Sharing the Cost of Multicast Transmissions in Wireless Networks

Paolo Penna and Carmine Ventre

Dipartimento di Informatica ed Applicazioni
"R.M. Capocelli"
Università di Salerno
via ***S. Allende*** *2*
I-84081 Baronissi (SA)
Italy

Abstract

We investigate the problem of sharing the cost of a multicast transmission in a wireless network in which each node (i.e., radio station) of the network corresponds to a (set of) user(s) potentially interested in receiving the transmission. As in the model considered by Feigenbaum *et al.* [2001], users may act *selfishly* and report a false "level of interest" in receiving the transmission trying to be charged less by the system. We consider the issue of designing so called *truthful mechanisms* for the problem of maximizing the *net worth* (i.e., the overall "satisfaction" of the users minus the cost of the transmission) for the case of *wireless* networks. Intuitively, truthful mechanisms guarantee that no user has an incentive in reporting a false valuation of the transmission. Unlike the "wired" network case, here the cost of a set of connections implementing a multicast tree is *not* the sum of the single edge costs, thus introducing a complicating factor in the problem. We provide both positive and negative results on the existence of optimal algorithms for the problem and their use to obtain VCG truthful mechanisms achieving the same performances.

Key words: Algorithmic mechanism design, Wireless networks, Energy consumption, Multicast trees, Cost sharing, Selfish agents.

Email addresses: penna@dia.unisa.it (Paolo Penna), ventre@dia.unisa.it (Carmine Ventre).
URLs: www.dia.unisa.it/∼penna (Paolo Penna), www.dia.unisa.it/∼ventre (Carmine Ventre).

1 Introduction

In many practical situations, one is given a communication network and a special node called the *source* that, potentially, can broadcast messages to any other node in the network. Whenever the source has to transmit to a certain subset of nodes, a suitable *multicast tree* is computed so to reduce the *overall cost* of transmitting (e.g., there is a cost for using each link and the cost of a multicast tree is the sum of the costs of its links). In several applications, the source can "offer" some kind of transmission (say a movie or a sport event) to users located at other nodes of the network. A user located at node j can receive the transmission only if that node does and, naturally, the larger the set of receivers the higher the costs for transmitting. Hence, one would like to find a good trade-off between transmitting costs and the overall users "satisfaction". Each user j may have her own *valuation* of the transmission which could be quantified according to some value v_j (i.e., how worth is the transmission for user j). If the source transmits to a subset S of users and this costs C, then the resulting *net worth* is equal to

$$\left(\sum_{j \in S} v_j\right) - C.$$

Intuitively speaking, the summation represents the *worth* associated to the subset S, i.e., the overall satisfaction obtained if users in S receive the transmission. The net worth is computed by subtracting the overall cost for transmitting. In the context of multicast transmission described above, each multicast tree determines a net worth in which S is the set of users located in some node of the tree and C is the cost of the tree (see below for formal definitions). Finding a multicast tree which maximizes the net worth can be difficult problem because of the following:

(1) A tree of minimal cost could be NP-hard to compute. If the cost of a tree is the sum of the costs of its links, then we would have to optimally solve the well-known Steiner tree problem which, on general networks, is NP-hard.
(2) The values v_j are not known to the source and they will have to be communicated by the users. Users j may act *selfishly* and lie about her v_j so to be included in the tree: if the underlying algorithm optimizes the net worth, then user j will be included whenever reporting a "very high" value b_j.

Feigenbaum *et al.* [13] considered a *game theoretic* approach in which users are charged for receiving the transmission. Whether a user j receives the transmission and at which price depends on her *reported* valuation b_j (and also on those reported by the other users). The solution by Feigenbaum *et al.* [13] has two interesting features:

(1) No user has an incentive in misreporting her valuation (i.e., the case $b_j \neq v_j$);
(2) The underlying algorithm returns a tree of maximal net worth if all users report their valuations correctly (i.e., $b_j = v_j$ for every user j).

The work by Feigenbaum *et al.* [13] is tailored for a "wired" network model in which the cost of a multicast tree is the sum of the costs (i.e., weights) of all of its links.

The main goal of this work is to investigate the scenario in which transmissions are performed over *wireless (ad-hoc)* networks. This type of networks is particularly attractive because of the possibility of communicating without any fixed infrastructure: Messages are sent by one node to another one by transmitting radio signals with a "sufficiently high" power. Implementing a multicast tree over a wireless network costs proportionally to the overall energy that must be spent by the stations for implementing the links in the tree, that is, as the *overall power consumption* of the network. Since energy is crucial in wireless networks, this turns out to be an important and well-studied metric.

Unfortunately, the construction of optimal (i.e., minimal cost) multicast trees for wireless networks departs significantly from that of the "wired" counterpart, i.e., the model used by Feigenbaum *et al.* [13]. Hence, the results in [13] do not apply to our model. In this work, we prove a number of new positive and negative results for the wireless network model which parallel with those in [13]. Our model, which we describe in detail in the remaining of this section, accounts for the same basic requirements of the model by Feigenbaum *et al.* [13].

Cost of Wireless Links. In a wireless (ad-hoc) network, each node is a radio transmitter/receiver also called *station.* Station i is able to *directly* transmit a message to station j if and only if the power $\mathcal{P}_i$ used by station i satisfies

$$\frac{\mathcal{P}_i}{d(i,j)^\alpha} \geq \gamma,$$

where $d(i,j)$ is the Euclidean distance between i and j, $\alpha \geq 1$ is the *attenuation parameter* depending on the environmental conditions [25] (e.g., in the empty space $\alpha = 2$), and γ is the *transmission quality parameter*: station j will receive the message with attenuated power equal to $\mathcal{P}' = \mathcal{P}_i/d(i,j)^\alpha$, and the message can be correctly interpreted if $\mathcal{P}' \geq \gamma$. We let $w(i,j)$ be the minimal power required to establish the direct link (i,j), that is, $w(i,j) := \gamma \cdot d(i,j)^\alpha$. (Typically γ is normalized to be 1.)

In general, because of different environmental conditions occurring at different places (e.g., an obstacle between two stations), there can be a different attenuation parameters α_{ij} for every pair (i,j). Each station can adjust its transmission power $\mathcal{P}_i$ to any value not larger than its *maximum transmission power* $\mathcal{P}_i^{\max}$, which depends on the (limited) battery capacity of station i. To model these aspects, we need to consider a directed weighted graph consisting of all possible wireless links that stations can implement, given their maximum transmission powers and the environmental conditions, along with the power required to implement each of these links. More precisely, we have a *communication (directed) graph* $\mathcal{G} = (\mathcal{S}, \mathcal{E}, w)$ where

- $\mathcal{S}$ is the set of stations and $|\mathcal{S}| = n$.
- $\mathcal{E}$ contains all possible wireless links: $(i,j) \in \mathcal{E}$ if and only if the maximum power of station i suffices for transmitting to station j, that is, $\mathcal{P}_i^{\max}/d(i,j)^{\alpha_{ij}} \geq \gamma$.
- $w(i,j)$ is the minimum power required for station i to directly transmit to station j, that is, the weight of link (i,j) satisfies $w(i,j) = \gamma \cdot d(i,j)^{\alpha_{ij}}$.

Stations use so called *omnidirectional antennae* which radiate in all directions. That is, if station i transmits with power $\mathcal{P}_i$, then *every* station j for which $w(i,j) \leq \mathcal{P}_i$ receives this transmission. In other words, the minimal power that station i must spend for implementing a set E_i of links directed to other stations is

$$\max_{(i,j)\in E_i} w(i,j).$$

Given any set of links $E \subseteq \mathcal{E}$, we define its cost as the *overall power consumption* required to implement all these links, that is,

$$Cost(E) := \sum_{i\in\mathcal{S}} \max_{j:(i,j)\in E} w(i,j). \qquad (1)$$

Observe that we do not sum up all edge weights in E. In terms of graphs, the cost of any $E \subseteq \mathcal{E}$ is the sum, over all nodes i, of the maximum edge weight among all edges in E that are directed from i to some other node.

Figure 1 shows an example of a wireless network in the two-dimensional Euclidean space with attenuation parameter α constant everywhere. Disks represent the regions where the attenuated power of the corresponding signal is not smaller than γ: station s transmits with power $d(s,3)^\alpha$ and reaches stations 1,

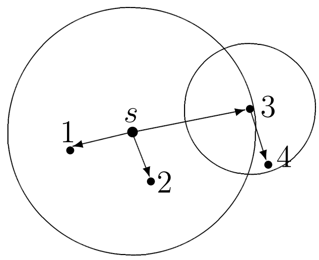

Fig. 1. An example of wireless network in the two-dimensional Euclidean space with constant attenuation parameter.

2 and 3, while transmission from s to 4 is performed in two hops via station 3.

Net Worth of a Multicast Transmission. In this work we are interested in sets $E \subseteq \mathcal{E}$ which guarantee that, given a distinguished *source node* s, the set E connects s to a suitable set $D(E, \langle \mathcal{G}, s \rangle) \subseteq \mathcal{S}$ of *destination nodes*, that is, there is a directed path from s to any node in $D(E, \langle \mathcal{G}, s \rangle)$ consisting of edges of E. (For the sake of readability, from now on $D(E, \langle \mathcal{G}, s \rangle)$ will be denoted simply as $D(E)$ as the input $\langle \mathcal{G}, s \rangle$ will be clear from the context.)

Each user is located close to some of the nodes in $\mathcal{S}$. The source s can send the transmission to a user j only if j is close to some of the destination nodes $D(E)$. In addition, every user j has a *valuation* v_j of the transmission representing how much she would benefit from receiving it (i.e., how much she would pay for it). As in the model of [13], we consider the situation in which each user j is sitting close to one station, say i; the latter represents the router of the network at distance one hop from user j. So, user j can receive the transmission only if node i does. Observe that, we can always reduce the case of several users located close to the same node to the case of (at most) one user close to one node (consider each user as a node with no outgoing edges and one ingoing edge of cost 0). The latter models the situation of ad-hoc networks in which every user is the owner of one node of the network. Given a subset E of edges, its *worth* is the sum of the v_is of the nodes/users receiving the transmission:

$$Worth(E, \mathbf{v}) := \sum_{i \in D(E)} v_i, \tag{2}$$

where $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is the vector of the agents' valuations. Then, the *net worth* is given by

$$NetWorth(E, \mathbf{v}) := Worth(E, \mathbf{v}) - Cost(E). \tag{3}$$

The *cost sharing problem* asks for a $E \subseteq \mathcal{E}$ that, for a given source s, maximizes the net worth function above. Observe that we can always assume E to form a

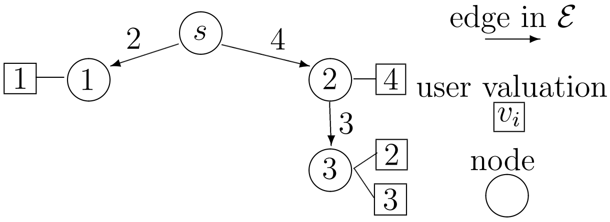

Fig. 2. An instance whose communication graph is a tree.

tree connecting s to a subset of the node (otherwise, we can find in polynomial time a subset $T \subset E$ which forms a multicast tree with the same worth and a non-larger cost). In the sequel we will thus consider algorithms which return subsets T of edges which form a multicast tree. For convenience, we will denote any multicast tree as the set of its edges.

In Figure 2, we show an example where the optimal net worth solution for the wired case is different from the wireless one, even when the communication graph $\mathcal{G}$ is a tree. The optimal solution for wireless networks is $T = \{(s, 1), (s, 2), (2, 3)\}$. The user at node 2 would receive the transmission also when reporting '3' (instead of her true valuation 4) to an optimal algorithm. If all connections would be payed, as for the "wired" case, then edge $(s, 1)$ should not be included since the user at node 1 cannot recover for its cost.

Observe that, we will always produce some solution T for which

$$NetWorth(T, \mathbf{v}) \geq 0,$$

since otherwise it is not worth to perform the transmission. We also would like to charge some amount of money to the receivers so to recover the cost of the transmission, unless the solution consists of the empty tree which does not connect the source to any node (by definition, this solution yields a non-negative net worth). More importantly, payments should provide an incentive for the users to report their valuations v_i correctly since this is essential for optimizing the net worth (see Equations 2-3).

