More Powerful and Simpler Cost-Sharing Methods*

(When Cross-Monotonicity Is the Wrong Way)

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
{penna, ventre}@dia.unisa.it

Abstract. We provide a new technique to derive *group strategyproof* mechanisms for the cost-sharing problem. Our technique is simpler and provably more powerful than the existing one based on so called *cross-monotonic* cost-sharing methods given by Moulin and Shenker [1997]. Indeed, our method yields the first *polynomial-time* mechanism for the Steiner tree game which is group strategyproof, *budget balance* and also meets other standard requirements (No Positive Transfer, Voluntary Participation and Consumer Sovereignty). A known result by Megiddo [1978] implies that this result cannot be achieved with cross-monotonic cost-sharing methods, even if using exponential-time mechanisms.

1 Introduction

Consider a service providing company $\mathtt{P}$ with a set of possible customers, also called *users*, U. For each subset $S \subseteq U$ of users, $C_{\mathtt{P}}(S)$ denotes the cost incurred by the company $\mathtt{P}$ to *jointly* service the users in S. The function $C_{\mathtt{P}}(\cdot)$ is usually termed the *cost function*. A typical scenario is that of company $\mathtt{P}$ broadcasting some kind of transmission (e.g., movies, sport events, news, etc) over a given network: in this case, $C_{\mathtt{P}}(S)$ is the cost of implementing a multicast tree connecting a source node s to all users is S. Each user i valuates the transmission an amount v_i: this value quantifies how much user i likes the transmission (or how much he/she would pay for it). A key point is that v_i is a property of user i (and not of the network) and, thus, this value is known to i only. If user i is required to pay p_i for receiving the transmission, then her *utility* is equal to $v_i - p_i$. The utility is naturally what each user i tries to maximize. Users may act *selfishly* and, thus, a user i may misreport her valuation at some other number b_i. (Consider a simple mechanism which charges to every user i an amount equal to her *reported* valuation b_i).

* Work supported by the European Project IST-2001-33135, Critical Resource Sharing for Cooperation in Complex Systems (CRESCCO).

G. Persiano and R. Solis-Oba (Eds.): WAOA 2004, LNCS 3351, pp. 97–110, 2005.

A *cost-sharing mechanism* should decide which user S should receive the transmission and at which price. The mechanism is said *strategyproof* if, for each user i, revealing the true value v_i is a dominant strategy: that is, reporting any $b_i \neq v_i$ cannot improve the utility of i (see Sect. 2 for a formal definition). The mechanism is *group strategyproof* if this holds also for coalitions of users. The mechanism is *budget balance* if the total amount of money payed by the users equals the servicing cost $C_{\mathsf{P}}(S)$. Finally, a mechanism is *efficient* if it maximizes, over all subsets S, the sum of the valuations of users in S minus the cost $C_{\mathsf{P}}(S)$.

A fundamental result by Moulin and Shenker [8, 7] shows that the existence of a so called *cross-monotonic cost-sharing method* (see Sect. 2 for a formal definition) for $C_{\mathsf{P}}(\cdot)$ gives rise to a group strategyproof and budget balance mechanism. Moreover, if $C_{\mathsf{P}}(\cdot)$ is submodular (see the definition in Sect. 2.1), the converse holds as well. These results point all in the direction of cross-monotonic cost-sharing methods: on the one hand, no other technique is known to derive such mechanisms; on the other hand, the "converse" part of Moulin and Shenker's theorem says that, for submodular functions, these type of mechanisms capture all possible ones.

Unfortunately, meeting the cross-monotonicity requirement is often far from trivial: some (optimal) cost functions do not admit such methods [6]; others require a rather involved use of primal-dual algorithms [3, 9, 5, 1]; for others, only some sort of approximation of the budget balance condition is guaranteed (e.g., the mechanism may create some surplus or recover only a fraction of the cost) [9, 5, 1].

