

# Exploiting Multi-Interface Networks: Connectivity and Cheapest Paths<sup>\*</sup> <sup>\*\*</sup>

Adrian Kosowski<sup>1,2</sup>, Alfredo Navarra<sup>3</sup>, and Maria Cristina Pinotti<sup>3</sup>

<sup>1</sup> LaBRI - Université Bordeaux 1, 351 cours de la Liberation, 33405 Talence, France.

<sup>2</sup> Department of Algorithms and System Modeling, Gdańsk University of Technology, Narutowicza 11/12, 80952 Gdańsk, Poland. E-mail: [kosowski@sphere.pl](mailto:kosowski@sphere.pl)

<sup>3</sup> Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, 06123 Perugia, Italy. E-mails: [{navarra, pinotti}@dmi.unipg.it](mailto:{navarra, pinotti}@dmi.unipg.it)

**Abstract.** Let  $G = (V, E)$  be a graph which models a set of wireless devices (nodes  $V$ ) that can communicate by means of multiple radio interfaces, according to proximity and common interfaces (edges  $E$ ). The problem of switching on (activating) the minimum cost set of interfaces at the nodes in order to guarantee the coverage of  $G$  was recently studied. A connection is covered (activated) when the endpoints of the corresponding edge share at least one active interface. In general, every node holds a subset of all the possible  $k$  interfaces. Such networks are known as multi-interface networks. In this setting, we study two basic problems: *Connectivity* and *Cheapest Path*.

The Connectivity problem corresponds to the well-known Minimum Spanning Tree problem in graph theory. In practice, we need to cover a subgraph of  $G$  of minimum cost which contains a spanning tree of  $G$ . The problem turns out to be *APX*-hard in general and for many restricted graph classes, however it is possible to provide approximation algorithms: a 2-approximation in general and a  $(2 - \frac{1}{k})$ -approximation for the unit cost interface case, i.e. when the cost of activating an interface is unitary for any interface. We also consider the problem in special graph classes, such as graphs of bounded degree, planar graphs, graphs of bounded treewidth, complete graphs.

The Cheapest Path problem corresponds to the well-known Shortest Path problem in graph theory. In the multi-interface setting this problem is still polynomially solvable, and we point out a simple Dijkstra-based algorithm with  $O(k|E| + k|V|\log(k + |V|))$  runtime in general and  $O(k(|E| + |V|))$  runtime for the unit cost interface case.

**Keywords:** energy saving, wireless network, multi-interface network

## 1 Introduction

As technology advances and more sophisticated and heterogeneous devices are released, special effort is required for managing new kinds of communication problems. Nowadays wireless devices hold multiple radio interfaces, and are capable of switching from one communication network to another according to required connectivity and related quality considerations. The selection of the most suitable radio interface for a specific connection might depend on various factors. Such factors include: its availability in specific devices, the required communication bandwidth, the cost (in terms of energy use) of maintaining an active interface, the available neighbours, 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

---

<sup>\*</sup> The research was partially funded by the European project COST Action 293, “Graphs and Algorithms in Communication Networks” (GRAAL).

<sup>\*\*</sup> Preliminary results concerning this paper appeared in [2, 15].

the network survivability might depend on their persistence in the network. This introduces challenging and natural optimisation problems which must take care of different variables at the same time.

In the considered model, the input network is described by a graph  $G = (V, E)$  with  $n = |V|$  and  $m = |E|$ , where  $V$  represents the set of wireless devices and  $E$  is the set of possible 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  $\bigcup_{v \in V} W(v)$ , and we denote the cardinality of this set by  $k$ .

We say that a connection is *satisfied* when the endpoints of the corresponding edge share at least one active interface. For an activation which satisfies some connections, the overall cost is given as the sum of cost values associated with each interface, taken over all nodes which activate the interface.

In these terms, the *Connectivity* problem can be described as follows. What is the cheapest way of establishing connectivity in the network, i.e., which subset of available interfaces must be activated at each node in order to guarantee a path between every pair of nodes in  $G$  while minimising the overall cost of activation? The *Cheapest Path* problem has the following intuitive formulation: given a source node  $s \in V$ , for each other node  $t \in V$ , which subset of available interfaces in some nodes must be activated in order to guarantee a path between  $s$  and  $t$ , while minimising the overall cost? Formal definitions of these problems are given in the subsequent sections of the paper.

The interest in these problems comes from the necessity to study, in the context of multi-interface networks, problems corresponding to the well-known problems of determining the Minimum Spanning Tree and the Shortest Path Tree for standard graphs.

## 1.1 Related Work

Multi-interface wireless networks have recently been studied in a variety of contexts, usually focusing on the benefits of multiple radio devices of each node. Many basic problems of standard wireless network optimisation can be reconsidered in such a setting [1], focusing in particular on issues related to routing [8] and network connectivity [6, 9].

The problems considered herein originate from [5] where a slightly different model is introduced. That model considers the necessity of activating all the connections expressed by  $G$  while minimising the overall cost. Such a problem can be referred to as *Coverage* of  $G$ . Different interfaces may have different costs and moreover, in some works, mutually exclusive interfaces have been considered (this means that some interfaces, if activated, preclude the activation of some other interfaces). In [5], a sketchy proof of *NP*-hardness of the *Coverage* problem was provided and experimental results were shown. In [12, 13] the *Coverage* problem was formally defined (under the name of *CMI*). The number of interfaces  $k$  was assumed to be a known constant. An algorithmic approach led to interesting hardness and approximation results for various graph classes like complete graphs, trees, planar graphs, graphs of bounded degree and general graphs. Moreover, both unit cost and non-unit cost interface cases were considered. Indeed, the unit cost model is equivalent to ask for the minimum total number

of activated interfaces inside the network in order to cover all the connections. Results in a similar context have been obtained in [12, 14] but for a slightly different scenario where  $k$  is not known in advance, i.e.,  $k$  depends on the input instance.

## 1.2 Our Results

For the *Connectivity* problem, the results presented in this paper are summarised in Table 1. *Connectivity* turns out to be *APX*-hard in general and also when restricted to graphs of maximum degree  $\Delta \geq 4$  (even for  $k = 2$  interfaces), as well as for bounded treewidth graphs (including trees) and complete graphs (when  $k$  is unbounded). It remains *NP*-hard for graphs of maximum degree  $\Delta \geq 3$  (when  $k \geq 2$ ) and for planar graphs (when  $k \geq 10$  and  $\Delta \geq 6$ ). Approximation algorithms are then provided for general graphs. A 2-approximation is given for the non-unit cost interface case, while for the unit cost interface case we have a  $(2 - \max\{\frac{1}{k}, \frac{1}{9(\frac{\Delta}{2}) - 4(\Delta - 1)}\})$ -approximation. For  $k = 2$  we also provide a  $\frac{4}{3}$ -approximation in the unit cost interface case. Optimal algorithms are put forward for graphs of bounded treewidth and complete graphs when  $k$  is bounded. When  $k$  is unbounded, an approximation algorithm is provided for complete graphs for the unit cost and non-unit cost interface cases.