Selfish Users and Economical Constraints. The users, knowing that they will be charged for receiving the transmission, may act *selfishly* and report false values trying to get the transmission at lower prices. [1] In particular, associated to each node there is a *selfish agent* reporting some (not necessarily true) valuation $b_i \geq 0$; the true valuation v_i is non-negative and is *privately known* to agent i. Based on the reported values $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ a *mechanism* $M = (A, P)$ uses algorithm A in order to construct a multicast tree,

[1] Interestingly, users may also lie if the transmission would be given for free to a subset of them. Indeed, each user i would try to get it by declaring a very high interest b_i.

i.e., $T = A(\mathbf{b})$ and charges, to each agent i, an amount of money to pay for receiving the transmission equal to $P^i(\mathbf{b})$, with $P = (P^1, P^2, \ldots, P^n)$. Notice that, for every agent i, both the computed solution and the amount of money charged to this agent depend on her reported valuation b_i. To stress this dependency, throughout the paper we use the following standard notation in mechanism design: For every agent i, we let

$$\mathbf{b}_{-i} := (b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n)$$

denote the vector of the valuations reported by all agents but i and

$$(b_i, \mathbf{b}_{-i}) := (b_1, b_2, \ldots, b_{i-1}, b_i, b_{i+1}, \ldots, b_n) = \mathbf{b}.$$

There is a number of natural constraints/goals that we would like a mechanism $M = (A, P)$ to satisfy/meet:

(1) **Truthfulness (or Strategyproofness).** [2] The *utility* of agent i when she reports b_i, the other agents report $\mathbf{b}_{-i}$, and her valuation is v_i, is equal to

$$u_i(b_i, \mathbf{b}_{-i}|v_i) := \begin{cases} v_i - P^i(b_i, \mathbf{b}_{-i}) & \text{if } T = A(b_i, \mathbf{b}_{-i}) \text{ and } i \in D(T), \\ 0 & \text{otherwise.} \end{cases}$$

We require that, for every $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, for every i, for every $\mathbf{b}_{-i}$, and for every b_i, it holds that

$$u_i(v_i, \mathbf{b}_{-i}|v_i) \geq u_i(b_i, \mathbf{b}_{-i}|v_i).$$

In other words, whatever strategy the other agents follow, agent i has no incentive to lie about her true valuation v_i. A mechanism satisfying this property is called *truthful*.

(2) **Efficiency.** The computed solution $T = A(\mathbf{b}) \subseteq \mathcal{E}$ maximizes the *net worth* relatively to the *reported* valuations. That is,

$$NetWorth(T, \mathbf{b}) = \max_{E \subseteq \mathcal{E}} \{NetWorth(E, \mathbf{b})\}.$$

Therefore, if all agents are truthtelling, i.e., $\mathbf{b} = \mathbf{v}$, then the computed solution optimizes the net worth with respect to the true valuations.

(3) **No Positive Transfer (NPT).** No user receives money from the mechanism, that is, for every i, for every b_i, and for every $\mathbf{b}_{-i}$, it holds that $P^i(b_i, \mathbf{b}_{-i}) \geq 0$.

[2] In Section 2.1 we provide a more general definition of truthfulness which applies to a wide class of problems involving selfish agents.

(4) **Voluntary Participation (VP).** We never charge a user an amount of money greater than her *reported* valuation, that is, for every i, for every b_i, and for every $\mathbf{b}_{-i}$, it holds that $P^i(b_i, \mathbf{b}_{-i}) \leq b_i$. In particular, a user has always the option on not paying for a message for which she is not interested in.

(5) **Consumer Sovereignty (CS).** Every user is guaranteed to receive the message if she reports a high enough valuation for the transmission.

(6) **Budget Balance (BB).** $\sum_i P^i(\mathbf{b}) = Cost(A(\mathbf{b}))$, i.e., the cost of broadcasting the message is recovered from all users and no surplus is created.

(7) **Cost Optimality (CO).** The computed solution $T = A(\mathbf{b}) \subseteq \mathcal{E}$ is optimal with respect to the set of receivers $D(T)$, that is,

$$Cost(T) = \min_{E \subseteq \mathcal{E}, D(T)=D(E)} \{Cost(E)\}.$$

In other words, there is no cheaper solution (i.e., of smaller cost) which yields the same set of receivers of the solution $T = A(\mathbf{b})$.

It is important to observe that a mechanism $M = (A, P)$ receives in input a vector $\mathbf{b}$ of reported valuations and it is *not* aware of the true valuations $\mathbf{v}$. Truthfulness guarantees that all agents will report their true valuations (i.e., $\mathbf{b} = \mathbf{v}$). Then, the Efficiency requirement implies that the mechanism returns a solution maximizing the net worth with respect to the true valuations.

Notice that, the Efficiency requirement implies the Cost Optimality: If T is not cost optimal, then there exits $E \subseteq \mathcal{E}$ such that $Cost(E) < Cost(T)$ and $D(E) = D(T)$. Clearly, this implies $Worth(E, \mathbf{b}) = Worth(T, \mathbf{b})$ and thus $NetWorth(E, \mathbf{b}) < NetWorth(T, \mathbf{b})$.

Unfortunately, in some cases it is impossible to achieve efficiency, so we will relax it to r*-efficiency*, that is,

$$r \cdot NetWorth(T, \mathbf{b}) \geq \max_{E \subseteq \mathcal{E}} \{NetWorth(E, \mathbf{b})\},$$

where $r \geq 1$. Similarly, we will relax CO to r-CO, that is,

$$Cost(T) \leq r \cdot \min_{E \subseteq \mathcal{E}, D(T)=D(E)} \{Cost(E)\}.$$

1.1 Previous Work

Power Consumption and Range Assignment Problems. Assigning transmission powers to the stations which (i) guarantee a "good" communication between the stations, and (ii) minimize the overall power consumption of the network gives rise to interesting algorithmic questions. In particular, [17,11,7,31] address the issue of computing a broadcast tree of minimal cost. Although this problem, in the case of "wired" networks, is clearly equivalent to the problem of computing an MST (all edges are counted in the cost function), in the case of wireless networks things are more complicated. Indeed, [17] proved that the problem is NP-hard to approximate within logarithmic factors, while it remains NP-hard even when considering geometric 2-dimensional networks [7]. In [21,10,11,7,31] other variants of this problem have been considered (see also [8] for a survey). However, to the best of our knowledge, no algorithmic solution for optimizing the net worth has been given so far.

Recently, the design of truthful mechanisms for the range assignment problem in presence of "selfish transmitters" (i.e., selfish agents that want to minimize the energy their station has to use) has been investigated in [1] for the strongly connectivity problem, and in [3] for point-to-point transmissions, respectively.

Mechanism Design and Cost-Sharing Mechanisms in Wired Networks. The theory of mechanism design dates back to the seminal papers by Vickrey [30], Clarke [6] and Groves [16]. Their celebrated *VCG mechanism* is still the prominent technique to derive truthful mechanisms for many problems (e.g., shortest path, minimum spanning tree, etc.). In particular, when applied to combinatorial optimization problems (see e.g., [23,29]), the VCG mechanisms guarantee the truthfulness under the hypothesis that the mechanism is able to compute the optimum and the optimization function is *utilitarian* (see Section 2.1 for a formal definition of utilitarian problem).

This technique is employed in [13] (and in this work) where the authors consider the wired networks case: in this case, *every* connection of a solution $E \subseteq \mathcal{E}$ contributes to the corresponding cost $Cost_{wrd}(E)$, that is

$$Cost_{wrd}(E) = \sum_{(i,j) \in E} w(i,j).$$

They indeed provide a *distributed* optimal algorithm for the case in which the communication graph is a directed tree. This yields a *distributed mechanism* [3] which, for this problem version, satisfies all requirements mentioned above (truthfulness, efficiency, etc.) except for budget balance.

Noticeably, a classical result in game theory [15,28] implies that, for this model, budget balance and efficiency are mutually exclusive. Additionally, in [12] (see also Theorem 5 in [5]) it is shown that no α-efficiency and β-efficiency can be guaranteed simultaneously, for any two $\alpha, \beta > 1$. So, the choice is to either

[3] The mechanism is able to compute both the solution and the payments in distributed fashion.

optimize the efficiency (as in [13]) or to meet budget balance (as in [18,19,5]). In the latter case, it is also possible to obtain so called *group strategyproofness*, a stronger notion of truthfulness which can also deal with *coalitions*. On the other hand, if we insist on efficiency, NPT, VP, and CS, then there is essentially only one such mechanism: the marginal-cost mechanism [22], which belongs to the VCG family.

Also notice that all such negative results also apply to our problem (i.e., wireless networks). Indeed, a simple observation is that every instance of the "wired" case can be reduced to the wireless one using the following trick: replace every edge (i, j), with two edges $(i, x(i, j))$ and $(x(i, j), j)$ whose weights are $w(i, x(i, j)) = 0$ and $w(x(i, j), j) = w(i, j)$. So, also for our problem we have to choose between either budget balance or efficiency.

1.2 Our Results

We consider the problem of designing mechanisms that satisfy truthfulness, efficiency, NPT, VP, CS, and CO in the case of wireless networks. We first show that, even though the problem is not utilitarian, it is possible to adapt the VCG technique so to obtain truthful mechanisms based on exact algorithms (Section 2). We stress that our result is constructive. Indeed, we define a (polynomial-time computable) payment scheme leading to a truthful mechanism with a given exact (polynomial-time) algorithm. Hence, if the algorithm runs in polynomial time, then the entire mechanism is polynomial time as well (see Theorem 3).

Unfortunately, exact algorithms may be hard to find as the underlying optimization problem turns out to be NP-hard in several cases (Section 3). We indeed show that no polynomial-time algorithm can guarantee r-efficiency, for any $r > 0$, unless $\mathsf{P} = \mathsf{NP}$. This result, which holds even if assuming the communication graph to be a layered graph (see Theorem 5), rules out the possibility of polynomial-time mechanisms with a "reasonable" efficiency in general (e.g., $O(n)$-efficiency is impossible for layered communication graphs). The use of VCG mechanisms requires exact algorithms (i.e., 1-efficiency) which cannot be obtained even for geometric Euclidean instances in which stations are locates on the plane (and in general on the ℓ-dimensional Euclidean space with $\ell \geq 2$ – see Theorem 6).

The positive results on the construction of truthful mechanisms and the hardness results motivate the study of interesting topologies for which exact polynomial-time algorithms exist (Section 4). We first consider the problem restricted to communication graphs that are trees and prove that, in this case, the optimal net worth can be computed via an $O(n)$-*time distributed* algorithm (Section 4.1). This is the analogous of the result for wired networks in [13] and its importance is threefold:

- It shows that the difficulty of the problem is confined in the choice of a "good" multicast tree, and not in its use: if an "oracle" provides us with a tree containing an optimal multicast tree, then we can compute the optimum in polynomial-time. This result cannot be obtained from the analogous result for wired networks in [13] since, in general, the optimal solutions differ in the two cases (see Figure 2).
- It is used to obtain a truthful *distributed* $O(n)$*-time* mechanism satisfying NPT, VP, CS, and efficiency when the given communication graph is a tree. In this case, both the solution and the payments can be computed in distributed fashion using *two messages per link* (Corollary 10). Again, the resulting payments are different from the once in [13] and, thus, the algorithm departs significantly from the one in [13].
- It can be used to approximate the problem in some situations for which a good "universal" tree exists, i.e., a tree containing a set of connections of cost not much larger than the optimal solution and reaching the same set of nodes. This approach is similar to that of several wireless multicast protocols [4] which construct a multicast tree by pruning a broadcast tree $\mathcal{T}$ (e.g., MIP, MLU and MLiMST in [11]). In all such cases, one can assume that the communication graph $\mathcal{G}$ is the tree $\mathcal{T}$.

We extend our positive result to a class of graphs denoted as trees with *metric free edges* (Section 4.2): basically, the set of possible links to choose from is a tree, but a chosen link may induce other links from a graph which is *not* a tree; the induced links are specified in terms of distance between nodes in the given tree (see Section 4.2.1 for a formal definition). Our technical contribution here is a non-trivial algorithm extending the technique and some of the results for trees (Section 4.2.2).

We complement our negative results on geometric Euclidean instances for $\ell \geq 2$, by showing that the case $\ell = 1$ can be efficiently solved, even with the additional constraint of multicast trees of *bounded depth* (Section 4.3). We build a polynomial-time mechanism satisfying truthfulness, efficiency, NPT, VP, CS and CO, with the additional property of ensuring multicast trees of depth at most h, for any $1 \leq h \leq n-1$ given in input. This mechanism exploits an exact polynomial-time algorithm for building optimal broadcast trees (i.e., $D(T) = \mathcal{S}$) of bounded depth in [9].

In Section 5, we investigate mechanisms for general communication graphs which are obtained by fixing a "universal" multicast tree (i.e., a tree chosen independently from the agents reported valuation and which spans all nodes).

[4] In this case the set of destination nodes, also termed multicast group, is given in input.

We first observe that a shortest-path tree can be used as universal tree so to obtain a polynomial-time mechanism satisfying truthfulness, NPT, VP, CS, and $O(n)$-CO, for the case of any communication graph. In addition, this mechanism guarantees $|D(T^*)|$-efficiency, for all instances that admit an optimal solution T^* satisfying $|D(T^*)| \leq \varphi \frac{Worth(T^*,\mathbf{v})}{Cost(T^*)}$, for some constant $\varphi < 1$ (Section 5). Theorem 5 implies that this result cannot be obtained in general, unless $\mathsf{P} = \mathsf{NP}$ (although it might be possible to relax our assumption). The mechanism based on a universal shortest-path tree can be improved in the case of geometric Euclidean distances, where $O(1)$-CO can be also achieved whenever the mechanism services all users (see Theorem 33).