In this work we provide a more powerful method to derive such mechanisms by introducing the concept of *self cross-monotonic* cost-sharing method (see Sect. 3). Our main result is that, given any such cost-sharing method, it is possible to obtain group strategyproof mechanisms. The resulting technique extends the one by Moulin and Shenker and is provably more powerful: it indeed applies to some optimal cost functions for which the method by Moulin and Shenker cannot be used and/or gives simpler constructions of the mechanisms (see Sect.s 2.1 and 2.2 for a more detailed discussion of previous and our results).

2 Model

We are given a set U of n users. Depending on the problem instance at hand, for every $Q \subseteq U$, and for every feasible solution T_Q which allows to provide the service to users in Q (e.g., a multicast tree connecting a source node s to all users in Q), we denote by $\mathsf{COST}(T_Q)$ the cost of this solution.[1] Hence, for a service providing company P that decides to service Q by implementing solution T_Q, we have a cost $C_{\mathsf{P}}(Q) = \mathsf{COST}(T_Q)$.

[1] A formal definition should be $\mathsf{COST}(Q, G)$ since the cost depends on the instance. However, for the sake of clarity, we will omit 'G' whenever this will be clear from the context.

Each user is a *selfish agent* reporting some (not necessarily true) valuation b_i; the true value v_i is *privately known* to agent i. Based on the *reported* values $b = (b_1, b_2, \ldots, b_n)$ a *mechanism* $M = (A, P)$ uses algorithm A to compute the following:

– A subset $Q(b) \subseteq U$ of users to be serviced;
– A feasible solution $T_{Q(b)}$ to be implemented in order to provide the service to the set $Q(b)$; solution $T_{Q(b)}$ does *not* provide the service to any $j \notin Q(b)$.

For the sake of convenience, one can imagine that an algorithm $\mathsf{A}(\cdot)$ is used by M in order to compute $T_{Q(b)}$ once a set $Q(b)$ has been selected, that is, $T_{Q(b)} = \mathsf{A}(Q(b))$. (For instance, $\mathsf{A}(\cdot)$ may be a multicast algorithm computing a tree connecting a source node s to a subset Q of the nodes of a network.) In this case, we let $C_{\mathsf{A}}(Q(b)) := \mathsf{COST}(\mathsf{A}(Q(b)))$.

In addition, for every user $i \in Q(b)$, the mechanism computes the cost $P^i(b)$ that user i must pay for getting the service, with $P = (P^1, P^2, \ldots, P^n)$. Hence, the *utility* of agent i when she reports b_i, and the other agents report $b_{-i} := (b_1, \ldots, b_{-i}, b_{i+1}, \ldots, b_n)$, is equal to

$$u_i(b_i, b_{-i}) := v_i \cdot \sigma_i(Q(b_i, b_{-i})) - P^i(b_i, b_{-i}),$$

where $(x, b_{-i}) = (b_1, \ldots, b_{i-1}, x, b_{i+1}, \ldots, b_n)$ and $\sigma_i(X)$ equals 1 if $i \in X$, and 0 otherwise. In the sequel, for every $C \subseteq U$ and any two vectors x and y of length n, (x_C, y_{-C}) denotes the vector $z = (z_1, \ldots, z_n)$ such that $z_i = x_i$ if $i \in C$ and $z_i = y_i$ if $i \notin C$.

There is a number of natural constraints/goals that, for every problem instance G, a mechanism $M = (A, P)$ should satisfy/meet:

1. **Cost Optimality (CO).** Let $C_{\mathsf{opt}}(Q)$ denote the minimum cost required to service all users in Q, for every $Q \subseteq U$. We require that the computed solution $T_{Q(b)}$ is optimal w.r.t. the set $Q(b)$, that is, $C_{\mathsf{A}}(Q(b)) = C_{\mathsf{opt}}(Q(b))$.
2. **No Positive Transfer (NPT).** No user receives money from the mechanism, i.e., $P^i(\cdot) \geq 0$.
3. **Voluntary Participation (VP).** We never charge an user an amount of money grater than her *reported* valuation, that is, $\forall b_i, \forall b_{-i} \;\; b_i \geq P^i(b_i, b_{-i})$. In particular, a user has always the option to not paying for a service for which she is not interested. Morever, $P^i(b) = 0$, for all $i \notin Q(b)$, i.e., only the users getting the service will pay.
4. **Consumer Sovereignty (CS).** Every user is guaranteed to get the service if she reports a high enough valuation.
5. **Budget Balance (BB).**
   (a) **Cost recovery.** $\sum_{i \in Q(b)} P^i(b) \geq C_{\mathsf{A}}(Q(b))$, i.e., the cost of the computed solution is recovered from all the users being serviced;
   (b) **Competitiveness.** $\sum_{i \in Q(b)} P^i(b) \leq C_{\mathsf{A}}(Q(b))$, i.e., no surplus is created. If some surplus were created, then a competitor may offer the same service at a better price.