**Table 1.** Hardness and approximability for Connectivity. In the table,  $C$  represents the ratio  $\lfloor \frac{c_{max}}{c_{min}} \rfloor$  between the maximum and the minimum cost interfaces available in the network. The ‘\*’ symbol stands for any other available approximation ratio provided for the general case.

Graph class	Complexity of Connectivity	
	unit costs	non-unit costs
General graphs	APX-hard $\forall k \geq 2$ $\frac{4}{3}$ -approx for $k = 2$ $(2 - \frac{1}{k})$ -approx $\forall k \geq 2$ $(2 - \frac{1}{9(\frac{\Delta}{2}) - 4(\Delta - 1)})$ -approx $\forall k \geq 2$	APX-hard $\forall k \geq 2$ 2-approx $\forall k \geq 2$
Bounded $\Delta$	NP-hard $\forall k \geq 2, \Delta \geq 3$ APX-hard $\forall k \geq 2, \Delta \geq 4$	NP-hard $\forall k \geq 2, \Delta \geq 3$ APX-hard $\forall k \geq 2, \Delta \geq 4$
Planar graphs	NP-hard $\forall k \geq 10, \Delta \geq 6$	NP-hard $\forall k \geq 10, \Delta \geq 6$
Bounded treewidth graphs	APX-hard (unbounded $k$ ) [12]	polynomial (bounded $k$ )
Complete graphs	$\min\{*, 1 + \frac{k}{n}\}$ -approx APX-hard (unbounded $k$ )	$\min\{2, C(1 + \frac{k}{n})\}$ -approx $O(n^2)$ (bounded $k$ ) [12]

Concerning *Cheapest Path*, we consider the most generic unbounded case, i.e. when the number  $k$  is not known a priori. Even in such a setting, the problem can be solved in  $O(km + kn \log(kn))$  time by making use of the standard Dijkstra’s technique [7] on a suitable graph. For the unit cost interface case, the time complexity of the algorithm reduces to  $O(k(m + n))$ .

## 1.3 Outline

The next section is devoted to the *Connectivity* problem. It provides definitions and notation, hardness results, and a series of approximation algorithms for general and particular graph

topologies. Namely, graphs of bounded treewidth, graphs of bounded degree, planar graphs and complete graphs are considered. For the *Cheapest Path* problem, an efficient general-case polynomial time algorithm is presented in Section 3. Finally, Section 4 contains concluding remarks.

## 2 Connectivity in Multi-Interface Networks

In this section, we present and study the *Connectivity* problem.

### 2.1 Definitions and Notation

For a graph  $H$ , we denote by  $V(H)$  its vertex set, by  $E(H)$  its edge set, by  $\gamma(H)$  the number of its connected components, by  $\Delta(H)$  its maximum vertex degree, and denote  $n(H) = |V(H)|$ ,  $m(H) = |E(H)|$ . Unless otherwise stated, the graph  $G = (V, E)$  representing the network is always assumed to be simple (i.e., without multiple edges), undirected and connected, with  $n(G) > 2$  (so  $\Delta(G) \geq 2$ ). When considering the network graph  $G$ , we simply denote the number of its vertices, edges, and maximum degree by  $n$ ,  $m$ ,  $\Delta$ , respectively.

An edge  $e \in E$  is said to cost  $\alpha$ , which is denoted by  $\text{cost}(e) = \alpha$ , if the cheapest common interface available at both its endpoints has cost  $\frac{\alpha}{2}$ .

A global characterisation of interfaces of respective nodes from  $V$  is given in terms of an appropriate interface assignment function  $W$ , according to the following definition.

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

The cost of activating an interface for a node is assumed to be identical for all nodes and given by cost function  $c: \{1, \dots, k\} \rightarrow \mathbb{R}_+$ , i.e., the cost of interface  $i$  is denoted as  $c_i$ . The considered *Connectivity* optimisation problem is formulated as follows.

---

### Connectivity in Multi-Interface Networks

---

**Input:** A graph  $G = (V, E)$ , an allocation of available interfaces  $W: V \rightarrow 2^{\{1, \dots, k\}}$  covering graph  $G$ , an interface cost function  $c: \{1, \dots, k\} \rightarrow \mathbb{R}_+$ .

**Solution:** An allocation of active interfaces  $W_A: V \rightarrow 2^{\{1, \dots, k\}}$  covering a connected spanning subgraph  $G' = (V, E')$  of  $G$  such that  $W_A(v) \subseteq W(v)$  for all  $v \in V$ , and  $E' \subseteq E$ .

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

---

Note that we can consider two variants of the above problem: the parameter  $k$  can be considered as part of the input (this is called the *unbounded case*), or  $k$  may be a fixed constant (the *bounded case*). In both cases we assume  $k \geq 2$ , since the case  $k = 1$  admits an obvious unique solution (all nodes must activate their unique interface).

*Connectivity* might be seen as the search for a spanning tree of  $G$  of minimum cost. Such a cost is determined by the set of interfaces that must be activated in order to cover all the edges of such a tree. On the other hand, it might happen that by activating the minimum cost set of interfaces for connectivity purposes, the resulting covered subgraph of  $G$  is not a tree.

## 2.2 Complexity Results

We first prove that the problem is, in general, *APX*-hard.

**Theorem 1.** *Connectivity is APX-hard even when restricted to the unit cost interface case for  $k \geq 2$  and maximum degree  $\Delta \geq 4$ .*

*Proof.* Given an allocation function of active interfaces for an instance of *Connectivity*, to check whether the induced subgraph is connected or not is linear in the number of edges of the input graph  $G$ . The proof then proceeds by a polynomial reduction of the well-known *Minimum 3-Set Cover* problem with *bounded occurrences*. The problem is known to be *APX*-hard [10] and it can be stated as follows:

---

*X3C* : Minimum 3 Set Cover with at most 3 occurrences

---

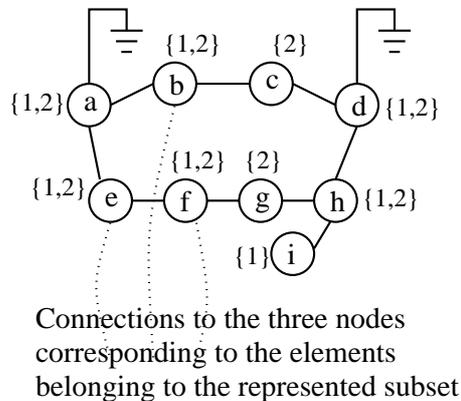
**Input:** Collection  $C$  of subsets, each of cardinality at most 3, of a finite set  $S$  such that each element occurs in at most 3 subsets of  $C$ .