1.3 Further Related Work

Independently from our work, Biló *et al.* [4] gave an alternative proof of the existence of a truthful mechanism for the case in which the communication graph is a tree. Their proof exploits a result by Moulin and Shenker [22] and goes through a property of the cost function $Cost(\cdot)$ when restricted to trees. Their mechanism is the same as ours (i.e., the marginal-cost mechanism which belongs to the VCG family). Though this mechanism is well-defined, the authors of [4] do not provide any polynomial-time algorithm for computing the required solution (a multicast tree of maximizing the net worth) and the associated payments, nor a distributed implementation of such a mechanism. Biló *et al.* [4] also mention that this mechanism could be used to solve a more general case in which the communication graph is not a tree and observe that, in general, this approach does not guarantee any constant approximation. For the one-dimensional Euclidean case, they provide a polynomial-time truthful mechanism (the marginal-cost mechanism) which is efficient, and satisfies NPT, VP, CS and CO. This mechanism does not solve the case of bounded hops though. The work [4] also focuses on (approximate) budget-balance mechanisms: a mechanism is α-approximate BB if the total amount of money collected from the agents recovers the cost $Cost(A(\mathbf{b}))$ of the computed solution $A(\mathbf{b})$ and does not exceed $\alpha C^*(D(A(\mathbf{b})))$, where $C^*(D(A(\mathbf{b})))$ is the optimal cost of connecting s to $D(A(\mathbf{b}))$. In particular, for the ℓ-dimensional Euclidean case, the authors give a polynomial-time $2(3^\ell - 1)$-approximate BB group strategyproof mechanism satisfying NPT, VP and CS. Though their mechanism is not efficient, it guarantees $2(3^\ell - 1)$-CO. Moreover, the work [4] shows how to obtain a polynomial-time BB mechanism for the one-dimensional Euclidean case. Recently, the result in [4] for Euclidean instances has been improved in [27] so to obtain a polynomial-time group strategyproof $(3^\ell - 1)$-approximate BB mechanism (and thus $(3^\ell - 1)$-CO) satisfying NPT, VP and CS. The technique developed in the same work [27] also allows for a non-polynomial-time BB mechanism for the same problem. Finally, an experimental analysis of the mechanism based on universal multicast trees obtained with LASTs has been performed in [26].

2 Optimal Algorithms Yield Truthful Mechanisms

For the sake of completeness, we first recall the classical technique to obtain truthful mechanisms for utilitarian problems known as VCG mechanism [30,6,16]. We then show how to adapt this technique to our (non-utilitarian) problem.

2.1 Utilitarian Problems and Truthful VCG Mechanisms

Let us first consider a more general situation in which each agent i has a *private* piece of information v_i (sometimes called the type of agent i – see e.g. [23,29]). Each agent associates a monetary valuation to every feasible outcome. In particular, for every feasible solution X and for every i, agent i associates a monetary valuation equal to $Valuation_i(X, v_i)$ to solution X, where $Valuation_i(\cdot, \cdot)$ is an arbitrary function known to the mechanism. Notice that the mechanism cannot directly compute the value $Valuation_i(X, v_i)$ since v_i is known to agent i only. A mechanism asks to each agent i to report the value v_i and this agent can misreport this piece of information to any value b_i. Although the set of feasible solutions is known to the mechanism and independent of the v_is, these values are needed in order to output a feasible solution which maximizes a function $g(X, \mathbf{v})$, where $\mathbf{v} = (v_1, v_2, \ldots, v_n)$. In this section we consider so called *utilitarian* problems in which the goal is to maximize the function

$$g(X, \mathbf{v}) = \sum_{j=1}^{n} Valuation_j(X, v_j). \tag{4}$$

Intuitively speaking, the word "utilitarian" denotes those problems for which the goal is to maximize the sum of all agents' valuations (see e.g. [23,29]). Utilitarian problems are appealing since there is a general technique for constructing truthful mechanisms (mechanisms based on this technique are commonly known as VCG mechanisms [30,6,16]). We next describe this technique before adapting it to our non-utilitarian problem.

In the sequel we adopt the standard notation used in mechanism design literature: Given a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, let $\mathbf{x}_{-i}$ denote the vector $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ and $(y, \mathbf{x}_{-i}) := (x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$.

Let Alg denote an optimal algorithm for the utilitarian problem in question. That is, on input $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, the computed solution $Alg(\mathbf{v})$ maximizes, over all feasible solutions, the function in Equation 4. Notice that, since the set of feasible solutions is independent from $\mathbf{v}$, algorithm Alg returns always a feasible solution (i.e, even when its input is different from $\mathbf{v}$). Let $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ denote the vector of values reported by the agents. Then, VCG payments are defined as follows:

$$P^i_{VCG}(\mathbf{b}) := h_i(\mathbf{b}_{-i}) - \sum_{j=1, j\neq i}^{n} Valuation_j(X^*, b_j), \tag{5}$$

where $X^* := Alg(\mathbf{b})$ and $h_i(\cdot)$ is any function independent of b_i. Observe that, by the hypothesis on algorithm Alg, solution X^* maximizes the function $g(X, \mathbf{b}) = \sum_{j=1}^{n} Valuation_j(X, b_j)$.

The combination of such an optimal algorithm Alg with the VCG payments above yields the so called VCG mechanism $M = (Alg, P_{VCG})$. The following is a classical result in mechanism design (for completeness, we also provide its proof):

Theorem 1 *[16] If Alg is an optimal algorithm for a utilitarian problem* Π*, then the mechanism* $M = (Alg, P_{VCG})$ *is truthful for* Π*.*

PROOF. Observe that, on input a vector $\mathbf{b} = (b_1, b_2, \ldots, b_n)$ of values reported by the agents, the mechanism computes a solution $X^* = Alg(\mathbf{b})$ and charges each agent i an amount equal to $P^i_{VCG}(\mathbf{b})$.[5] Since $Valuation_i(X^*, v_i)$ is the monetary valuation that agent i associates to solution X^*, the corresponding utility is equal to

$$u_i(b_i, \mathbf{b}_{-i}|v_i) := Valuation_i(X^*, v_i) - P^i_{VCG}(\mathbf{b}),$$

From Equations 4 and 5 we obtain

$$\begin{aligned} u_i(b_i, \mathbf{b}_{-i}|v_i) &= Valuation_i(X^*, v_i) - h_i(\mathbf{b}_{-i}) \\ &\quad + g(X^*, \mathbf{b}) - Valuation_i(X^*, b_i) \\ &= -h_i(\mathbf{b}_{-i}) + g(X^*, (v_i, \mathbf{b}_{-i})) \\ &= -h_i(\mathbf{b}_{-i}) + g(Alg(b_i, \mathbf{b}_{-i}), (v_i, \mathbf{b}_{-i})). \end{aligned}$$

[5] As in our problem, we define the payments as the amount of "money" that an agent must pay to the mechanism. Clearly, this is without loss of generality since negative values of $P^i_{VCG}(\cdot)$ can model the case in which agent i receives money from the mechanism.

Since Alg is optimal, we have that

$$g(Alg(v_i, \mathbf{b}_{-i}), (v_i, \mathbf{b}_{-i})) \geq g(X', (v_i, \mathbf{b}_{-i})),$$

for every other solution X'. In particular, we have

$$g(Alg((v_i, \mathbf{b}_{-i}), (v_i, \mathbf{b}_{-i})) \geq g(Alg(b_i, \mathbf{b}_{-i}), (v_i, \mathbf{b}_{-i})),$$

thus implying $u_i(v_i, \mathbf{b}_{-i}|v_i) \geq u_i(b_i, \mathbf{b}_{-i}|v_i)$. This completes the proof. □

2.2 Reducing the Problem to a Utilitarian One

In our problem the valuation of a solution T of agent i is equal to

$$Valuation_i(T, v_i) := \begin{cases} v_i \text{ if } i \in D(T), \\ 0 \text{ otherwise.} \end{cases}$$

However, the optimization function $NetWorth(\cdot)$ does not satisfy the definition of utilitarian problems, since

$$NetWorth(T, \mathbf{v}) \neq \sum_{i=1}^{n} Valuation_i(T, v_i) = \sum_{i \in D(T)} v_i = Worth(T, \mathbf{v}).$$

Nevertheless, we next show that the VCG technique can be adapted to our problem. The main idea is to initially charge each node by the cost of its ingoing edge in the tree (computed as in the wireless network case) so to "reduce" our problem to a utilitarian one. Towards this end, we first define the following indicator function which tells us whether an edge (i, k) is counted in the function $Cost(\cdot)$ of a solution T:

Definition 2 *For any tree $T \subseteq \mathcal{E}$ and for any node i, let $e(T, i)$ be the edge connecting i with its parent in T. Also let*

$$FreeEdges(i, j) := \{(i, k) \in \mathcal{E} | w(i, k) \leq w(i, j) \wedge k < j\}.$$

$$Charge(T, i) := \begin{cases} 0 & \textit{if } e(T, i) \in FreeEdges(l, m) \textit{ with } (l, m) \in T \\ w(e(T, i)) & \textit{otherwise.} \end{cases}$$

Notice that, when the solution contains more than one outgoing edge of maximal length, then we consider the cost of the one connecting to the node j of higher index.

Every rooted tree T contains one ingoing edge $e(T,i)$ for each non-root node $i \in D(T)$. Thus from Definition 2 and Equation 1 we have that

$$\sum_{i \in D(T)} Charge(T,i) = \sum_{i \in \mathcal{S}} \max_{j:(i,j) \in T} w(i,j) = Cost(T). \tag{6}$$

Theorem 3 *Let A be a (polynomial-time) exact algorithm for maximizing the $NetWorth(\cdot)$ function. Then the cost sharing problem on wireless networks admits a (polynomial-time) mechanism $M = (A, P)$ satisfying truthfulness, efficiency, NPT, VP, CS and CO.*

PROOF. Truthfulness. Let $T \subseteq \mathcal{E}$ be any subtree of $\mathcal{G}$ rooted at s, and let $Serviced_i(T) = 1$ if $i \in D(T)$, and 0 otherwise. Let us first consider the problem in which each agent i has a valuation equal to

$$Valuation'_i(T, v_i) := Serviced_i(T)(v_i - Charge(T,i)). \tag{7}$$

Intuitively, we start by charging to each agent receiving the transmission the cost of its ingoing edge, if that edge is not "for free" in T. We next show that, when considering functions $Valuation'_i(\cdot,\cdot)$, maximizing $NetWorth(\cdot)$ is a utilitarian problem (see Equation 4). That is, for any solution $T \subseteq \mathcal{E}$,

$$NetWorth(T, \mathbf{v}) = \sum_{i=1}^{n} Valuation'_i(T, v_i), \tag{8}$$

where $NetWorth(\cdot)$ is defined in Equations 2 and 3.

Since $NetWorth(T,\mathbf{v}) = Worth(T,\mathbf{v}) - Cost(T)$, Equations 7 and 6 imply

$$NetWorth(T,\mathbf{v}) = \sum_{i \in D(T)} v_i - \sum_{i \in D(T)} Charge(T,i) \tag{9}$$

$$= \sum_{i \in D(T)} Valuation'_i(T, v_i) = \sum_{i=1}^{n} Valuation'_i(T, v_i). \tag{10}$$

We have thus proven Equation 8. Notice that, any optimal algorithm A for $NetWorth(\cdot)$ returns a directed subtree of $\mathcal{G}$ rooted at s. Moreover, on input the true valuations $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, the solution $A(\mathbf{v})$ maximizes the function in Equation 8, which satisfies the definition of utilitarian problem (see Equation 4) with respect to the valuation functions $Valuation'_i(\cdot,\cdot)$. We can thus apply the VCG paradigm (see Theorem 1) and define payments $P^i_{VCG}(\cdot)$ such that $M = (A, P_{VCG})$ is truthful for agents whose valuation functions are $Valuation'_i(\cdot,\cdot)$. That is, for every i, for every b_i, and for every $\mathbf{b}_{-i}$, the following inequality holds:

$$Valuation'_i(A(v_i, \mathbf{b}_{-i}), v_i) - P^i_{VCG}(v_i, \mathbf{b}_{-i}) \geq \\ Valuation'_i(A(b_i, \mathbf{b}_{-i}), v_i) - P^i_{VCG}(b_i, \mathbf{b}_{-i}) \quad (11)$$

where $P^i_{VCG}(\cdot)$ is defined as in Equation 5.

We next modify the payments so that the resulting mechanism is truthful for the original valuation functions $Valuation_i(T, v_i) = Serviced_i(T) \cdot v_i$. For any agent i, let us consider the following payment function:

$$P^i(\mathbf{b}) := P^i_{VCG}(\mathbf{b}) + Serviced_i(A(\mathbf{b})) \cdot Charge(A(\mathbf{b}), i), \quad (12)$$

where $P^i_{VCG}(\cdot)$ is defined as above. Let us observe that

$$\begin{aligned} u_i(b_i, \mathbf{b}_{-i}|v_i) := & Valuation_i(A(b_i, \mathbf{b}_{-i}), v_i) - P^i(b_i, \mathbf{b}_{-i}) \\ = & Serviced_i(A(b_i, \mathbf{b}_{-i})) \cdot v_i - P^i(b_i, \mathbf{b}_{-i}) \\ = & Serviced_i(A(b_i, \mathbf{b}_{-i})) \cdot v_i - P^i_{VCG}(b_i, \mathbf{b}_{-i}) \\ - & Serviced_i(A(b_i, \mathbf{b}_{-i})) \cdot Charge(A(\mathbf{b}), i) \\ = & Serviced_i(A(b_i, \mathbf{b}_{-i})) \cdot (v_i - Charge(A(\mathbf{b}), i)) \\ - & P^i_{VCG}(b_i, \mathbf{b}_{-i}) \\ = & Valuation'_i(A(b_i, \mathbf{b}_{-i}), v_i) - P^i_{VCG}(b_i, \mathbf{b}_{-i}). \end{aligned}$$

From the above identities and from the inequality in Equation 11 we obtain

$$\begin{aligned} u_i(v_i, \mathbf{b}_{-i}|v_i) = & Valuation'_i(A(v_i, \mathbf{b}_{-i}), v_i) - P^i_{VCG}(v_i, \mathbf{b}_{-i}) \\ \geq & Valuation'_i(A(b_i, \mathbf{b}_{-i}), v_i) - P^i_{VCG}(b_i, \mathbf{b}_{-i}) = u_i(b_i, \mathbf{b}_{-i}|v_i). \end{aligned}$$

This proves the truthfulness.