6. **Group Strategyproofness.** We require that a user $i \in U$ that misreport her valuation (i.e., $b_i \neq v_i$) cannot improve her utility (strategyproofness or truthfulness) nor improve the utility of other users without worsening her own utility (otherwise, a coalition C containing i would secede). Consider a coalition $C \subseteq U$ of users. Let $b_i = v_j$ for all $j \notin C$. The group strategyproofness requires that if the inequality

$$v_i \cdot \sigma_i(Q(b_C, v_{-C})) - P^i(b_C, v_{-C}) \geq v_i \cdot \sigma_i(Q(v_C, v_{-C})) - P^i(v_C, v_{-C}) \quad (1)$$

holds for all $i \in C$ then it must hold with equality for all $i \in C$ as well.

A *cost-sharing method* is a function $\xi(\cdot)$ which distributes the cost $C_{\mathsf{A}}(\cdot)$ to the users that get the service. Intuitively speaking, we will use the function $\xi(\cdot)$ in order to define the payments $P^i(\cdot)$. More formally, $\xi(\cdot)$ takes two arguments: a set of users Q and a user i and returns a *nonnegative* real number satisfying the following:

$$\text{if } i \notin Q \text{ then } \xi(Q, i) = 0 \text{ and} \quad (2)$$

$$\sum_{i \in Q} \xi(Q, i) = C_{\mathsf{A}}(Q). \quad (3)$$

Observe that, if we take $P^i(b) := \xi(Q(b), i)$, then the payments recover exactly the cost $C_{\mathsf{A}}(Q(b))$ from all and only users in $Q(b)$. Also the NPT condition holds. The other requirements depend on how the mechanism selects $Q(b)$ and $T_{Q(b)}$.

In the context of *multicast routing*, we are given a weighted undirected graph $G = (U \cup \{s\}, E, c)$, where $s \notin U$ is the source node and c_e is the cost of using link $e \in E$. A feasible solution is a pair $T_Q = (Q, T)$, where T is a tree connecting s to a subset Q of users contained in T. The corresponding cost is the weight of T, i.e., $\sum_{e \in T} c_e$. The optimal cost function $C_{\mathsf{opt}}(Q)$ to service Q is the cost of an optimal Steiner tree of G connecting s to Q, thus possibly containing some Steiner nodes in $U \setminus Q$. This is the *Steiner tree game* and we let $\sigma_i(T_Q) = 1$ if and only if $i \in Q$. In the *minimum spanning tree game* the feasible solution is any spanning tree T_Q containing s and the set Q only.

Approximation Concepts. The use of optimal cost functions $C_{\mathsf{opt}}(\cdot)$ for the given problem may suffer from the following drawbacks: (1) there may not exists a cross-monotonic cost-sharing method, and (2) computing a solution having that cost may be NP-hard. Therefore, one considers the effects of using approximation algorithms on the CO and the BB conditions.

