover, for any graph $g \leq m$, and then for general graphs $b = O(\sqrt{m})$. Finally, in [4] it has been observed that for general graphs $b = O\left(\frac{m}{n}\right)$. All these b-bounded functions can be computed in polynomial time. The resulting bounds are summarized in Table 1.

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.

The approximation algorithm is given in Figure 2. It is based on suitable instances of Set Cover. Here we remind the definition of such a problem:

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 $\|S'\|$.

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

Algorithm 2.

Algorithm 2
1. Compute a b-bounded accounting function Own for G
2. **for each** node $u \in V$
3. Define an instance $I_{SC}(u)$ 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 approximation algorithm for SC and activate at u the corresponding set of interfaces
8. Activate at each $v \in N(u) \setminus Own'(u)$ the interface of minimal cost in $\{i \mid S_i \in S'\} \cap W(v)$

interfaces corresponding to a solution of $I_{SC}(u)$. While edges $\{u, v\}$, $v \in Own'(u)$ are covered during the iteration related to node v as, 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 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 MMCC where the input graph admits a 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 $c_{max} = \max_{i \in \{1,\dots k\}} c(i)$.*

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 $C \leq (\ln(\Delta) + 1 + b \cdot \min\{\ln(\Delta) + 1, c_{max}\}) \cdot \text{OPT}$. Given a node $u \in V$, let us denote as $\text{OPT}_{SC}(u)$ 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 $\text{OPT}_{SC} = \max_{u \in V}\{\text{OPT}_{SC}(u)\}$ 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 $|Own'(u)|$ 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 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 [13], there exists a $(\ln|U|+1)$-approximation algorithm for weighted SC that can be applied at code line 7. Therefore, since $|U| \leq \Delta$,

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

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

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

and hence,

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

□

Note that, the previous theorem is a generalization of the result stated in [8] for trees. In fact, tree topologies induce $b = 1$, hence obtaining a $2(\ln(\Delta)+1)$-approximation factor.

Finally, it is worth to mention a further approximation factor that can be achieved for the unit cost case.

Theorem 7. *In the unit cost case, MMCC is $\frac{k}{2}$-approximable in $O(n)$ time.*

The theorem is based on the fact that an optimal solution either activates the same interface among all the nodes (if possible) or it must activates at least two interfaces at some node. Hence, a simple algorithm can check whether all the nodes share a common interface or activates all the available interfaces at all the nodes. Such interfaces are at most k by definition.

5 Conclusion

We have considered the Coverage problem 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 problem hardness and approximation factors in general and more specific settings. In summary, *MMCC* 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 *MMCC* remains open.

Concerning approximation results for *MMCC*, we show 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 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 $c_{max} = \max_{i \in \{1,\ldots k\}} c(i)$ and b being a parameter depending on structural properties of the input graph. Another approximation algorithm guarantees a $\frac{k}{2}$ factor of approximation.

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. S. Athanassopoulos, I. Caragiannis, C. Kaklamanis, and E. Papaioannou. Energy-efficient communication in multi-interface wireless networks. In *34th Int. Symp. on Mathematical Foundations of Computer Science (MFCS)*, volume 5743 of *LNCS*, pages 102–111. Springer, 2009.
2. P. Bahl, A. Adya, J. Padhye, and A. Walman. Reconsidering wireless systems with multiple radios. *SIGCOMM Comput. Commun. Rev.*, 34(5):39–46, 2004.
3. F. Barsi, A. Navarra, and M.C. Pinotti. Cheapest paths in multi-interface networks. In *10th Int. Conf. on Distributed Computing and Networking (ICDCN)*, volume 5408 of *LNCS*, pages 37–42. Springer, 2009.
4. 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.
5. M. Caporuscio, D. Charlet, V. Issarny, and A. Navarra. Energetic Performance of Service-oriented Multi-radio Networks: Issues and Perspectives. In *6th Int. Workshop on Software and Performance (WOSP)*, pages 42–45. ACM Press, 2007.
6. 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.
7. M. Chrobak and D. Eppstein. Planar orientations with low out-degree and compaction of adjacency matrices. *Theoretical Computer Science*, 86(2):243 – 266, 1991.
8. G. D'Angelo, G. Di Stefano, and A. Navarra. Minimizing the Maximum Duty for Connectivity in Multi-Interface Networks. In *Proceedings of the 4th Annual International Conference on Combinatorial Optimization and Applications (COCOA)*, volume 6509 Part II of *LNCS*, pages 254–267. Springer, 2010.
9. R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *10th annual international conference on Mobile computing and networking (MobiCom)*, pages 114–128. ACM, 2004.
10. A. Faragó and S. Basagni. The effect of multi-radio nodes on network connectivity—a graph theoretic analysis. In *IEEE Int. Workshop on Wireless Distributed Networks (WDM)*. IEEE, 2008.
11. D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni. Semidynamic algorithms for maintaining single-source shortest path trees. *Algorithmica*, 22(3):250–274, 1998.
12. 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.
13. D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and Sysntem Sciences*, 9:256–278, 1974.
14. 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.
15. A. Kosowski, A. Navarra, and M.C. Pinotti. Exploiting Multi-Interface Networks: Connectivity and Cheapest Paths. *Wireless Networks*, 16(4):1063–1073, 2010.
16. S.M. Malitz. Genus g graphs have pagenumber $O(\sqrt{g})$. *Journal of Algorithms*, 17(1):85–109, 1994.