**Solution:** A set cover for  $S$ , i.e. a subset  $C' \subseteq C$  such that every element of  $S$  belongs to at least one member of  $C'$ .

**Goal:** Cardinality of the set cover, i.e.,  $|C'|$ .

---

For our reduction we consider the unit cost interface case and  $k = 2$ . Given an instance  $(C, S)$  of *X3C*, we transform it into an instance  $(G, W)$  of *Connectivity* with  $k = 2$  unit cost interfaces. Each element of  $S$  becomes a node of the input graph of *Connectivity* holding only interface 1. For each subset belonging to the collection  $C$ , we introduce a gadget of nine nodes to the *Connectivity* input instance as shown in Figure 1.



**Fig. 1.** Auxiliary gadget for the reduction from *X3C* to *Connectivity*

To complete the instance transformation in the reduction, we need to explain the meaning of the ground symbol in the figure. This is a connection to an auxiliary graph as will be shown later; for the moment, we can assume that all of the ground connections lead to the same additional node (called the root) shared by all the gadgets. Such a node holds both

interfaces 1 and 2, and it is connected to two further auxiliary nodes holding only interface 1 and interface 2, respectively.

In the modified graph, we see that activating a total of 9 interfaces is enough in order to connect all the nodes of a single gadget to the root. This is done by activating interface 1 at nodes  $d$ ,  $h$  and  $i$ , and interface 2 at all the remaining 6 nodes. In this way, all the nodes of the gadget admit a path to at least one ground connection. Note that this is the only possible activation of 9 interfaces having the property that each node of the gadget admits a connection path to the root. Moreover, this activation cannot satisfy any connection (dotted lines of Figure 1) from the gadget to the elements of the represented subset as by construction the corresponding nodes hold only interface 1.

However, if we allow a cost bigger than 9 for a gadget, then there exists an interface allocation of cost 10 which allows us to connect the gadget to both the root and all the three external nodes corresponding to the represented subset. In fact, by activating interface 1 at node  $i$ , and both interfaces 1 and 2 at node  $h$ , then we can activate interface 1 at nodes  $a$ ,  $b$ ,  $e$  and  $f$ , and interface 2 at the remaining nodes  $c$ ,  $d$  and  $g$ . This configuration connects the gadget to all the nodes of the represented subset to the root.

Finally, regardless of the constraints for the other nodes, we are forced to activate interface 1 for each of the nodes representing elements of  $S$ , both interfaces for the root, and the one available interface for the two auxiliary neighbours of the root. We thus obtain that there exists a solution to the original  $X3C$  instance with cardinality  $|C'|$  if and only if there exists a solution to *Connectivity* in the modified graph with cardinality  $9|C| + |C'| + |S| + 4$ . As  $X3C$  is known to be *APX*-hard even with the additional restriction that  $|C| \leq |S|$  (see [10]), and obviously  $|C'| \geq |S|/3$ , clearly *Connectivity* is *APX*-hard as well.

As the root has high degree, we can modify our construction in order to claim the above result for networks of maximum degree greater than or equal to 4. Instead of the root, for each of the  $2|C|$  ground connections of the gadgets we add a structure behaving like the previous root, i.e. one node holding both interfaces 1 and 2 connected to two nodes holding interface 1 and interface 2, respectively. All these “roots” are then treated as the leaves of a tree of maximum degree 4, and joined together to form such a tree using new auxiliary internal nodes of the tree, which hold only interface 1. In this way all the nodes of the constructed graph have degree less than or equal to 4, and the claim holds.  $\square$

**Corollary 1.** *Connectivity is NP-hard even when restricted to the unit cost interface case for  $k \geq 2$  and maximum degree  $\Delta = 3$ .*

*Proof.* Following the proof of Theorem 1, in order to decrease  $\Delta$  to 3 we need another construction to connect all the roots. To each root we connect the root of a new binary tree that has a number of leaves not less than the square of the number of gadgets,  $|C|^2$ . Nodes of the trees hold both interfaces 1 and 2. Then the  $j$ -th,  $1 \leq j \leq |C|$ , leaf of the  $i$ -th,  $1 \leq i \leq |C|$ , tree is connected to the  $i$ -th leaf of the  $j$ -th tree. In this way, each tree is covered by activating only one interface per node, and all the trees sharing the same active interface are connected. In order to guarantee connectivity we only require that one node among all the nodes of the trees activates both interfaces (this globally affects the cost of the activation in the whole graph by a constant of 1; this is negligible since  $X3C$  is *APX*-hard). Note that

in this way we only prove the *NP*-hardness of *Connectivity*, not its *APX*-hardness, as the number of nodes of the constructed graph is much larger than  $|C|$ .  $\square$

**Theorem 2.** *Connectivity is NP-hard, even when restricted to the unit cost interface case in planar graphs, for all values of  $k \geq 10$  and maximum graph degree  $\Delta \geq 6$ .*

*Proof.* The proof proceeds by reduction from the *Hamiltonian Path in 3-Regular Planar Graphs* problem, which is known to be *NP*-complete [11].

---

*HP3P : Hamiltonian Path in 3-Regular Planar Graphs*

---

**Input:** A 3-regular planar graph  $G$ .

**Question:** Does  $G$  admit a Hamiltonian path?

---

Consider an arbitrary 3-regular planar graph  $G$  which is an instance of the *HP3P* problem. We construct an instance of *Connectivity* as follows. First, we obtain graph  $G'$  from  $G$  by adding to  $G$ , for each vertex  $v \in V(G)$ , exactly three new vertices  $v'_1, v'_2, v'_3$ , and three new edges  $\{v'_1, v\}, \{v'_2, v\}, \{v'_3, v\}$ . Then, we assign interfaces to vertices of  $G'$  as follows. Consider an edge coloring of  $G$  such that edges which are at distance at most 2 (i.e. edges which share a vertex, or which are both adjacent to some other edge) receive different colorings. Such a coloring is known as a *strong edge coloring*, and for 3-regular graphs can always be constructed in polynomial time using at most 10 colors [?]. Now, to each vertex  $v \in V(G)$  we assign exactly 3 interfaces from the range  $\{1, \dots, 10\}$ , corresponding to the colors of edges adjacent to  $v$ . If the interfaces assigned to vertex  $v \in V(G)$  are  $\{c_1, c_2, c_3\}$ , then the added vertices  $v'_1, v'_2, v'_3$  are assigned interfaces  $\{c_2, c_3\}, \{c_1, c_3\}$ , and  $\{c_1, c_2\}$ , respectively. All interfaces have unit cost. We now show that  $G'$  admits a solution to *Connectivity* with cost at most  $5|V(G)|$  if and only if  $G$  has a Hamiltonian path.