Let $M = (A, P)$ be a mechanism whose cost function is $C_{\mathsf{A}}(\cdot)$. Mechanism M is *α-approximate BB* if it is cost recovery for $C_{\mathsf{A}}(\cdot)$ and *α-competitive*, that is, $\sum_{i \in U} P^i(b) \leq \alpha \cdot C_{\mathsf{opt}}(Q(b))$. A *$\beta$-surplus* mechanism M satisfies $\sum_{i \in U} P^i(b) \leq (1 + \beta) \cdot C_{\mathsf{A}}(Q(b))$. A *$\rho$-recovery* mechanism guarantees that $\sum_{i \in U} P^i(b) \geq \rho \cdot C_{\mathsf{A}}(Q(b))$, for some $\rho \leq 1$. Clearly, if A is an α-approximation algorithm and the mechanism is 0-surplus, then it is also α-approximate BB. The converse does not always hold as an α-approximate BB mechanism may not be 0-surplus. A *β-cost-sharing method* $\xi(\cdot)$ satisfies Eq. 2 and the following relaxation of Eq. 3: $C_{\mathsf{A}}(Q) \leq \sum_{i \in Q} \xi(Q, i) \leq \beta \cdot C_{\mathsf{A}}(Q)$.

2.1 Previous Work

A fundamental result by Moulin and Shenker states that, if a cost-sharing *cross-monotonic* method for $C_{\mathsf{A}}(\cdot)$ exists, then it is possible to define a group strategyproof mechanism (see Theorem 3): a cost-sharing method $\xi(\cdot)$ is cross-monotonic if, for every $Q' \subset Q \subseteq U$, $\xi(Q', i) \geq \xi(Q, i)$, for all $i \in Q'$. The converse of their result also holds whenever $C_{\mathsf{A}}(\cdot)$ is *submodular* [8, 7], that is, $C_{\mathsf{A}}(\emptyset) = 0$ and, for any two subsets of users Q_1 and Q_2, it holds that

$$C_{\mathsf{A}}(Q_1) + C_{\mathsf{A}}(Q_2) \geq C_{\mathsf{A}}(Q_1 \cup Q_2) + C_{\mathsf{A}}(Q_1 \cap Q_2).$$

The *Shapley value* for multicast routing [11] and the *egalitarian* method due to Dutta and Ray [2] are just two examples of cost-sharing methods which, for functions that are nondecreasing[2] and submodular, are cross monotonic.

The existence of a cross-monotonic method can be related to the *core* concept (see e.g. [3] for a definition): if the core of $C_{\mathsf{A}}(\cdot)$ is empty, then no cross-monotonic cost-sharing method $\xi(\cdot)$ for this cost function exists.

Megiddo proved that the optimal cost function $C_{\mathsf{opt}}(\cdot)$ for the Euclidean Steiner tree game has an empty core [6]. Kent and Skorin-Kapov provided the first cross-monotonic cost-sharing method for the minimum spanning tree game [4]. A more general approach has been given by Jain and Vazirani that use primal-dual methods in order to obtain a family of *polynomial-time computable* cross-monotonic methods [3]. These results yield, for the case of metric graphs, a 2-approximate BB, 0-surplus, group strategyproof mechanism for the Steiner tree and for the TSP games. The mechanism also meets NPT, VP and CS.

Biló *et al* [1] considered the muticast routing game in *wireless* networks. They proved that the resulting optimal cost function has an empty core even for d-dimensional Euclidean instances, for $d \geq 2$. Moreover, upon the results for the MST game, they build a $2(3^d - 1)$-approximate BB, group strategyproof mechanism which also meets NPT, VP and CS. This mechanism, however, is not 0-surplus.

2.2 Our Contribution

In this work, we first show how to get around the difficulties of dealing with cross-monotonic cost-sharing methods by providing a new technique for obtaining group strategyproof cost-sharing mechanisms. In particular, we prove that the a weaker property (which we call *self cross-monotonicity*– see Def. 2) suffices (Theorem 2). We prove the following results showing that our method is simpler and more powerful than the one by Moulin and Shenker [8, 7]:

- Self cross-monotonic methods for $C_{\mathsf{A}}(\cdot)$ can be trivially obtained whenever the algorithm A is *reasonable* (see Def. 3).
  The resulting mechanism M_{A} satisfies the NPT, VP, CS, cost recovery, is 0-surplus and, if A is an (polynomial-time) α-approximation algorithm, then M_{A} is (polynomial-time) α-approximate BB (Theorem 2).