In order to guarantee NPT and VP, we have to define a suitable function $h_i(\mathbf{b}_{-i})$, where $h_i(\mathbf{b}_{-i})$ is the function used in the definition of $P^i_{VCG}(\cdot)$ in Equation 5. Towards this end, we let

$$(0, \mathbf{b}_{-i}) := (b_1, b_2, \ldots, b_{i-1}, 0, b_{i+1}, \ldots, b_n).$$

For any $E \subseteq \mathcal{E}$, let us consider

$$NetWorth(E, (0, \mathbf{b}_{-i})) = \left(\sum_{j \neq i, j \in D(E)} b_j \right) - Cost(E),$$

that is, the net worth of E computed with respect to $(0, \mathbf{b}_{-i})$. We define

$$h_i(\mathbf{b}_{-i}) := NetWorth(A(0, \mathbf{b}_{-i}), (0, \mathbf{b}_{-i})).$$

NPT. Let us observe that

$$\begin{aligned}\sum_{j\neq i,j=1}^{n} Valuation'_j(A(\mathbf{b}),b_j) &= \sum_{j\neq i,j\in D(A(\mathbf{b}))} b_j - Cost(A(\mathbf{b}))\\ &+ Serviced_i(A(\mathbf{b}))\cdot Charge(A(\mathbf{b}),i)\\ &= NetWorth(A(\mathbf{b}),(0,\mathbf{b}_{-i}))\\ &+ Serviced_i(A(\mathbf{b}))\cdot Charge(A(\mathbf{b}),i).\end{aligned}$$

The definition of $h_i(\mathbf{b}_{-i})$, Equation 5 and Equation 12 imply the following chain of equalities:

$$\begin{aligned}P^i(b_i,\mathbf{b}_{-i}) &= P^i_{VCG}(\mathbf{b}) + Serviced_i(A(\mathbf{b}))\cdot Charge(A(\mathbf{b}),i)\\ &= h_i(\mathbf{b}_{-i}) - \sum_{j\neq i,j=1}^{n} Valuation'_j(A(\mathbf{b}),b_j)\\ &+ Serviced_i(A(\mathbf{b}))\cdot Charge(A(\mathbf{b}),i)\\ &= NetWorth(A(0,\mathbf{b}_{-i}),(0,\mathbf{b}_{-i})) - NetWorth(A(\mathbf{b}),(0,\mathbf{b}_{-i})).\end{aligned} \tag{13}$$

Since A is an exact algorithm, $T = A(0,\mathbf{b}_{-i})$ is an optimal solution for the instance modified by letting the user valuations be equal to $(0,\mathbf{b}_{-i})$. Thus, $NetWorth(T,(0,\mathbf{b}_{-i})) \geq NetWorth(E,(0,\mathbf{b}_{-i}))$, for every $E \subseteq \mathcal{E}$. In particular, for $E = A(\mathbf{b})$, Equation 13 implies that $P^i(b_i,\mathbf{b}_{-i}) \geq 0$.

VP. By contradiction, let us assume that $P^i(\mathbf{b}) > b_i$, for some i and for some $\mathbf{b} = (b_i,\mathbf{b}_{-i})$. By the identities in Equation 13 and since $Serviced_i(\cdot)$ is either 0 or 1, we have

$$\begin{aligned}P^i(b_i,\mathbf{b}_{-i}) &= NetWorth(A(0,\mathbf{b}_{-i}),(0,\mathbf{b}_{-i})) - NetWorth(A(\mathbf{b}),(0,\mathbf{b}_{-i}))\\ &> b_i \geq b_i\cdot Serviced_i(A(\mathbf{b})).\end{aligned}$$

From this inequality, from the fact that $Serviced_i(A(\mathbf{b})) = 1$ only if $i \in D(A(\mathbf{b}))$, and from the definition of $NetWorth(\cdot)$, we obtain

$$\begin{aligned}NetWorth(A(0,\mathbf{b}_{-i}),(0,\mathbf{b}_{-i})) &> b_i\cdot Serviced_i(A(\mathbf{b}))\\ &+ NetWorth(A(\mathbf{b}),(0,\mathbf{b}_{-i}))\\ &= b_i\cdot Serviced_i(A(\mathbf{b}))\\ &+ \left(\sum_{j\neq i,j\in D(A(\mathbf{b}))} b_j\right) - Cost(A(\mathbf{b}))\\ &= \left(\sum_{j\in D(A(\mathbf{b}))} b_j\right) - Cost(A(\mathbf{b}))\end{aligned}$$

$$= NetWorth(A(\mathbf{b}), \mathbf{b}). \tag{14}$$

From the definition of $NetWorth(\cdot)$ we also obtain

$$\begin{aligned} NetWorth(A(0, \mathbf{b}_{-i}), \mathbf{b}) &= \left(\sum_{j \in D(A(0,\mathbf{b}_{-i}))} b_j \right) - Cost(A(0, \mathbf{b}_{-i})) \\ &\geq \left(\sum_{j \neq i, j \in D(A(0,\mathbf{b}_{-i}))} b_j \right) - Cost(A(0, \mathbf{b}_{-i})) \\ &= NetWorth(A(0, \mathbf{b}_{-i}), (0, \mathbf{b}_{-i})). \end{aligned} \tag{15}$$

By combining (14) with (15) we have

$$NetWorth(A(0, \mathbf{b}_{-i}), \mathbf{b}) > NetWorth(A(\mathbf{b}), \mathbf{b}),$$

thus implying that solution $A(\mathbf{b})$ in not optimal for $\mathbf{b}$. This contradicts the fact that A is an exact algorithm. Hence, it must be the case $P^i(\mathbf{b}) \leq b_i$ for all i and all $\mathbf{b} = (b_i, \mathbf{b}_{-i})$.

Efficiency, CO, CS. The efficiency follows from the fact that A is an exact algorithm for maximizing the $NetWorth(\cdot)$ function. Since Efficiency implies CO (see discussion at page 8) the latter condition is also satisfied. Moreover, for a sufficiently large b_i, since A is an exact algorithm, it must eventually send the transmission to the corresponding user. Thus the CS property holds. □

We will sometimes restrict the output of the algorithm to fixed subset of feasible solutions. These solutions are either given by some problem variant we consider (e.g., trees with a bounded height – see Section 4.3), or they are introduced to "approximate" the net worth in polynomial time when the problem is NP-hard (see Sections 3 and 5). It turns out that the proof of the above theorem extends to such cases:

Remark 4 *[24] The mechanism of Theorem 3 satisfies truthfulness, NPT, VP, and CS also for algorithms that optimize the net worth function only with respect to a fixed family $\mathcal{F}$ of subsets of edges such that, for every i, the family $\mathcal{F}$ contains a subset of edges for which i is serviced. The mechanism satisfies efficiency and the CO with respect to the set of solutions in $\mathcal{F}$ (i.e., net worth and cost are optimized over all solutions in $\mathcal{F}$). When $\mathcal{F}$ is given by the problem, we simply say that the mechanism satisfies the efficiency and the CO conditions.*

3 Hardness Results

In this section we show that, unless $\mathsf{P} = \mathsf{NP}$, it is impossible to optimize the net worth on general communication graphs. In particular, we show that the problem is even hard to approximate within any "reasonable" factor and that it remains NP-hard for geometric Euclidean instances. Our negative results hold even when considering polynomial-time algorithms that (somehow) are provided with the true agents' valuation. So, our hardness results apply to polynomial-time mechanisms as well.

Theorem 5 *For any $r > 0$, no polynomial-time r-approximation algorithm (mechanism) exists, unless $\mathsf{P} = \mathsf{NP}$. This also holds for three-layer graphs in which layer 1 contains s only.*

PROOF. Our proof is a simple modification of the proof in [13]. We reduce 3-SAT to our problem as follows. Let $f = (x_1, \ldots, x_n; C_1, \ldots, C_m)$ be a SAT instance. W.l.o.g., let us assume that f contains all clauses $(x_i \vee \overline{x}_i)$, $i = 1, \ldots, n$. Consider a three-level graph $\mathcal{G}_f$ defined as follows:

- the first level contains only the source s, the second level contains all $2n$ literals and the third level all m clauses;
- s is connected to every node in the second level and the weight of each of these edges is $m \cdot K - n - 1$, where $K \gg m$;
- every literal node l_i in the second level is connected to every clause node corresponding to those clauses containing l_i; the weight of every edge between the second and the third level is 1;
- s has valuation ϵ, every clause node has valuation K, and literal nodes have all valuation 0.

We first observe that $\mathcal{G}_f$ has a trivial solution of $NetWorth$ equal to ϵ: take s only.

In order to obtain a better $NetWorth$ we must include at least one of the edges from s to the second level. This will include all such edges since they all have the same weight. This yield a $Cost$ of $m \cdot K - n - 1$, which forces us to reach all nodes of level three (otherwise $NetWorth \leq \epsilon + (m-1) \cdot K - m \cdot K + n + 1 < \epsilon$). Since f contains all clauses $(x_i \vee \overline{x}_i)$, between the second and the third level, we must pick all edges of x_i or all edges of $\overline{x}_i$. Thus, we must pay a $Cost$ of at least n. If this cost is at least $n + 1$, then the resulting net worth is at most $\epsilon + m \cdot K - m \cdot K + n + 1 - n - 1 = \epsilon$. So, solutions of $NetWorth$ better than ϵ correspond to satisfiable formulas f; in this case $NetWorth^*(\mathcal{G}_f) = \epsilon + m \cdot K - n - m \cdot K + n + 1 = 1 + \epsilon$, where $NetWorth^*(\mathcal{G}_f)$ denotes the optimum for the instance $\mathcal{G}_f$. If f is unsatisfiable, then $NetWorth^*(\mathcal{G}_f) = \epsilon$.

If f is satisfiable, then any r-approximate algorithm for our problem returns a solution of net worth N, with N satisfying $N \geq NetWorth^*(\mathcal{G}_f)/r = (1+\epsilon)/r$. By choosing, in the above reduction, ϵ sufficiently small we can guarantee that $\frac{1+\epsilon}{\epsilon} > r$, thus implying $(1+\epsilon)/r > \epsilon$. So, our r-approximate algorithm would be able to distinguish whether the optimum is ϵ (i.e., f unsatisfiable) or greater (i.e., f satisfiable). This implies that approximating the optimal net worth within any factor $r > 0$ is NP-hard. Thus the theorem follows. □

We next consider the restriction in which stations are located on the ℓ-dimensional Euclidean space and each station has a maximum transmission range which suffices to directly transmit to any other station. In this case, the communication graph is a *complete graph* with $w(i,j) := d(i,j)^\alpha$, where $\alpha \geq 1$ is a fixed constant and $d : \mathbb{R}^\ell \to \mathbb{R}^+$ is the Euclidean distance. The problem of maximizing the net worth is NP-hard even when stations are located on the plane:

Theorem 6 *The problem of maximizing $NetWorth(\cdot)$ is NP-hard, even when restricted to geometric wireless networks, for any $\ell \geq 2$ and any $\alpha > 1$.*

PROOF. We reduce from the Minimum Energy Consumption Broadcast Subgraph (MECBS) problem: the input of the problem is the same as in our problem, and the goal is to construct a tree T such that $D(T) = \mathcal{S}$ and $Cost(T)$ is minimized. Notice that the v_is play no role in this problem. In [7] it has been proved that the problem remains NP-hard even for $\ell = 2$ and for any $\alpha > 1$. The reduction works as follows. Given an instance I_{MECBS} of MECBS, we consider the instance I_{NW} of the cost sharing problem having the same set of stations and every v_i equal to a sufficiently large L. So, any optimal solution T^* for I_{NW} must guarantee $D(T^*) = \mathcal{S}$. Because of this, the optimum is reached when the cost is minimized, that is, when a minimum cost solution for I_{MECBS} is computed. This completes the proof. □

4 Special Communication Graphs

In this section, we will focus on restrictions of the problem which admits *exact polynomial-time* algorithms, and thus, truthful exact polynomial-time mechanisms. In particular, we consider several families of communication graphs for which the problem of computing a tree with optimal net worth becomes tractable.