First, assume that  $G$  has a Hamiltonian path. For a vertex  $v \in V(G)$ , let  $u$  and  $w$  be its neighbours in the path. Then vertex  $v$  enables interfaces corresponding to the colors of edges  $\{u, v\}$  and  $\{v, w\}$ . (For an endpoint of the Hamiltonian path, enable one interface corresponding to the color of the edge of the path, and one more available interface.) In this way, communication is possible along the Hamiltonian path, hence connectivity within  $V(G)$  is achieved. Connectivity can be extended to the whole of  $V(G)$  by enabling exactly one interface for each vertex from  $V(G') \setminus V(G)$ . This means that exactly two interfaces are enabled for each vertex from  $V(G)$ , and one interface is enabled for each vertex from  $V(G')$ , thus the total cost of the activation is at most  $|V(G)| + |V(G')| = 5|V(G)|$ .

Conversely, suppose that there exists an activation of interfaces with cost at most  $5|V(G)|$ . To enable communication among vertices  $\{v, v'_1, v'_2, v'_3\}$ , for any  $v \in V(G)$ , it is necessary to activate at least 5 interfaces in total for any such 4 nodes, and each vertex must enable at least 1 interface. Since the total cost of the activation is at most  $5|V(G)|$ , we have that each vertex  $v \in V(G)$  can only enable at most 2 interfaces. By the properties of the edge coloring used in the interface assignment, this allows  $v$  to communicate with at most two other neighbours in  $G$ . Hence, since connectivity is achieved by assumption, graph  $G$  must admit a spanning tree of maximum degree 2, which means  $G$  has a Hamiltonian path.  $\square$

**Theorem 3.** *Connectivity is 2-approximable.*

*Proof.* Our algorithm  $\mathcal{A}$  for *Connectivity* first chooses a spanning tree  $T_G$  of the input graph  $G$  of minimum cost with respect to the edge cost function provided in Section 2.1. Once  $T_G$  is chosen, the algorithm proceeds as follows. It simply activates the cheapest interface for covering each edge of  $T_G$ . This means that for each covered edge at most one interface at both endpoints is activated. Moreover, according to the considered cost function, the cost of  $T_G$ , understood as the sum of the costs of its edges, is minimum. In fact, the set of edges required for connectivity purposes by an optimal solution cannot have a cost smaller than the cost of  $T_G$ . Indeed, consider any optimal interface activation  $W_{\text{OPT}}$  for *Connectivity* in  $G$ . It induces some connected spanning subgraph  $G' \subseteq G$ , and we consider an arbitrary spanning tree  $T_{G'}$  of  $G'$ . For each edge  $e \in E(T_{G'})$  there must exist an interface of cost not less than  $\text{cost}(e)$  which is activated in  $W_{\text{OPT}}$  by both the endpoints of  $e$ . Since edges of  $T_{G'}$  can be mapped into distinct vertices of  $V$ , and by the minimality of spanning tree  $T_G$ , we obtain:

$$c(W_{\text{OPT}}) \geq \sum_{e \in E(T_{G'})} \text{cost}(e) \geq \sum_{e \in E(T_G)} \text{cost}(e).$$

On the other hand, for each edge  $e \in E(T_{G'})$ , the proposed algorithm activates the interface of cost  $\text{cost}(e)$  for both its endpoints, leading to an activation  $W_{\mathcal{A}}$  such that:

$$c(W_{\mathcal{A}}) \leq 2 \sum_{e \in E(T_G)} \text{cost}(e).$$

Clearly,  $c(W_{\mathcal{A}}) \leq 2c(W_{\text{OPT}})$ , and the claim holds.  $\square$

**Theorem 4.** *Connectivity is  $(2 - \frac{1}{k})$ -approximable for the unit cost interface case.*

*Proof.* For a not necessarily connected subgraph  $H \subseteq G$  without isolated vertices, let the *gain*  $g(H)$  be defined as  $g(H) = n(H) - 2\gamma(H)$ , where  $n(H)$  is the the number of vertices of  $H$ , and  $\gamma(H)$  is the number of connected components of  $H$ .

The proposed approximation algorithm consists of the following steps:

1. For each interface  $i \in \{1, \dots, k\}$  determine the gain  $g(H_i)$ , where  $H_i$  is obtained by removing all isolated vertices from the subgraph of  $G$  induced by the set of vertices equipped with interface  $i$ ,  $\{v \in V : i \in W(v)\}$ .
2. Let  $j$  be any interface such that  $g(H_j) = \max_{i \in \{1, \dots, k\}} g(H_i)$ . Activate interface  $j$  at all vertices of  $H_j$ . Let  $T$  be a forest in  $G$  consisting of arbitrary spanning trees of all connected components of  $H_j$ .
3. One by one, consider all edges  $\{u, v\} \in E(G) \setminus E(T)$ . If  $T \cup \{u, v\}$  does not contain a cycle, set  $T := T \cup \{u, v\}$  and activate an arbitrary interface shared by nodes  $u$  and  $v$  at both node  $u$  and node  $v$ .

After completing step (2) of the procedure, the number of connected components of the graph, with respect to the current activation, is at most  $n - n(H_j) + \gamma(H_j)$ , while the cost of activation is exactly  $n(H_j)$ . In step (3), it is thus necessary to perform activation for at most  $n - n(H_j) + \gamma(H_j) - 1$  edges, with an activation cost of at most  $2(n - n(H_j) + \gamma(H_j) - 1)$ .

Thus, the total cost  $c(W_a)$  of activation throughout steps (2) and (3) can be bounded from above by:

$$c(W_A) \leq n(H_j) + 2(n - n(H_j) + \gamma(H_j) - 1) = 2(n - 1) - g(H_j). \quad (1)$$

Consider now any optimal activation of interfaces  $W_{opt}$  for graph  $G$ . Let  $T_{opt}$  be an arbitrary spanning tree of the connected subgraph of  $G$  which is activated using  $W_{opt}$ . We can decompose tree  $T_{opt}$  into a union of disjoint forests,  $T_{opt} = T_1 \cup \dots \cup T_k$ , such that in forest  $T_i$  all of the nodes must activate interface  $i$ .<sup>4</sup> Thus, the cost of the optimal solution is bounded from below by  $c(W_{opt}) \geq \sum_{i=1}^k n(T_i) = \sum_{i=1}^k (|E(T_i)| + \gamma(T_i))$ . Taking into account that  $\sum_{i=1}^k |E(T_i)| = |E(T)| = n - 1$  and  $g(T_i) = n(T_i) - 2\gamma(T_i) = |E(T_i)| - \gamma(T_i)$ , we have:

$$c(W_{opt}) \geq 2(n - 1) - \sum_{i=1}^k g(T_i). \quad (2)$$

Now, observe that for all  $i$ , since  $T_i \subseteq H_i$  and  $H_i$  has no isolated vertices, we may write  $g(T_i) \leq g(H_i)$ . Thus, we may develop expression (1) as follows:

$$c(W_A) \leq 2(n - 1) - g(H_j) \leq 2(n - 1) - \frac{1}{k} \sum_{i=1}^k g(H_i) \leq 2(n - 1) - \frac{1}{k} \sum_{i=1}^k g(T_i).$$