[2] A function $C_{\mathsf{A}}(\cdot)$ is nondecreasing if, for every $Q \subset Q' \subseteq U$, $C_{\mathsf{A}}(Q) \leq C_{\mathsf{A}}(Q')$.

- Our method gives the first *polynomial-time* mechanism for the Steiner tree game which is group strategyproof, meets NPT, VP, CS and, more importantly, is BB (Corollary 1). Notice that the latter property implies that we are able to build a multicast tree which is *optimal* for the chosen receivers $Q(b)$, that is, $C_{\mathsf{A}}(Q(b)) = C_{\mathsf{opt}}(Q(b))$.

Besides the improvement over the 2-approximate BB mechanism in [3], the fact that our mechanism is BB is somewhat surprising: indeed, the result of Megiddo [6] implies that our result cannot be achieved using cross-monotonic methods; moreover, the NP-hardness of the underlying problem (i.e., given $Q \subset U$, find a minimum cost Steiner tree) seems to require α-approximate BB if we aim at polynomial-time mechanisms (see e.g. [3]). This intuition is wrong! Clearly, our result does *not* imply $\mathsf{P} = \mathsf{NP}$ since our mechanism is "driven" through a family of sets $Q_0, Q_1, \ldots, Q_n$ for which an optimal Steiner tree does *not* use any Steiner node (thus solvable in polynomial-time). We accomplish this by relating the sets Q_j's to the execution of Prim's MST algorithm (Theorem 4).

These results already prove that focusing (only) on cross-monotonic methods may be the "wrong" thing to do. We continue along this line and consider the wireless multicast game [1], another problem for which our method is provably better. We indeed obtain the following results on it:

- A polynomial-time mechanism which is $(3^d - 1)$-approximate BB, 0-surplus, group strategyproof, and meets NPT, VP, and CS (Theorem 5). This improves over the $2(3^d - 1)$-approximate BB mechanism in [1] which is not 0-surplus.
- A wide class of mechanisms for this game cannot be 0-surplus. This class includes the mechanism by Biló *et al* and, for certain "bad" instances, the surplus increases exponentially in d (for $d = 1$ and $d = 2$ it cannot be smaller than 1 and 5, respectively).

Mechanism in this class are those which use a multicast algorithm A for which an A-bad instance G exists (see Def. 4). These algorithms are not optimal (Theorem 8) and the cost function $C_{\mathsf{A}}(\cdot)$ is *not* submodular (Theorem 9). Hence, the "inverse" of the result by Moulin and Shenker [8] does not apply to such functions $C_{\mathsf{A}}(\cdot)$. Therefore, it is possible to have BB mechanisms which do *not* use cross-monotonic cost-sharing methods. Finally, we observe that there is no equivalence between bad algorithms A and the non submodularity of $C_{\mathsf{A}}(\cdot)$: indeed, there exists an instance G for which $C_{\mathsf{MST}}(\cdot)$ is not submodular, while G is not MST-bad (Theorem 10).

Paper Organization. We briefly recall the result by Moulin and Shenker in Sect. 3, and provide our extension in Sect.s 3.1-3.2; We apply our result to the Steiner tree game in Sect. 3.3; Wireless muticast is considered in Sect. 4.

Due to lack of space some proofs are omitted. The interested reader can find them in [10].

3 A New Method for Cost Sharing

Moulin and Shenker [8, 7] provide an elegant solution by considering the following scheme for obtaining mechanisms:

Mechanism $M(\xi)$

1. Q is initialized to U;
2. If there exists a user i in Q with $v_i < \xi(Q, i)$ then drop i from Q. Keep repeating this step, in arbitrary order, until for every user i in Q, $v_i \geq \xi(Q, i)$;
3. Set $P^i(b) := \xi(Q, i)$, for all $i \in U$.

A sharing method $\xi(\cdot)$ is *cross-monotonic* if, for every two sets Q and Q', with $Q' \subset Q \subseteq U$, it holds that $\xi(Q, i) \leq \xi(Q', i)$, for every $i \in Q'$.