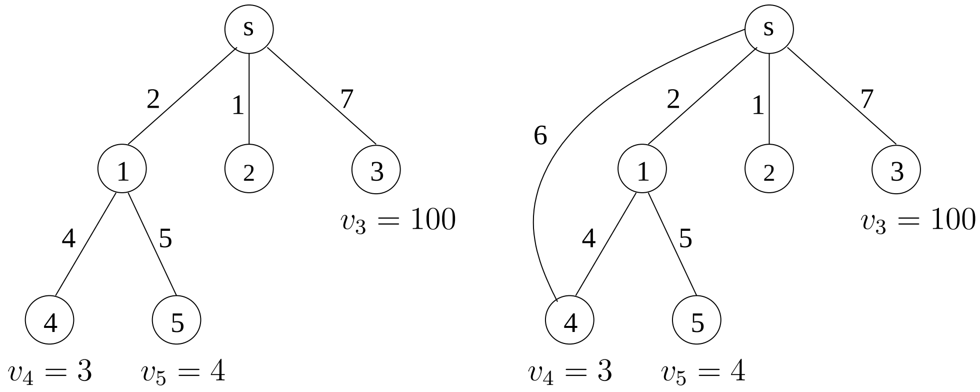

Fig. 3. An example explaining why the recursion is possible for trees and why in general graph instances it is not. On the left side we have a tree in which the optimal solution involves the optimum of the subtree rooted at node 1 (the optimum is obtained by transmitting to all nodes and this is also optimal when restricting to the subtree rooted at 1). Notice that the only way to reach the children of node 1 is by transmitting to this node. On the right side an instance obtained by adding an edge $(s, 4)$ with $w(s, 4) = 6$. The optimal solution of the resulting graph does not use any edge in the subtree rooted in 1 since node 4 can be reached using edge $(s, 4)$ which is for free. Indeed, using edge $(1, 5)$ would cost 5 and increase the worth only by $v_5 = 4$.

4.1 Trees

We proceed similarly to [13] and assume that the communication graph is a directed tree $\mathcal{T}$. This has interesting consequences on the set of "free edges": a solution containing an edge (i, j) also contains all edges (i, k) such that $w(i, k) \leq w(i, j)$. More importantly, since $\mathcal{T}$ is a tree, this is the only case in which an edge (i, k) is "for free".

Definition 7 *For every $i \in \mathcal{T}$, let c_i be the weight of the edge connecting i to its parent in $\mathcal{T}$, i.e., $c_i = w(e(\mathcal{T}, i))$ and $c_s = 0$. We denote by $Children(i)$ the set of nodes children of i in $\mathcal{T}$. $\mathcal{T}_i$ denotes the subtree rooted at i. Finally, $NetWorth^*(i, \mathbf{v})$ denotes the* optimal net worth of *$\mathcal{T}_i$, that is, the optimum for the instance in which i is the source node, the tree is $\mathcal{T}_i$, and $\mathbf{v}$ is the vector of users valuations.*

In the remaining of this section, we fix an arbitrary vector $\mathbf{v} = (v_1, \ldots, v_n)$ of agents valuations and show how to compute the optimum w.r.t. this vector. For the sake of readability, we omit '$\mathbf{v}$' in the definitions and notation used throughout this section.

Intuitively speaking, once choosing an edge (i, j), the problem breaks down into subproblems in which:

- every child k of i, such that $w(i, k) \leq w(i, j)$, becomes a new root;

- every node in $\mathcal{T}_k$ can be reached only via node k.

This gives us the possibility of using a recursive algorithm. On the contrary, if we consider general graph instances, then a simple argument shows that this approach does not lead to optimal solutions (see the examples in Figure 3).

The next lemma provides the basic recursion rule to be used in our algorithm.

Lemma 8 *For every node i, it holds that*

$$NetWorth^*(i) = v_i + \max \begin{cases} 0 \\ \max_{j \in Children(i)} \{PayEdge(i,j)\} \end{cases} \tag{16}$$

where

$$PayEdge(i,j) := -c_j + \sum_{\substack{k \in Children(i), \\ c_k \leq c_j}} NetWorth^*(k).$$

PROOF. The proof is by induction on the hight h of $\mathcal{T}_i$. For $h = 1$, since i is a leaf node, then $NetWorth^*(i) = v_i$. Let us now assume that the lemma holds for any $h' \leq h-1$, and let us prove it for h. Let T_i^* denote an optimal solution for $\mathcal{T}_i$. Let us first observe that, if $NetWorth^*(i) = NetWorth(T_i^*) > v_i$, then T_i^* must contain at least one outgoing edge from node i. Let (i,j) be the longest such edge in T_i^*. For any node $k \in \mathcal{T}_i$, let $T_{i,k}^*$ denote the subtree of T_i^* rooted at k. Since $(i,j) \in T_i^*$, then every edge (i,k), with $w(i,k) \leq w(i,j)$, is for free. Hence, $k \in D(T_i^*)$, for all $k \in Children(i)$ such that $c_k \leq c_j$. Therefore,

$$Cost(T_i^*) = c_j + \sum_{\substack{k \in Children(i), \\ c_k \leq c_j}} Cost(T_{i,k}^*), \tag{17}$$

$$Worth(T_i^*) = v_i + \sum_{\substack{k \in Children(i), \\ c_k \leq c_j}} Worth(T_{i,k}^*). \tag{18}$$

Let us now suppose, by contradiction, that there exists a $l \in Children(i)$, with $c_l \leq c_j$ and $NetWorth(T_{i,l}^*) < NetWorth^*(l)$. Let T_l' denote the subtree of $\mathcal{T}_l$ yielding an optimal net worth w.r.t. $\mathcal{T}_l$, that is, $NetWorth(T_l') = NetWorth^*(l)$. Let T_i' be the solution obtained by replacing, in T_i^*, $T_{i,l}^*$ with T_l'. Since T_i' still contains all edges (i,k), with $w(i,k) \leq w(i,j)$, we have that

$$Worth(T_i') \geq Worth(T_l') + v_i + \sum_{\substack{k \in Children(i), \\ c_k \leq c_j, k \neq l}} Worth(T_{i,k}^*). \tag{19}$$

```
Algorithm Wireless_Trees at node i
 (1) After receiving a message μ^j from each child j ∈ Children(i)
     do
     (a) Add(j) := −c_j + Σ_{k∈Children(i),c_k≤c_j} μ^k;
     (b) T(i) := ∅;
     (c) if max_{j∈Children(i)} Add(j) < 0 then μ^i := v_i
     (d) else do
           (i) Add := max_{j∈Children(i)} Add(j);
          (ii) μ^i := v_i + Add;
         (iii) j* := max{j ∈ Children(i)|Add(j) = Add};
          (iv) T(i) := {(i,j)|w(i,j) ≤ w(i,j*)};
 (2) send μ^i to parent p(i);
```

Fig. 4. The distributed algorithm for trees computing an optimal solution in bottom-up fashion.

Clearly, $Cost(T_i') = Cost(T_i^*) - Cost(T_{i,l}^*) + Cost(T_l')$. This, combined with Equation 19, yields

$$NetWorth(T_i') \geq NetWorth(T_i^*) - NetWorth(T_{i,l}^*) + NetWorth(T_l').$$

From the hypothesis $NetWorth(T_{i,l}^*) < NetWorth^*(l) = NetWorth(T_l')$, we obtain $NetWorth(T_i') > NetWorth(T_i^*)$, thus contradicting the optimality of T_i^*. So, for every $k \in Children(i)$ with $c_k \leq c_j$, it must hold $NetWorth(T_{i,k}^*) = NetWorth^*(k)$. From Equations 17-18 we obtain

$$\begin{aligned} NetWorth^*(i) = NetWorth(T_i^*) &= v_i - c_j + \sum_{\substack{k\in Children(i),\\ c_k\leq c_j}} NetWorth(T_{i,k}^*) \\ &= v_i - c_j + \sum_{\substack{k\in Children(i),\\ c_k\leq c_j}} NetWorth^*(k). \end{aligned} \tag{20}$$

The optimality of T_i^* implies that, if $NetWorth^*(i) > v_i$, then j must be the node in $Children(i)$ maximizing the right quantity in Equation 20. The lemma thus follows from the fact that $NetWorth^*(i) \geq v_i$: taking no edges in $\mathcal{T}_i$ yield a net worth equal to v_i. This completes the proof. □

Theorem 9 *For any communication graph $\mathcal{T}$ which is a tree, algorithm* Wireless_Trees *in Fig. 4 computes the optimal net worth in $O(n)$ time, using $O(n)$ total messages and sending one message per link.*

PROOF. We first observe that every node $i \neq s$ sends exactly one message to its parent. This implies the message complexity. As for the running time, we

Algorithm Wireless_Trees_Pay at node $i = p(j)$

(1) After receiving the message $(NetWorth^*, \lambda^i)$ from parent $p(i)$, for each child $j \in \{j|\ (i,j) \in T(i)\}$ do
/* compute the payments for all $j \in Children(i)$ that receive the transmission */
  (a) $\lambda^{j=0} := NetWorth^* - v_j$;   /* compute $NetWorth(A(\mathbf{v}), (0, \mathbf{v}_{-j}))$ */
  (b) $x := \max_{k \in Children(i), k \neq j} \{-c_k + \sum_{l \in Children(i), c_l \leq c_k} \mu^l\}$;
  (c) $\lambda^{-j} := NetWorth^* - \mu^i + v_i + \max\{0, x\}$;   /* compute $NetWorth(A(0, \mathbf{v}_{-j}), (0, \mathbf{v}_{-j}))$ */
  (d) $\lambda^j := \lambda^{-j} - \lambda^{j=0}$;
  (e) send $(NetWorth^*, \lambda^j)$ to child j;

Fig. 5. The distributed algorithm for computing the payments in top-down fashion. The code refers to a non-source node; the computation is initialized by s which, at the end of the bottom-up phase of Algorithm Wireless_Trees, has computed the value $NetWorth^* = \mu^s$ and it executes the instructions 1a-1e for every $j \in Children(s)$ that receive the transmission.

observe that every node requires a time linear in the number of its children. So, the overall running time is linear in n, since Wireless_Trees proceeds in bottom-up fashion. Finally, let T be the solution obtained by considering the component of $\cup_{i=1}^n T(i)$ connected to s (this can be easily obtained in top-down fashion), where $T(i)$ is defined in Wireless_Trees (in Step 1b and modified in Step 1d.iv in case i has at least one child). The optimality of T can be proved by induction on the hight of i. In particular, we show that, for every node i, the net worth of T w.r.t. $\mathcal{T}_i$ is equal to μ^i and that $\mu^i = NetWorth^*(i)$, where $NetWorth^*(i)$ is defined as in Definition 7. The base step (i.e., when i is a leaf) is trivial. Let us assume that, for every child j of i, $\mu^j = NetWorth^*(j)$. The case $\mu^i = v_i$ is trivial since $T(i) = \emptyset$. As for the other case, let j^* be defined as in Wireless_Trees. Then, by inductive hypothesis and by the definition of $T(i)$, the net worth of T w.r.t. $\mathcal{T}_i$ is equal to

$$
\begin{aligned}
v_i - c_j + \sum_{\substack{k \in Children(i), \\ c_k \leq c_{j^*}}} \mu^k &= v_i - c_j + \sum_{\substack{k \in Children(i), \\ c_k \leq c_{j^*}}} NetWorth^*(k) \\
&= v_i + Add(j^*) = \mu^i.
\end{aligned}
$$

From the definition of j^*, from the "if then else" instructions, and by Lemma 8, we obtain that the above quantity coincides with $NetWorth^*(i)$. This completes the proof. □

In Figure 5 we show a distributed top-down algorithm for computing the payments $P^j(\cdot)$ of Theorem 3. Using the algorithm in Figure 5, from Theorem 9

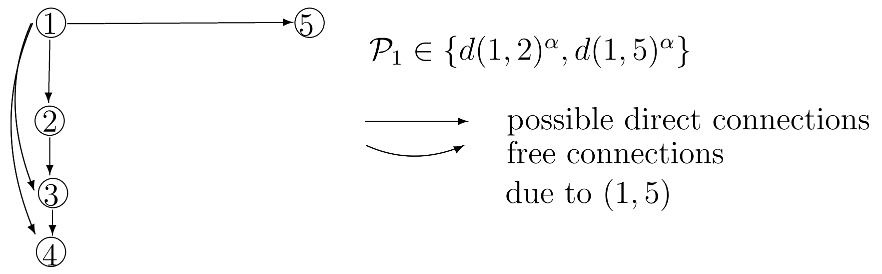

Fig. 6. If the power that a station can transmit with is predetermined so to form a tree, the subgraph containing the connections obtained for free may not be a tree.

and from Theorem 3 we obtain the following:

Corollary 10 *If the communication graph is a tree, then the cost sharing problem on wireless networks admits a distributed $O(n)$-time truthful mechanism $M = (A, P)$ satisfying efficiency, NPT, VP, CS and CO. Moreover, M requires a total of $O(n)$ messages and 2 messages per link.*

4.2 Trees with Metric Free Edges

Let us consider the geometric instance in Figure 6 with the following additional constraint. Station 1 can choose only between two powers: either to reach 2 or 5; all other stations have maximum power only sufficient to reach the closest one. Thus, the set of possible edges that we can use to build a solution is a tree (the set of all straight-line edges in Figure 6). However, if station 1 directly transmits to station 5, then it also reaches all other stations. So, when adding the corresponding free edges, the resulting graph is no longer a tree.

A simple argument shows that our algorithm for trees does not always compute an optimal solution for instances like the one in Figure 6. However, in the sequel we will define a class of instances, termed *trees with metric free edges*, for which an exact polynomial-time algorithm exists. This yields a truthful polynomial-time mechanism satisfying also NPT, VP, CO and CS.