Taking into account that  $c(W_{opt}) \geq n$ , we finally obtain from expression (2) and the above:

$$\frac{c(W_A)}{c(W_{opt})} \leq \frac{2(n - 1) - \frac{1}{k} \sum_{i=1}^k g(T_i)}{\max\{n, 2(n - 1) - \sum_{i=1}^k g(T_i)\}} \leq 2 - \frac{1}{k}.$$

□

**Theorem 5.** *Connectivity is  $(2 - \frac{1}{9\binom{\Delta}{2} - 4(\Delta - 1)})$ -approximable for the unit cost interface case.*

*Proof.* We make use of the following observation: if for some activation of two vertices  $u, v$  we have that edge  $\{u, v\}$  is the only edge adjacent to  $u$  or  $v$  which uses some interface  $i$  in the activation, then the contribution of such an edge to the overall cost of the activation is equal to 2 (since interface  $i$  has to be activated by both  $u$  and  $v$  specifically for communication along this edge). Let us build a graph  $P$  whose vertices are all three-vertex paths  $(u, v, w)$  in  $G$  such that  $\{u, v\} \in E(G)$ ,  $\{v, w\} \in E(G)$ , and  $u, v, w$  share at least one available interface, and let two vertices of  $P$  be connected by an edge if the corresponding paths in  $G$  share at least one vertex. By applying once more a standard spanning-tree-based argument, we easily obtain that the cost of any, and thus also any optimal, interface activation  $W_{opt}$  can be bounded from below as:

$$c(W_{opt}) \geq \max\{n, 2(n - 1) - |V(P)|\}. \quad (3)$$

Now, consider an activation obtained using the following procedure:

<sup>4</sup> From the optimality of  $T_{opt}$ , it follows that each  $T_i$  does not contain isolated vertices, as otherwise, switching off the corresponding interfaces would lead to a feasible cheapest solution.

1. Compute an independent set  $M \subseteq V(P)$  in graph  $P$ .
2. For all  $(u, v, w) \in M$ , activate the same shared interface for nodes  $u, v, w$ .
3. Complete the activation by activating one shared interface at the endpoint of at most  $n - 1 - 2|M|$  edges, so as to obtain connectivity.

The correctness of the construction in step (3) lies in the observation that after the completion of step 2, the number of connected components in the already enabled communication graph is at most  $n - 2|M|$  (since we have exactly  $|M|$  three-vertex paths and  $n - 3|M|$  isolated vertices). The cost of the obtained activation is thus equal to at most

$$c(W_{\mathcal{A}}) \leq 3|M| + 2(n - 1 - 2|M|) \leq 2(n - 1) - |M|. \quad (4)$$

It remains to be observed that in graph  $P$  we can find a sufficiently large independent set. Indeed, using a  $(\Delta(P) + 1)$ -coloring based approach [4], we can find in polynomial time an independent set  $M$  such that  $|M| \geq \frac{|V(P)|}{\Delta(P)+1}$ . By a simple local analysis of the possible adjacency relations of three-vertex paths in  $G$ , we have  $\Delta(P) \leq 9\binom{\Delta}{2} - 4(\Delta - 1) - 1$ , thus

$$|M| \geq \frac{1}{9\binom{\Delta}{2} - 4(\Delta - 1)} |V(P)|. \quad (5)$$

Combining relations (3), (4), and (5), we finally reach the sought bound:

$$\frac{c(W_{\mathcal{A}})}{c(W_{opt})} \leq 2 - \frac{1}{9\binom{\Delta}{2} - 4(\Delta - 1)}.$$

□

**Theorem 6.** *Connectivity is  $\frac{4}{3}$ -approximable for  $k = 2$  in the unit cost interface case.*

*Proof.* For any  $I \subseteq \{1, 2\}$ , let  $V_I \subseteq V$  denote all those nodes whose set of available interfaces is exactly  $I$ , giving the partition  $V = V_{\{1\}} \cup V_{\{2\}} \cup V_{\{1,2\}}$ . We confine ourselves to considerations of instances for which  $|V_{\{1,2\}}| > \frac{1}{3}n$ , otherwise the sought approximation ratio can be trivially achieved by activating all interfaces for all nodes of the graph. Without loss of generality we may assume that  $|V_{\{2\}}| \leq |V_{\{1\}}|$ , hence  $|V_{\{2\}}| < \frac{1}{3}n$ . Let  $H_1$  be the induced subgraph of  $G$  with vertex set  $V_{\{1\}} \cup V_{\{1,2\}}$ , and let  $\gamma(H_1)$  be the number of its connected components. Again, without loss of generality we assume that each connected component of  $H_1$  contains at least one node from  $V_{\{1\}}$  (otherwise, all the nodes of this component can be treated as if they belonged to  $V_{\{2\}}$ ). The proposed approximation algorithm consists of the following steps:

1. Activate interface 1 for each node from  $V_{\{1\}} \cup V_{\{1,2\}}$  and interface 2 for each node from  $V_{\{2\}}$ .
2. Activate interface 2 for at most  $|V_{\{2\}}| + \gamma(H_1) - 1$  nodes from  $V_{\{1,2\}}$  so as to establish connectivity.

The details of step (2) require some comment. In fact, after completion of step (1) the graph for which communication is already possible with the current activation has at most  $|V_{\{2\}}| + \gamma(H_1)$  connected components: at most  $|V_{\{2\}}|$  components within  $V_{\{2\}}$  and exactly  $\gamma(H_1)$  connected components within  $V_{\{1\}} \cup V_{\{1,2\}}$ . Thus, by enabling a connection on at most  $|V_{\{2\}}| + \gamma(H_1) - 1$  further edges of the graph we can obtain connectivity with a single connected component. These edges must connect some vertex from  $V_{\{1,2\}}$  with some vertex from  $V_{\{2\}}$  (since for all other edges communication is already possible after step (1)); enabling such an edge requires only the activation of interface 2 on the endpoint of the edge within  $V_{\{1,2\}}$ .