4.2.1 Defining Trees with Metric Free Edges

We consider the case in which the set of edges of the communication graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ is partitioned into two sets $\mathcal{T} \cup FreeEdges^*$. Similarly to the case considered in Section 4.1, $\mathcal{T}$ induces a directed spanning tree of $\mathcal{G}$ rooted at s, i.e., $D(\mathcal{T}) = \mathcal{S}$. The tree is directed downward and, for every node i, we must select a set of outgoing edges in the set $\mathcal{T}$. However, each edge (i, j) in $\mathcal{T}$ induces an additional set of "free edges" $FreeEdges^*(i, j)$, defined below. Adding all such connections to the solution does not increase its cost. These connections are specified as follows:

Definition 11 *For every node i, we let $\mathcal{T}_i$ denote the subtree rooted at i and $\mathcal{S}_i$ denote its subset of nodes. Let $w^*(i,j)$ denote the weight of the path in $\mathcal{T}$ connecting the node i to one of its descendants $j \in \mathcal{S}_i$. Also, for any $(i,j) \in \mathcal{T}$, let*

$$FreeEdges^*(i,j) := \{(i,k) \notin \mathcal{T} \mid \; k \in \mathcal{S}_i, \;\; w^*(i,k) \leq w(i,j)\}.$$

Moreover, for every $T \subseteq \mathcal{T}$, let $FreeEdges^*(T) := \bigcup_{(i,j)\in T} FreeEdges^*(i,j)$. An instance of trees with metric free edges *is a tree $\mathcal{T}$ and its set of feasible solutions consists of*

$$Metric(\mathcal{T}) := \{T \cup F \mid \; T \subseteq \mathcal{T}, F \subseteq FreeEdges^*(T)\}.$$

By referring to the example in Figure 6, we have

$$\mathcal{T} = \{(1,2),(2,3),(3,4),(1,5)\}$$

and $FreeEdges^*(1,5) = \{(1,3),(1,4)\} = FreeEdges^*(\mathcal{T})$. Hence, Definition 11 captures the example in Figure 6 when $\alpha = 1$. (When $\alpha = 1$ the problem is only known to be in P when the source station has enough power to reach directly all other stations.)

Since we do not allow for all possible subset of edges but only those in $Metric(\mathcal{T})$, we adapt the definitions of efficiency and CO in the natural way. That is, by considering only these feasible subsets of edges and thus by replacing '$E \subseteq \mathcal{E}$' with '$E \in Metric(\mathcal{T})$'.

Notice that, for every feasible $E \subseteq Metric(\mathcal{T})$, no edge in $FreeEdges^*(\mathcal{T})$ can appear as the longest outgoing edge of a node. Thus, we have

$$Cost(E) = Cost(E \cap \mathcal{T}).$$

For trees with metric free edges, we can always restrict to solutions that build a subtree T of $\mathcal{T}$ and then add all possible edges in $FreeEdges^*(T)$:

Remark 12 *For every instance $\mathcal{T}$ of trees with metric free edges and for every feasible solution $S \in Metric(\mathcal{T})$ it is possible to construct, in polynomial time, another feasible solution $E \in Metric(\mathcal{T})$ such that $E = T \cup F$, T is a subtree of $\mathcal{T}$, $F = FreeEdges^*(T)$, and $NetWorth(E, \mathbf{v}) \geq NetWorth(S, \mathbf{v})$.*

4.2.2 The Algorithm

We next provide a recursive way of optimizing the function $NetWorth(\cdot)$ in the case of trees with metric free edges. In the remaining of this section, we fix an arbitrary vector $\mathbf{v} = (v_1, \ldots, v_n)$ of agents valuations and show how to compute the optimum w.r.t. this vector. For the sake of readability, we omit '$\mathbf{v}$' in the definitions and notation used throughout this section.

We will restrict to feasible solutions $E \in Metric(\mathcal{T})$ as in Remark 12. Essential in characterizing and computing such an optimal $E \subseteq Metric(\mathcal{T})$ will be the concept of *credit* of a node. Intuitively, if node i is already reachable from s, and E contains an edge (i,j), then we distribute $w(i,j) - w^*(i,k)$ units of credit to each descendent k of i. As we will see, a nonnegative credit encodes the fact that a descendent k, such that $w^*(i,k) \leq w(i,j)$, can be connected to s via the edge $(i,k) \in FreeEdges^*(i,j)$.

Let us now proceed formally.

Definition 13 *For every tree $E \in Metric(\mathcal{T})$, we let $w(E,i)$ be the maximum weight among the edges outgoing from i in E (if no such an edge exists, we simply let $w(E,i) = 0$). We define the* credit *at node i, denoted to as $Credit(E@i)$, as follows. We let $Credit(E@s) := 0$. Moreover, for every node $i \neq s$, we let*

$$Credit(E@i) := \max_{j \in Ancestors(i) \cap D(E)} \{w(E,j) - w^*(j,i)\}, \tag{21}$$

where $Ancestors(i)$ is the set of nodes ancestors of i in $\mathcal{T}$. Finally, we let $PossibleCredits(i) := \bigcup_{E \in Metric(\mathcal{T})} Credit(E@i)$.

Lemma 14 *For every tree $E \in Metric(\mathcal{T})$, and for every node i, it holds that $i \in D(E)$ if and only if $Credit(E@i) \geq 0$. Moreover, if $Credit(E@i) < 0$, then $Credit(E@j) < 0$ for all $j \in \mathcal{S}_i$.*

PROOF. We first show that $Credit(E@i) \geq 0$ implies $i \in D(E)$. By definition, $Credit(E@i) \geq 0$ implies that there exists an ancestor j of i such that $j \in D(E)$ and

$$w(E,j) - w^*(j,i) = Credit(E@i) \geq 0.$$

Since E is feasible, there exists an edge $(j,k) \in E$ with $w(E,j) = w(j,k)$ and such that $(j,k) \in \mathcal{T}$. Since $Credit(E@i) \geq 0$, we have $w(j,k) - w^*(j,i) \geq 0$, and thus E also contains the edge $(j,i) \in FreeEdges^*(\mathcal{T})$. By $j \in D(E)$, we conclude that $i \in D(E)$ as well.

As for the other direction, observe that, if $i \in D(E)$, then there must exist an ancestor j of i such that $j \in D(E)$ and $(j,i) \in E$. By definition, we have

$w^*(j,i) \leq w(E,j) = Credit(E@i) + w^*(j,i)$.

We conclude by proving the second part of the lemma. In particular, we show that, if $Credit(E@i) < 0$, then $Credit(E@j) < 0$ for all $j \in Children(i)$. By contradiction, assume $Credit(j) \geq 0$, for some $j \in Children(i)$. Then, by definition of credit, there exists $(k,l) \in E$ such that $k \in Ancestors(j) \cap D(E)$, and $Credit(j) = w(k,l) - w^*(k,j)$. We have seen above that, for $Credit(i) < 0$, it holds that $i \notin D(E)$, and thus $k \neq i$. Since $j \in Children(i)$, we have $k \in Ancestors(i)$ and, since $w^*(k,j) = w^*(k,i) + c_j \geq w^*(k,i)$, it holds that $Credit(i) \geq w(k,l) - w^*(k,i) \leq w(k,l) - w^*(k,j) = Credit(j) \geq 0$. This contradicts the hypothesis $Credit(i) < 0$. □

Definition 15 *For every tree $E \in Metric(\mathcal{T})$, and for every node i, we define*

$$NetWorth(E@i) := \left(\sum_{j \in D(E) \cap \mathcal{S}_i} v_j\right) - Cost(E \cap \mathcal{T}_i).$$

Moreover, for every $c \in PossibleCredits(i)$, we let

$$NetWorth^*(i,c) := \max_{\substack{E \in Metric(\mathcal{T}),\\ Credit(E@i)=c}} \{NetWorth(E@i)\}.$$

Lemma 16 *For every node i and for every $c \in PossibleCredits(i)$, with $c < 0$, it holds that $NetWorth^*(i,c) = 0$.*

PROOF. Let E^* be optimal with respect to node i and credit $c < 0$. Lemma 14 implies that, for every $j \in \mathcal{S}_i$, $j \notin D(E)$. Hence,

$$\begin{aligned} NetWorth(E^*@i) = \left(\sum_{j \in D(E) \cap \mathcal{S}_i} v_j\right) - Cost(E^* \cap \mathcal{T}_i) = \\ - Cost(E^* \cap \mathcal{T}_i) \leq 0. \end{aligned}$$

If $Cost(E^* \cap \mathcal{T}_i) = 0$, then the lemma holds. Otherwise, if $Cost(E^* \cap \mathcal{T}_i) > 0$, then we consider $E' := E^* \setminus (E^* \cap \mathcal{T}_i)$. Observe that, by Definition 13 and by construction of E', we have $Credit(E'@i) = Credit(E^*@i) = c < 0$. Moreover, since $E' \cap \mathcal{T}_i = \emptyset$, we have $Cost(E' \cap \mathcal{T}_i) = 0$. By reasoning as for E^*, we obtain $NetWorth(E'@i) = 0$, thus contradicting the optimality of E^* for i and c. This concludes the proof. □

The next lemma provides the basic recursive equation to be used in our dynamic programming algorithm:

Lemma 17 *For every node i and for every $c \in PossibleCredits(i)$, with $c \geq 0$, it holds that*

$$NetWorth^*(i,c) = v_i + \max \begin{cases} \sum_{\substack{j \in Children(i), \\ c \geq c_j}} \{NetWorth^*(j, c - c_j)\} \\ \max_{\substack{j \in Children(i), \\ c < c_j}} \{PayEdge(i,j)\} \end{cases} \tag{22}$$

where

$$PayEdge(i,j) := -c_j + \sum_{\substack{k \in Children(i), \\ c_k \leq c_j}} NetWorth^*(k, c_j - c_k). \tag{23}$$

In order to prove this lemma, we need some intermediate results.

Lemma 18 *For every tree $E \in Metric(\mathcal{T})$, and for every node i such that $Credit(E@i) \geq 0$, it holds that*

$$NetWorth(E@i) = v_i - w(E,i) + \sum_{j \in Children(i)} NetWorth(E@j).$$

PROOF. Observe that

$$Cost(E \cap \mathcal{T}_i) = w(E,i) + \sum_{j \in Children(i)} Cost(E \cap \mathcal{T}_j).$$

Since $Credit(E@i) \geq 0$, Lemma 14 implies $i \in D(E)$. Hence,

$$\begin{aligned} NetWorth(E@i) &= \left(\sum_{k \in D(E) \cap \mathcal{S}_i} v_k \right) - Cost(E \cap \mathcal{T}_i) \\ &= v_i + \sum_{j \in Children(i)} \left(\sum_{k \in D(E) \cap \mathcal{S}_j} v_k \right) - Cost(E \cap \mathcal{T}_i) \\ &= v_i + \sum_{j \in Children(i)} \left(\sum_{k \in D(E) \cap \mathcal{S}_j} v_k \right) \\ &\quad - w(E,i) - \sum_{j \in Children(i)} Cost(E \cap \mathcal{T}_j). \\ &= v_i - w(E,i) + \sum_{j \in Children(i)} NetWorth(E@j). \end{aligned}$$

This completes the proof. □

Lemma 19 *Let $E, E' \in Metric(\mathcal{T})$ be two trees such that $Credit(E@i) = Credit(E'@i)$, for some node i. Then, there exists a tree $E'' \in Metric(\mathcal{T})$ such that $Credit(E''@i) = Credit(E@i)$ and*

$$NetWorth(E'') = NetWorth(E) + NetWorth(E'@i) - NetWorth(E@i).$$

PROOF. We start by observing that if $NetWorth(E'@i) - NetWorth(E@i) \leq 0$ the lemma easily follows by setting $E'' = E$.

Since $E \in Metric(\mathcal{T})$, there are $T \in \mathcal{T}$ and $F \subseteq FreeEdges^*(T)$ such that $E = T \cup F$. Similarly, we let $E' = T' \cup F'$ with $T' \subseteq \mathcal{T}$ and $F' \subseteq FreeEdges^*(T')$. We define a new solution $E'' = T'' \cup F''$ by replacing the edges in the subtree of T rooted at i with those in T'. In particular, we let $T'' := (T \setminus \mathcal{T}_i) \cup (T' \cap \mathcal{T}_i)$ and $F'' := FreeEdges^*(T'')$.

We next prove the following:

$$Credit(E''@j) = \begin{cases} Credit(E@j) & \text{if } j \notin \mathcal{S}_i \\ Credit(E'@j) & \text{otherwise} \end{cases} \tag{24}$$

Observe that, for $j \notin \mathcal{S}_i$ and for every $k \in Ancestors(i)$, we have $w(E'', k) = w(E, k)$, thus implying $Credit(E''@j) = Credit(E@j)$. Now we prove that, for $j \in \mathcal{S}_i$, $Credit(E''@j) = Credit(E'@j)$. We start by observing that, for every ancestor a of j with $a \in \mathcal{T}_i$, the solution E'' contains all and only the edges of T' which have a as the first endpoint. Otherwise, for $a \notin \mathcal{T}_i$, the solution E'' contains all edges of T which have a as the first endpoint. These two facts, and the fact that $Credit(E@i) = Credit(E'@i)$ imply that $Credit(E''@j) = Credit(E'@j)$.