From the description of steps (1) and (2) of the algorithm we can immediately bound the cost of activation of the obtained activation  $W_{\mathcal{A}}$  from above as follows:

$$c(W_{\mathcal{A}}) \leq n + |V_{\{2\}}| + \gamma(H_1) - 1 \quad (6)$$

On the other hand, for any activation of interfaces, each node of the graph must activate at least one interface; moreover, in each connected component  $C$  of  $H_1$  (with the exception of at most one), there must exist a node with two active interfaces, so as to enable communication between component  $C$  and the rest of the graph (observe that by assumption  $C$  contains a vertex from  $V_{\{1\}}$ , while all vertices adjacent to  $C$ , not belonging to  $C$ , belong to  $V_{\{1\}}$ ). Hence, for any optimal activation  $W_{opt}$  we may write:

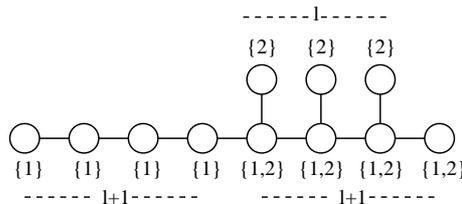
$$c(W_{opt}) \geq n + \gamma(H_1) - 1 \quad (7)$$

Combining expressions (6) and (7) we eventually obtain the sought ratio:

$$\frac{c(W_{\mathcal{A}})}{c(W_{opt})} \leq \frac{n + |V_{\{2\}}| + \gamma(H_1) - 1}{n + \gamma(H_1) - 1} \leq 1 + \frac{|V_{\{2\}}|}{n + \gamma(H_1) - 1} \leq 1 + \frac{|V_{\{2\}}|}{n} < \frac{4}{3}.$$

□

The analysis of the approximation ratio for the above algorithm is tight (for the case  $|V_{\{1,2\}}| > \frac{1}{3}n$ ). Consider for example the class of instances of *Connectivity* shown in Fig. 2. For some



**Fig. 2.** Example of a worst-case instance for the algorithm from Theorem 6

integer  $l \geq 1$ , we have  $n = 3l + 2$ ,  $|V_{\{2\}}| = l$ , and  $|V_{\{1,2\}}| = |V_{\{1\}}| = l + 1$ . An optimal solution requires the activation of interface 2 for all nodes from  $V_{\{2\}}$  and  $V_{\{1,2\}}$ , and interface 1 for all nodes from  $V_{\{1\}}$  and exactly one adjacent node from  $V_{\{1,2\}}$ , hence its cost is  $3l + 3$ . The considered approximation algorithm will activate all the available interfaces, except for the rightmost node from  $V_{\{1,2\}}$ , obtaining a solution with cost  $4l + 2$ . Thus the obtained approximation ratio is arbitrarily close to  $\frac{4}{3}$ .

### 2.3 Particular Graph Topologies

In this section we show how *Connectivity* behaves with respect to particular graph classes such as graphs of bounded treewidth and complete graphs.

**Graphs of Bounded Treewidth.** When the input graph  $G$  is a tree, connections need to be established along all edges of  $G$ , hence *Connectivity* is equivalent to the Cost Minimisation in Multi-Interface Networks problem previously studied in [12]. Since such a problem is known to be hard even for trees, *Connectivity* also remains *APX*-hard for trees when the value of  $k$  is unbounded, even in the unit cost interface case.

When  $k$  is bounded, the locality of the *Connectivity* optimisation criterion makes it possible to apply a dynamic programming technique described by Bodlaender [3] to solve *Connectivity* optimally and in polynomial time for the class of graphs with bounded treewidth, which includes trees, outerplanar graphs, and series-parallel graphs. (For the details of formalising the connectivity condition in such an approach, cf. e.g. the LCC-formulation for the *Degree-bounded Connected Subgraph* problem in [3].)

**Corollary 2.** *For any constant  $t \in \mathbb{N}^+$ , *Connectivity* can be optimally solved in polynomial time for graphs of treewidth  $t$ .*

In particular, when  $t = 1$  (i.e. graph  $G$  is a tree), such an approach provides an optimal solution to bounded *Connectivity* in  $O(n)$  time.

**Complete Graphs.** When the input graph  $G$  is complete, clearly the 2-approximation in general and the  $(2 - \frac{1}{k})$ -approximation for the unit cost interface case still hold, but we can provide a  $(C(1 + \frac{k}{n}))$ -approximation algorithm with  $C$  being the ratio between the costs  $c_{max}$  and  $c_{min}$  of the most expensive and the cheapest available interfaces among  $G$ , respectively. In the unit cost interface case,  $C = 1$ . The algorithm simply activates all the edges from a chosen source to all the other nodes by means of the cheapest available interfaces for each edge. By definition, such connections are available, i.e., all the nodes share at least one interface with the source. This is clearly a solution to *Connectivity* since it induces a star graph from the source to all the other nodes of  $G$ . The sought approximation ratio guaranteed by this algorithm is shown by observing that, in general, in the constructed solution we might activate up to  $n + k$  interfaces of cost  $c_{max}$ , whereas the optimum solution might find a spanning tree by activating  $n$  interfaces of cost  $c_{min}$ . It follows that:

$$\frac{(n + k)c_{max}}{nc_{min}} < C \left(1 + \frac{k}{n}\right).$$

For complete graphs, *Connectivity* remains *APX*-hard when  $k$  is unbounded even in the unit cost interface case. This is immediately clear, since a unit-cost instance of *Connectivity* in an arbitrary graph  $G$  of order  $n$  can be reduced to an instance of *Connectivity* in the complete graph  $K_n$ : for each pair of nodes  $u, v$  which are not connected by an edge in  $G$ , we add a new interface, available only for  $u$  and  $v$ . Any activation of interfaces in  $G$  is also

a valid activation for the modified instance in  $K_n$ ; conversely, there does not exist in  $K_n$  an activation of smaller cost than the one in  $G$ . Note that the assumption about unbounded  $k$  is relevant (since we add a new interface number for potentially  $\Theta(n^2)$  edges); in fact, for bounded  $k$  the *Connectivity* problem in complete graphs can be shown to admit an  $O(n^2)$ -time solution, similar to that presented for the *Coverage* problem considered in [12].

### 3 Cheapest Paths in Multi-Interface Networks

In this section, we present and study the *Cheapest Path* problem.

#### 3.1 Definitions and Notation

As we considered in the previous section for defining *Connectivity*, let  $G = (V, E)$  be the network graph always assumed to be simple, undirected and connected. The set of all the available interfaces in  $v$  is denoted by  $W(v)$ , and the cardinality of all the available and distinct interfaces in  $G$  by  $k$ .

For each node  $v \in V$ , let  $W_A(v)$  be the set of switched on (activated) interfaces. Clearly,  $W_A(v) \subseteq W(v)$  where  $W(v)$  is the same function of Definition 1.

Again, the cost of activating an interface for a node is assumed to be identical for all nodes and given by cost function  $c: \{1, \dots, k\} \rightarrow \mathbb{N}$ . The cost of interface  $i$  is denoted as  $c_i$ .

A path  $P$  in  $G$  from a given source node  $s$  to a target node  $v$  is denoted by a sequence of couples: for each node  $v_j \in P$ , besides node  $v_j$  itself, the interface  $i_j$  used to reach  $v_j$  is given. Namely,  $i_j \in W_A(v_j)$ . For example, the sequence  $P = \langle (s \equiv v_0, 0), (v_1, i_1), \dots, (v \equiv v_t, i_t) \rangle$ , denotes a path  $P$  from  $s$  to  $v$  that moves on the nodes  $s, v_1, \dots, v_{t-1}, v$  and that reaches node  $v_j$  via interface  $i_j$ , for  $1 \leq j \leq t$ . Interface 0 is used to denote “no interface” since the source is not reached by any other node in  $P$ . Conventionally,  $c_0 = +\infty$ . However, the source needs to activate interface  $i_1$  in order to reach  $v_1$ , hence the cost of activating the edge  $(s, v_1)$  is  $2c_1$ . In general, the cost for activating the path  $P$  is

$$d_P(v) = \sum_{j=1}^t \text{cost}((v_{j-1}, i_{j-1}), (v_j, i_j))$$

where

$$\text{cost}((v_{j-1}, i_{j-1}), (v_j, i_j)) = \begin{cases} 2c_{i_j} & \text{if } i_{j-1} \neq i_j \\ c_{i_j} & \text{otherwise} \end{cases}$$

Let  $\delta(v)$  be the minimum cost to activate a path from the source node  $s$  to node  $v$ , that is,  $\delta(v) = \min\{d_P(v): P \text{ is any path from } s \text{ to } v\}$ . In addition, let the *cheapest path*  $CP_v$  from the source  $s$  to  $v$  be any path  $P$  from  $s$  to  $v$  such that  $d_P(v) = \delta(v)$ . An  $i$ -path  $P$  from  $s$  to  $v$  is a path from  $s$  to  $v$  that reaches  $v$  via interface  $i$ . Let  $d_P(v, i)$  denote the cost of the  $i$ -path  $P$ , whereas  $\delta(v, i)$  denotes the minimum cost among all the  $i$ -paths from  $s$  to  $v$ . Besides, let the *cheapest  $i$ -path*  $CP_{v,i}$  from the source node  $s$  to node  $v$  be any  $i$ -path  $P$  such that  $d_P(v, i) = \delta(v, i)$ . Clearly,  $\delta(v) = \min\{\delta(v, i): i \in W(v)\}$ . Whenever clear by the context we remove  $P$  from the notation  $d_P$ .

We study the usually called Shortest Path problem but in the context of multi-interface networks. Actually, in these networks, dealing with shortest paths is neither of practical nor of theoretical interest. In fact, as shown above, the cost of an edge is not set up a priori as in the standard problem, but depends on the activated interfaces at its endpoint. The *Cheapest Path* (*CP* for short) problem can be formulated as follows.

---

**Cheapest Path (CP)**

---

**Input:** A graph  $G = (V, E)$ , an allocation of available interfaces  $W: V \rightarrow 2^{\{1, \dots, k\}}$  covering graph  $G$ , an interface cost function  $c: \{1, \dots, k\} \rightarrow \mathbb{R}_+$  and a source node  $s \in V$ .

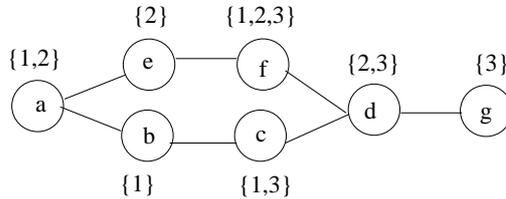
**Solution:** A set of  $n - 1$  paths, one for each node but  $s$ . For each node  $v \in V \setminus \{s\}$ , a path  $P$  from  $s$  to  $v$  must be specified by a sequence of couples of the form  $(v_j, i_j)$ , with  $v_0 = s$ ,  $i_0 = 0$ ,  $v_t = v$  and  $v_j \in V$ ,  $i_j \in W(v_j)$  for  $1 \leq j \leq t$  with the meaning that node  $v_j$  is reached by means of interface  $i_j$ .

**Goal:** For each node  $v \in V$ , find  $\delta(v)$  along with a cheapest path  $CP_v$ .

---

### 3.2 A Polynomial Time Algorithm

In order to better understand differences with the standard shortest path problem, let us consider the simple network of Figure 3, and assume we want to solve the cheapest path problem from source node  $a$  in the graph  $G$ , in a setting where the costs of interfaces 1, 2 and 3 are 1.5, 1.5 and 1, respectively. We find that  $\delta(d) = 6$  and  $CP_d = \langle (a, 0), (e, 2), (f, 2), (d, 2) \rangle$ ,



**Fig. 3.** A sample network. In brackets for each node there are the corresponding available interfaces.

whereas  $\delta(g) = \frac{15}{2}$  and  $CP_g = \langle (a, 0), (b, 1), (c, 1), (d, 3), (g, 3) \rangle$ . However the subpath of  $CP_g$  from  $a$  to  $d$ , that is  $\langle (a, 0), (b, 1), (c, 1), (d, 3) \rangle$ , has cost  $\frac{13}{2} > \delta(d)$ . Therefore, it turns out, that the main property of the standard shortest path problem does not hold. Precisely:

**Proposition 1.** *A subpath of a cheapest path is not necessarily a cheapest path itself.*

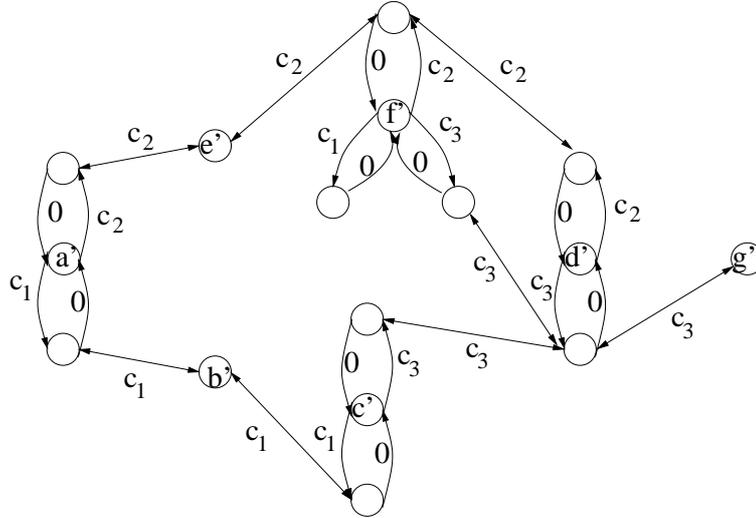
While the sub-optimality property does not hold in general, it holds when we consider a subpath of a cheapest path characterised not only by its final node, but also by the interface used in its last hop.