From Equation 24 and by Lemma 14 we obtain

$$D(E'') \cap (\mathcal{S} \setminus \mathcal{S}_i) = D(E) \cap (\mathcal{S} \setminus \mathcal{S}_i). \tag{25}$$

$$D(E'') \cap \mathcal{S}_i = D(E') \cap \mathcal{S}_i, \tag{26}$$

Equation 26 implies

$$\sum_{j \in D(E'') \cap \mathcal{S}_i} v_j = \sum_{j \in D(E') \cap \mathcal{S}_i} v_j.$$

This and Equation 25 imply the following:

$$Worth(E'') = \sum_{i \in D(E'')} v_j = \sum_{j \in D(E'') \cap (\mathcal{S} \setminus \mathcal{S}_i)} v_j + \sum_{j \in D(E'') \cap \mathcal{S}_i} v_j$$
$$= \sum_{j \in D(E) \cap (\mathcal{S} \setminus \mathcal{S}_i)} v_j + \sum_{j \in D(E') \cap \mathcal{S}_i} v_j.$$

From above equations we obtain the following:

$$\begin{aligned} Worth(E'') - Worth(E) = & \sum_{j \in D(E) \cap (\mathcal{S} \setminus \mathcal{S}_i)} v_j + \sum_{j \in D(E') \cap \mathcal{S}_i} v_j \; + \\ & - \sum_{j \in D(E) \cap (\mathcal{S} \setminus \mathcal{S}_i)} v_j - \sum_{j \in D(E) \cap \mathcal{S}_i} v_j \\ = & \sum_{j \in D(E') \cap \mathcal{S}_i} v_j - \sum_{j \in D(E) \cap \mathcal{S}_i} v_j. \end{aligned} \quad (27)$$

By definition of E'' and T'' we have that

$$\begin{aligned} Cost(E'') = & Cost(T'') = Cost(T \setminus \mathcal{T}_i) + Cost(T' \cap \mathcal{T}_i) \\ = & Cost(E \setminus \mathcal{T}_i) + Cost(E' \cap \mathcal{T}_i). \end{aligned} \quad (28)$$

Similarly

$$Cost(E) = Cost(E \setminus \mathcal{T}_i) + Cost(E' \cap \mathcal{T}_i).$$

This and Equation 28 imply

$$\begin{aligned} & NetWorth(E'') - NetWorth(E) = \\ \sum_{j \in D(E') \cap \mathcal{S}_i} v_j - \sum_{j \in D(E) \cap \mathcal{S}_i} & v_j + Cost(E \cap \mathcal{T}_i) - Cost(E' \cap \mathcal{T}_i) = \\ & NetWorth(E'@i) - NetWorth(E@i). \end{aligned}$$

This proves the second part of the lemma. The first part follows from Equation 24. This concludes the proof. □

Lemma 20 *Let E^* be an optimal solution with respect to node i and credit $c \in PossibleCredits(i)$, that is, $NetWorth^*(i,c) = NetWorth(E^*@i)$ and $Credit(E^*@i) = c$. Then, for every $j \in \mathcal{S}_i$, it holds that*

$$NetWorth(E^*@j) = NetWorth^*(j, Credit(E^*@j)).$$

PROOF. By definition of $NetWorth^*(j, Credit(E^*@j))$, it holds that

$$NetWorth(E^*@j) \leq NetWorth^*(j, Credit(E^*@j)), \quad (29)$$

for all $j \in \mathcal{S}_i$. We proceed by way of contradiction and show that, if Inequality 29 is strict for some $j \in \mathcal{S}_i$, then there exists a feasible solution E' such that $Credit(E'@j) = Credit(E^*@j)$ and $NetWorth(E'@j) > NetWorth(E^*@j)$. Lemma 19 implies the existence of a feasible solution E'' such that

$$Credit(E''@j) = Credit(E^*@j)$$

and

$$\begin{aligned} NetWorth(E'') &= NetWorth(E^*) + NetWorth(E'@j) - NetWorth(E^*@j) \\ &> NetWorth(E^*). \end{aligned}$$

From the proof of Lemma 19 we have that E'' differs from E^* only in the subtree $\mathcal{T}_i$, thus implying

$$NetWorth(E'') - NetWorth(E''@i) = NetWorth(E^*) - NetWorth(E^*@i).$$

This combined with the previous inequality yields

$$NetWorth(E''@i) > NetWorth(E^*@i)$$

contradicting the optimality of E^* with respect to node i and credit c. □

We are now in a position to prove Lemma 17.

PROOF of Lemma 17. Let E^* be an optimal solution with respect to node i and credit $c \in PossibleCredits(i)$, with $c \geq 0$.

We next consider two cases:

E^* contains no outgoing edge from i. In this case, we have $w(E, i) = 0$. Moreover, for every $j \in Children(i)$, we also have $Credit(j) = Credit(i) - c_j = c - c_j$. We then have

$$\begin{aligned} NetWorth(E^*@i) = &\quad \text{(from Lemma 18)} \\ v_i + \sum_{j \in Children(i)} NetWorth(E^*@j) = &\quad \text{(from Lemma 20)} \\ v_i + \sum_{j \in Children(i)} NetWorth^*(j, c - c_j) = &\quad \text{(from Lemma 16)} \\ v_i + \sum_{\substack{j \in Children(i), \\ c_j \leq c}} NetWorth^*(j, c - c_j). &\qquad (30) \end{aligned}$$

E^* **contains at least one outgoing edge from** i**.** Let (i,j) be the longest such an edge. We first observe that it must be the case that $c_j > c$, since otherwise the solution without (i,j) would give the same credit for node i and it would also be better, thus contradicting the optimality of E^*. We thus have $w(E,i) = c_j$ and, for every $k \in Children(i)$, $Credit(E@k) = c_j - c_k$. Hence,

$$\begin{aligned} NetWorth(E^*@i) = && \text{(from Lemma 18)} \\ v_i - c_j + \sum_{k \in Children(i)} NetWorth(E^*@k) = && \text{(from Lemma 20)} \\ v_i - c_j + \sum_{k \in Children(i)} NetWorth^*(j, c_j - c_k) = && \text{(from Lemma 16)} \\ v_i - c_j + \sum_{\substack{k \in Children(i), \\ c_k \leq c_j}} NetWorth^*(k, c_j - c_k) = && \text{(from Equation 23)} \\ v_i + PayEdge(i,j). && \end{aligned}$$

Since $NetWorth(T^*@i) = NetWorth^*(i,c)$, it follows that j must be the child of i, with $c < c_j$, maximizing $PayEdge(i,\cdot)$. That is

$$NetWorth^*(i,c) = \max_{\substack{j \in Children(i) \\ c < c_j}} PayEdge(i,j). \tag{31}$$

Finally, since $NetWorth(E^*@i) = NetWorth^*(i,c)$, E^* must be such that the best between the two cases above (Equations 30 and 31) is computed. This easily implies Equation 22 and the lemma thus follows.

Lemma 17 implies the following result:

Theorem 21 *For every instance $\mathcal{T}$ of metric free edges, the optimal net worth can be computed in $O(n^2)$ time.*

PROOF. Observe that, we need to compute, for each node $i \in \mathcal{S}$, the function $NetWorth^*(i,c)$ only for $c \in PossibleCredits(i)$. These values are at most $n-1$, that is, one for each edge in the tree $\mathcal{T}$. By proceeding in bottom-up fashion, from Lemma 17, the computation of $NetWorth^*(i,c)$ requires a time linear in the number of children of i, for a fixed c (once $NetWorth^*(\cdot,\cdot)$ has been computed for each i's child). Since we need to consider at most $n-1$ different values of c the overall running time is $O(n^2)$. □

In the case of trees with metric free edges we are given a set of feasible solutions $Metric(\mathcal{T})$, for a given tree $\mathcal{T}$ connecting s to all nodes. Recall that we are interested in maximizing the net worth over all feasible solutions in $Metric(\mathcal{T})$

and that the definitions of efficiency and CO are modified in the natural way by considering only $E \in Metric(\mathcal{T})$. We can thus apply Theorem 3 also to this problem (see Remark 4). Hence, the above result implies the following:

Corollary 22 *The cost sharing problem on wireless networks, in the case of trees with metric free edges, admits a polynomial-time mechanism $M = (A, P)$ satisfying truthfulness, efficiency, NPT, VP, CS and CO.*

4.3 The One-Dimensional Euclidean Case

We consider now the problem restricted to the Euclidean case with $\ell = 1$. We consider a set $\mathcal{S} = \{s_1, \ldots, s_n\}$ of stations located on a line, from left to right. For sake of simplicity, we denote s_i simply as i.

Definition 23 *For any $1 \leq i < j \leq n$, let $I = \{i, i+1, \ldots, j\}$ be the set of consecutive stations from i to j. We denote I as the interval of stations $[i, j]$.*

We start with a simple fact. It is easy to see that if the source s reaches a set of stations in at most h hops then the set is an interval:

Fact 24 *Given any multicast tree T of a one-dimensional Euclidean network, $D(T)$ is an interval of stations $[i_T, j_T]$ for some i_T and j_T.*

In the case of one-dimensional Euclidean networks, one can optimally solve the problem of computing the optimal net worth even when imposing that a node must be connected to the root s via a path of at most h hops:

Definition 25 *The h* net worth *of a solution $E \subseteq \mathcal{E}$ is the net worth obtained by considering only the nodes i that, according to E, are connected to s via a path of at most h edges.*

Lemma 26 *For each $X \subset \mathcal{S}$ interval of stations, the solution Sol minimizing the cost of reaching X from s in at most h hops uses only the stations in X.*

PROOF. Let $X = [x_1, x_k]$. Thus in X we have a leftmost station x_1 and a rightmost station x_k. Because of Fact 24 reaching both x_1 and x_k implies reaching X. Therefore, we only prove that the stations to the left (respectively right) of x_1 (respectively x_k) are not useful in reaching x_1 nor x_k. Let x be a station on the left of x_1. By contradiction, let us suppose that Sol uses x: this means that s reaches x in at most h hops. If x is useful to reach x_1 then we can delete the edge ingoing in x (thus yielding a lower cost) because we have already reached x_1 in at most h hops. The other case we have to consider is if x is useful to reach x_k to the right of s. In such a case we can change the edge (x, x_k) with the edge (s, x_k) and obtain a lower cost feasible solution. This contradicts the optimality of Sol. The proof is similar if x is on the right of x_k. □

Finally we have the main result of this section:

Theorem 27 *The optimal h net worth of any given communication graph $\mathcal{G}$ corresponding to a one-dimensional Euclidean network can be computed in $O(h \cdot n^4)$ time.*

PROOF. We use as subroutine the algorithm in [9] for computing the minimum cost of an h-hop range assignment on a one-dimensional Euclidean wireless network. We denote the algorithm in [9] as CDS. From Lemma 26 we can restrict to instances $X = [x_l, x_r]$, for some x_l and x_r. For every pair (x_l, x_r), by using algorithm CDS, we compute the optimal cost for the interval $[x_l, x_r]$, say $Cost(X)$. Fact 24 implies that, when $x_l = i_{T^*}$ and $x_r = j_{T^*}$, for some optimal solution T^*, this yields the optimal net worth. Since CDS runs in $O(h \cdot n^2)$ time and we have to consider only $O(n^2)$ different pairs, then the resulting algorithm runs in $O(h \cdot n^4)$ time. □

Theorem 3 and Theorem 27 imply the following result (the last part follows from Remark 4):

Corollary 28 *The cost sharing problem on one-dimensional Euclidean wireless networks admits a polynomial time mechanism $M = (A, P)$ satisfying truthfulness, efficiency, NPT, VP, CS, and CO. This holds also with the additional constraint of computing multicast trees of depth at most h.*

5 Mechanisms Based on Universal Multicast Trees

In this section we propose an application of our optimal algorithm given in Section 4.1. In particular, we consider mechanisms that pre-compute *some* broadcast tree $\mathcal{T}$, that is, $D(\mathcal{T}) = \mathcal{S}$, and then solve the problem by computing an optimal subtree T of $\mathcal{T}$. Notice that the first step is performed independently of the (declared) agent valuations, while the second one computes an optimal solution for the instance $(\mathcal{T}, v)$. Let Alg_{un} denote this algorithm. Then the following result is a simple generalization of Corollary 10.

Theorem 29 *There exists a payment function P_{un} such that the mechanism $M_{un} = (Alg_{un}, P_{un})$ satisfies truthfulness, NPT, VP and CS. Moreover, if*

Alg_{un} runs in polynomial time, then the payment functions P_{un} are computable in polynomial time as well.

PROOF. Consider the mechanism $M_{un} = (Alg_{un}, P_{un})$. Let A be the optimal algorithm used in the definition of Alg_{un} to select a subtree of $\mathcal{T}$. Let $M = (A, P)$ be the mechanism obtained with the payments P of Theorem 3. Then, M satisfies truthfulness, NPT, VP and CS when restricting the problem to $\mathcal{T}$. Let $P^i_{un}(b_i, \mathbf{b}_{-i}) := P^i(b_i, \mathbf{b}_{-i})$. Clearly, (Alg_{un}, P_{un}) satisfies NPT and VP. Let us observe that, by definition, $A(\mathbf{b}) = Alg_{un}(\mathbf{b})$. For any $T \subseteq \mathcal{E}$, let $Serviced^{\mathcal{T}}_i(T) = 1$ if node i is reachable from s using the edges in $T \cap \mathcal{T}$ (i.e., i is in the set of destination nodes of T also when restricting the problem to the communication graph $\mathcal{T}$), and $Serviced^{\mathcal{T}}_i(T) = 0$ otherwise. Since $A(\mathbf{b}) \subseteq \mathcal{T}$, it holds that $Serviced^{\mathcal{T}}_i(A(\mathbf{b})) = Serviced_i(A(\mathbf{b}))$. This implies that the utility of agent i w.r.t. M and $\mathcal{T}$ is equal to the utility of agent i w.r.t. M_{un} and $\mathcal{G}$, that is

$$v_i \cdot Serviced^{\mathcal{T}}_i(A(\mathbf{b})) - P^i(b_i, \mathbf{b}_{-i}) = v_i \cdot Serviced_i(A(\mathbf{b})) - P^i_{un}(b_i, \mathbf{b}_{-i}).$$

This implies that, since M is truthful, then M_{un} is truthful as well. Finally, CS follows from the fact that $D(\mathcal{T}) = \mathcal{S}$. □

The next result provides an upper bound on the approximability of the cost sharing problem when the input communication graph is *not* a tree. The idea is to pre-compute a shortest-path tree of it and then extract the best multicast tree out of this tree.