**Lemma 1.** *Given a graph  $G = (V, E)$  and a source node  $s \in V$ , let  $CP_v$  be a cheapest path from  $s$  to  $v$  that passes through node  $u$  and reaches  $u$  via interface  $i$ . Then, the  $i$ -subpath of  $CP_v$  from  $s$  to  $u$  is a cheapest  $i$ -path.*

*Proof.* Let  $P$  denote the  $i$ -subpath of  $CP_v$  from  $s$  to  $u$ . By contradiction, suppose that  $P$  is not a cheapest  $i$ -path, and hence  $d_P(u, i) > \delta(u, i)$ . Since a cheapest path  $CP_{u,i}$  reaches  $u$  via interface  $i$ , replacing  $P$  with  $CP_{u,i}$  in  $CP_v$ , a new path  $P'$  from  $s$  to  $v$  with cost  $d_{P'}(v) = \delta(v) - (d_P(u, i) - \delta(u, i)) < \delta(v)$  is found, contradicting the definition of cheapest path.  $\square$

As a consequence, the cost of a cheapest path from  $s$  to  $v$  can be easily determined when the endpoints of the final edge of such a path and the interface used to reach  $v$  are given.

We can thus solve *Cheapest Path* on a graph  $G = (V, E)$  by making use of the standard Dijkstra's algorithm [7] on a slightly modified input instance  $\vec{G}$ . The directed weighted graph  $\vec{G} = (V', E')$  with  $n' = |V'|$  and  $m' = |E'|$  is obtained from  $G$  as follows. Each node  $v \in V$  is replaced by  $|W(v)| + 1$  nodes: one node for each interface available at node  $v$ ,  $\{v_i : i \in W(v)\}$ , and one extra node  $v'$ . We connect these nodes into a star with center  $v'$  as follows: for all  $i \in W(v)$ , edges  $(v_i, v')$  have weight 0, whereas edges  $(v', v_i)$  have weight  $c_i$ . Moreover, each edge  $\{u, v\} \in E$  is replaced by  $|W(u) \cap W(v)|$  pairs of edges: for each  $i \in W(u) \cap W(v)$ , we define a new pair of arcs  $(u_i, v_i)$  and  $(v_i, u_i)$ , each of weight  $c_i$ . An example of the performed transformation is shown in Fig. 4. Solving the *Cheapest Path*



**Fig. 4.** The directed graph obtained from the graph of Figure 3 for applying Dijkstra's algorithm.

problem with source  $s$  on  $G$  is equivalent to solve the shortest path tree problem on  $\vec{G}$  with source  $s'$ , and the cost of the path from the source to a generic node  $v \in V$  in  $G$  is equal to  $d_P(v)$  in  $\vec{G}$ .

Hence, Dijkstra’s algorithm can be applied on  $\vec{G}$ , achieving a time complexity of  $O(m' + n' \log n') = O(km + kn \log(kn))$ . For the unit cost interface case, each of the arcs of  $\vec{G}$  has a weight of either 0 or 1, and thus the time complexity reduces to  $O(m' + n') = O(k(m + n))$ .

## 4 Conclusion

We have considered two basic problems in the context of multi-interface networks. Namely, the well-known minimum spanning tree determination has become the *Connectivity* problem, and the well-known shortest path tree determination has become the *Cheapest Path* problem. For *Connectivity* we have shown complexity results in both general and restricted settings. For most network topologies, *Connectivity* turns out to have fundamentally different approximability and hardness characteristics than previously studied problems, such as the Coverage problem which activates all possible connections in the network graph. Concerning *Cheapest Path*, we have shown that the problem is polynomially solvable even though some minor investigations were required. We have provided a suitable reduction for any input graph which allows us to make use of the standard Dijkstra’s algorithm in order to solve the problem.

The obtained results give new insight in the context of multi-interface networks. Many other interesting problems related to multi-interface networks remain unexplored. The study of other standard optimisation problems in this context is challenging for future work. Another interesting direction would be to explore such problems in a distributed setting.

## Acknowledgement

The authors would like to thank the anonymous referees for many insightful comments.

## References

1. P. Bahl, A. Adya, J. Padhye, and A. Walman. Reconsidering wireless systems with multiple radios. *SIGCOMM Comput. Commun. Rev.*, 34(5):39–46, 2004.
2. F. Barsi, A. Navarra, and M.C. Pinotti. Cheapest paths in multi-interface networks. In *Proceedings of the 10th International Conference on Distributed Computing and Networking (ICDCN)*, Lecture Notes in Computer Science. Springer-Verlag, 2009, to appear.
3. H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In *Proceedings of the 15th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 317 of LNCS, pages 105–118, 1988.
4. R. L. Brooks. On coloring the nodes of a network. *Proceedings of Cambridge Philosophical Society*, 37:194–197, 1941.
5. M. Caporuscio, D. Charlet, V. Issarny, and A. Navarra. Energetic Performance of Service-oriented Multi-radio Networks: Issues and Perspectives. In *Proceedings of the 6th International 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 *Proceedings of the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1322–1326. IEEE, 2005.
7. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, 2001.

8. R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom)*, pages 114–128. ACM, 2004.
9. A. Faragó and S. Basagni. The effect of multi-radio nodes on network connectivity—a graph theoretic analysis. In *Proceedings of the IEEE International Workshop on Wireless Distributed Networks (WDM)*. IEEE, 2008.
10. 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.
11. M. R. Garey, David S. Johnson, and Robert Endre Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5(4):704–714, 1976.
12. R. Klasing, A. Kosowski, and A. Navarra. Cost minimisation in wireless networks with bounded and unbounded number of interfaces. *to appear in Networks*.
13. R. Klasing, A. Kosowski, and A. Navarra. Cost minimisation in multi-interface networks. In *Proceedings of the 1st EuroFGI International Conference on Network Control and Optimization (NET-COOP)*, volume 4465 of *Lecture Notes in Computer Science*, pages 276–285. Springer-Verlag, 2007.
14. A. Kosowski and A. Navarra. Cost minimisation in unbounded multi-interface networks. In *Proceedings of the 2nd PPAM Workshop on Scheduling for Parallel Computing (SPC)*, volume 4967 of *Lecture Notes in Computer Science*, pages 1039–1047. Springer-Verlag, 2007.
15. A. Kosowski, A. Navarra, and M.C. Pinotti. Connectivity in Multi-Interface Networks. In *Proceedings of the 4th International Symposium on Trustworthy Global Computing (TGC)*, volume 5474 of *Lecture Notes in Computer Science*, pages 157–170. Springer-Verlag, 2008.