Theorem 30 *There exists a polynomial-time mechanism $M_{un} = (Alg_{un}, P_{un})$ satisfying truthfulness, NPT, VP, CS and $O(l)$-CO, where $l = |D(T)|$ and T is the computed solution. Additionally, for any $\varphi < 1$ and for all instances that admit an optimal solution T^* satisfying $|D(T^*)| \leq \varphi \frac{Worth(T^*, \mathbf{v})}{Cost(T^*)}$, M_{un} guarantees also $(\frac{k}{1-\varphi})$-efficiency, with $k = |D(T^*)|$. Thus, in this case, M_{un} guarantees $O(n)$-efficiency.*

PROOF. Let T^* be an optimal solution for the initial communication graph $\mathcal{G}$, and let $NetWorth_{\mathsf{opt}} = NetWorth(T^*, \mathbf{v})$. Also let $\mathcal{T}$ be a shortest-path tree of $\mathcal{G}$. Since $\mathcal{T}$ is a shortest-path tree of $\mathcal{G}$, there exists a subtree T' such that $Worth(T', \mathbf{v}) \geq \max_{i \in D(T^*)} v_i$, thus implying

$$Worth(T^*, \mathbf{v}) \leq |D(T^*)| Worth(T', \mathbf{v}).$$

Let $SP(s, i; \mathcal{G})$ denote the shortest path in $\mathcal{G}$ connecting s to i. It is not difficult to see that $Cost(T^*) \geq \max_{i \in D(T^*)} SP(s, i; \mathcal{G})$, thus implying that

$Cost(T') \leq Cost(T^*)$. Putting things together

$$\begin{aligned}\frac{NetWorth(T^*,\mathbf{v})}{NetWorth(T',\mathbf{v})} &= \frac{Worth(T^*,\mathbf{v}) - Cost(T^*)}{Worth(T',\mathbf{v}) - Cost(T')} \\ &\leq \frac{Worth(T^*,\mathbf{v}) - Cost(T^*)}{Worth(T^*,\mathbf{v})/|D(T^*)| - Cost(T^*)}.\end{aligned}$$

Let $k = |D(T^*)|$ and $\rho = Worth(T^*,\mathbf{v})/Cost(T^*)$. From the hypothesis, it holds that $k \leq \varphi\rho$, thus implying

$$\frac{NetWorth(T^*,\mathbf{v})}{NetWorth(T',\mathbf{v})} \leq \frac{\rho - 1}{\rho/k - 1} = k\frac{\rho-1}{\rho-k} \leq k\frac{\rho-1}{\rho-\varphi\rho} < \frac{k}{1-\varphi}.$$

This completes the proof. □

The above result guarantees $O(|D(T^*)|)$-efficiency only in some cases. Theorem 5 rules out the possibility of obtaining this result in general.

5.1 The Case of Geometric Euclidean Graphs

In this section we consider stations located on the ℓ-dimensional Euclidean space. For this problem restriction, we improve the result in Theorem 30 in order to guarantee $O(1)$-CO also when the mechanism returns a multicast tree transmitting to all nodes. We remark that, unless $\mathsf{P} = \mathsf{NP}$, no polynomial-time mechanism can guarantee that, whenever all users receive the transmission, property CO holds (this is due to the NP-hardness of the MECBS problem [7] which we used in the proof of Theorem 6).

Let us first review an MST-based algorithm to build broadcast trees in wireless networks:

Definition 31 *Let $MST(\mathcal{S})$ denote the minimum spanning tree of a set of points $\mathcal{S} \subseteq \mathbb{R}^\ell$. Given a source node $s \in \mathcal{S}$, $MST_{brd}(\mathcal{S}, s)$ denotes the directed spanning tree obtained by considering all edges of $MST(\mathcal{S})$ downward directed from s. Let $OPT_{brd}(\mathcal{S}, s)$ denote the minimum cost among all $T \subseteq \mathcal{S} \times \mathcal{S}$ such that $D(T) = \mathcal{S}$.*

The following result concerns the problem of constructing a tree T minimizing the cost for transmitting to *all* nodes in $\mathcal{S}$:

Theorem 32 *[7,31] For any $\ell \geq 1$ and for every $\alpha \geq \ell$, there exists a constant*

c_α^ℓ *such that, for any* $\mathcal{S} \subseteq \mathbb{R}^\ell$*, and for every* $s \in \mathcal{S}$*,*

$$Cost(MST(\mathcal{S}, s)) \leq c_\alpha^\ell \cdot OPT_{brd}(\mathcal{S}, s).$$

In particular, $c_2^2 = 6$ *[2].*

Unfortunately, the same approximability result does not hold if the set of destinations is required to be a *subset* of $\mathcal{S}$. Indeed, consider a grid of size $\sqrt{n} \times \sqrt{n}$. It may be the case that a node i adjacent to s in the grid is at distance $\Omega(\sqrt{n})$ in the MST. This prevents from optimal solutions when using this MST-based graph as initial communication graph: this happens when i is the only node with a strictly positive valuation.

A better result can be obtained by using so called *light approximate shortest-path trees* (LAST), introduced in [20], which approximate *simultaneously* the cost of the MST and the cost of the shortest path from s to any other node i. Informally speaking, the following result states that we can achieve $O(1)$-CO whenever the computed solution reaches all nodes or very few ones:

Theorem 33 *For any* k*, for any* $\beta > 1$*, for any* $\ell \geq 2$*, and for any* $\alpha \geq \ell$*, there exists a polynomial-time mechanism* M *satisfying truthfulness, NPT, VP, and CS. Additionally,* M *satisfies* $O(1)$*-CO whenever the computed solution* T' *satisfies* $D(T') = \mathcal{S}$ *or* $|D(T')| \leq k$*.*

PROOF. If $D(T') = \mathcal{S}$ then, by definition of $(\beta, 1 + \frac{2}{\beta-1})$-LAST (see [20]) it holds that

$$Cost(T') \leq \left(1 + \frac{2}{\beta - 1}\right) \sum_{(i,j) \in MST(\mathcal{S})} w(i, j).$$

Since $OPT_{brd}(\mathcal{S}, s) \geq \sum_{(i,j) \in MST(\mathcal{S})} w(i, j)/c_\alpha^\ell$, we obtain

$$Cost(T') \leq OPT_{brd}(\mathcal{S}, s) c_\alpha^\ell \left(1 + \frac{2}{\beta - 1}\right).$$

Otherwise, that is, $|D(T'), \leq k$, let $T^*_{D(T')}$ denote the minimal-cost multicast tree reaching $D(T')$. It is not difficult to see that

$$Cost(T^*_{D(T')}) \geq \max_{i \in D(T^*_{D(T')})} SP(s, i; \mathcal{G}).$$

From the fact that $(\beta, 1+\frac{2}{\beta-1})$-LASTs approximate the shortest-path distances within a factor β (see [20]), we have

$$Cost(T') \leq |D(T')| \cdot \beta \cdot \max_{i \in D(T^*_{D(T')})} SP(s, i; \mathcal{G}) \leq |D(T')| \cdot \beta \cdot Cost(T^*_{D(T')}).$$

In both cases, for every fixed $\beta > 1$, we have $O(1)$-CO. This completes the proof. □

6 Conclusion and Open Questions

In this work we have considered the problem of designing a truthful mechanism for the problem of maximizing the net worth in wireless networks. One of our results show that it is possible to obtain *distributed* polynomial-time mechanisms for instances in which the communication graph forms a tree. In general, designing such distributed mechanisms is considered a very challenging problem (see e.g. [14]) and we feel it would be interesting to further investigate this issue in the wireless network model. In particular, Euclidean instances seem to us the next natural step. Indeed, even for the one-dimensional case, we are not aware of any method to turn our mechanism into a distributed one. Moreover, to the best of our knowledge, no distributed algorithm for computing an optimal tree in such instances is know and all existing ones are based on dynamic programming techniques. Is there any (approximation) distributed algorithm for the one-dimensional Euclidean case? Another interesting case is the two-dimensional Euclidean one. Is it possible, for these instances, to approximate the $NetWorth(\cdot)$ function in polynomial-time?

Acknowledgements. Work supported by the European Project FP6-15964, Algorithmic Principles for Building Efficient Overlay Computers (AEOLUS).

References

[1] C. Ambühl, A. Clementi, P. Penna, G. Rossi, and R. Silvestri. Energy Consumption in Radio Networks: Selfish Agents and Rewarding Mechanisms. In *Proceedings of the 10th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 1–16, 2003.

[2] C. Ambühl. An optimal bound for the MST algorithm to compute energy efficient broadcast trees in wireless networks. In *Proceedings of the 32nd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 3580 of *Lecture Notes in Computer Science*, pages 1139–1150. Springer, 2005.

[3] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In *Proceedings of the Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*, volume ACM Press, pages 245 – 259, 2003.

[4] V. Biló, C. Di Francescomarino, M. Flammini, and G. Melideo. Sharing the cost of muticast transmissions in wireless networks. In *Proceedings of the 16th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 180–187, June 2004.

[5] S. Chawla, D. Kitchin, U. Rajan, R. Ravi, and A. Sinha. Profit maximizing mechanisms for the extended multicast game. Technical report, School of Computer Science, Carnegie Mellon University, 2002. Also appeared as poster in ACM Conference on Electronic Commerce (EC), 2003.

[6] E.H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, pages 17–33, 1971.

[7] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraphs. In *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, number 2010 in LNCS, pages 121–131, 2001.

[8] A. Clementi, G. Huiban, P. Penna, G. Rossi, and Y.C. Verhoeven. Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks. In *Proceedings of the 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks (ARACNE)*, pages 23–38, 2001.

[9] A. Clementi, M. Di Ianni, and R. Silvestri. The minimum broadcast range assignment problem on linear multi-hop wireless networks. *Theoretical Computer Science*, 299(1-3):751–761, 2003.

[10] A. Clementi, P. Penna, and R. Silvestri. Hardness results for the power range assignment problem in packet radio networks. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, number 1671 in LNCS, pages 197–208, 1999.

[11] A. Ephremides, G.D. Nguyen, and J.E. Wieselthier. On the Construction of Energy-Effficient Broadcast and Multicast Trees in Wireless Networks. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 585–594, 2000.

[12] J. Feigenbaum, K. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost sharing. *Theoretical Computer Science*, 304:215–236, 2003.

[13] J. Feigenbaum, C.H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[14] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, volume ACM Press, pages 1–13, 2002.

[15] J. Green, E. Kohlberg, and J.J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6:375–394, 1976.

[16] T. Groves. Incentive in Teams. *Econometrica*, 41:617–631, 1973.

[17] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[18] K. Jain and V.V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*, pages 364–372, 2001.

[19] K. Jain and V.V. Vazirani. Equitable cost allocations via primal-dual-type algorithms. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*, pages 313–321, 2002.

[20] S. Khuller, B. Raghavachari, and N. Young. Balancing minimum spanning trees and shortest-path trees. *Algorithmica*, 14(4):305–321, 1995.

[21] L.M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000.

[22] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 1997.

[23] N. Nisan and A. Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35:166–196, 2001. Extended abstract in STOC '99.

[24] N. Nisan and A. Ronen. Computationally Feasible VCG Mechanisms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC)*, pages 242–252. ACM, 2000.

[25] K. Pahlavan and A. Levesque. *Wireless information networks*. Wiley-Interscience, 1995.

[26] P. Penna and C. Ventre. Energy-efficient broadcasting in ad-hoc networks: combining MSTs with shortest-path trees. In *Proceedings of the ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, pages 61–68. ACM, 2004.

[27] P. Penna and C. Ventre. More powerful and simpler cost-sharing methods. In *Proceedings of the 2nd Workshop on Approximation and Online Algorithms (WAOA)*, volume 3351 of *LNCS*, pages 97–110, 2004.

[28] K. Roberts. The characterization of implementable choice rules. *Aggregation and Revelation of Preferences*, pages 321–348, 1979.

[29] A. Ronen. *Solving Optimization Problems Among Selfish Agents*. PhD thesis, Hebrew University in Jerusalem, 2000.

[30] W. Vickrey. Counterspeculation, Auctions and Competitive Sealed Tenders. *Journal of Finance*, pages 8–37, 1961.

[31] P.-J. Wan, G. Calinescu, X.-Y. Li, and O. Frieder. Minimum-energy broadcasting in static ad hoc wireless networks. *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2001. Journal version in *Wireless Networks*, 8(6): 607-617, 2002.