The fundamental result by Moulin and Shenker reduces the problem of designing a mechanism to the problem of finding a *cross-monotonic* sharing method $\xi(\cdot)$ for a cost function $C_{\mathsf{A}}(\cdot)$. The resulting mechanism $M_{\mathsf{A}}(\xi)$ uses the scheme $M(\xi)$ to compute the set $Q = Q(b)$ and the payments $P^i(b) = \xi(Q(b), i)$, and then simply builds a feasible solution $T_{Q(b)} = \mathsf{A}(Q(b))$. Then the following holds:

Theorem 1. *[8, 7]* [3] *For any optimal (respectively, α-approximation) algorithm* A *and any cross-monotonic cost-sharing method $\xi(\cdot)$ for $C_{\mathsf{A}}(\cdot)$, the mechanism $M_{\mathsf{A}}(\xi)$ is group strategyproof, BB (respectively, α-approximate BB),* 0*-surplus and satisfies NPT, VP and CS.*

3.1 Extending Moulin and Shenker Approach

We will show that the cross-monotonicity property can be relaxed so to hold only for certain sets that mechanism $M(\xi)$ can actually output.

Definition 1. *Given any function $\xi : 2^U \times U \to \mathbf{R}^+ \cup \{0\}$, we define $\mathcal{Q}_0^\xi := U$, and $\mathcal{Q}_j^\xi := \{Q \setminus \{i\} |\ Q \in \mathcal{Q}_{j-1}^\xi \wedge \xi(Q, i) > 0\}$. Moreover, $\mathcal{Q}^\xi := \cup_{j\geq 0}\mathcal{Q}_j^\xi$.*

A key point is that mechanisms $M_{\mathsf{A}}(\xi)$ can generate only those subsets of receivers in $\mathcal{Q}^\xi$:

Lemma 1. *At each round of $M(\xi)$, the set Q considered in Step 2 satisfies $Q \in \mathcal{Q}^\xi$.*

Definition 2. *A function $\xi : 2^U \times U \to \mathbf{R}^+ \cup \{0\}$ is* self cross-monotonic *if, for every $Q, Q' \in \mathcal{Q}^\xi$ with $Q' \subset Q$, it holds that $\xi(Q', i) \geq \xi(Q, i)$, for every $i \in Q'$.*

We next prove the main result of this section. Its proof is similar to the one given in [3].

---

[3] The result presented here is sightly more general then the one in [8, 7]; indeed, as first observed in [3], their result can also deal with α-approximate BB mechanism.

Theorem 2. *For any optimal (respectively, α-approximation) algorithm* A *and any self cross-monotonic β-cost-sharing method $\xi(\cdot)$ for $C_{\mathsf{A}}(\cdot)$, the mechanism $M_{\mathsf{A}}(\xi)$ is group strategyproof, β-approximate BB (respectively, $\alpha\beta$-approximate BB), $(\beta-1)$-surplus and satisfies NPT, VP and CS. Moreover, $M_{\mathsf{A}}(\xi)$ runs in polynomial time if* A *and $\xi(\cdot)$ are polynomial time.*

Proof. Condition CS follows from the fact that a user i is dropped in Step 2 only if $b_i < \xi(Q, i)$. The NPT and VP conditions thus follow from the properties of $\xi(\cdot)$.

We next prove the group strategyproofness. Consider a coalition $C \subseteq U$ such that

$$j \notin C \Rightarrow b_j = v_j, \tag{4}$$

$$i \in C, b_i \neq v_i \Rightarrow v_i \cdot \sigma_i(Q^{false}) - P^i(b_C, v_{-C}) \geq v_i \cdot \sigma_i(Q^{true}) - P^i(v_C, v_{-C}), \tag{5}$$

where Q^{false} and Q^{true} denote the sets of receivers returned by $M_{\mathsf{A}}(\xi)$ on input (b_C, v_{-C}) and (v_C, v_{-C}), respectively. We have to show that the above inequality cannot hold with '>'. Observe that, if $i \notin Q^{false}$, then the NPT and the CS conditions imply that Eq. 5 holds with '='. We thus assume $i \in Q^{false}$ and we consider the following two cases:

$Q^{false} \subseteq Q^{true}$. From Lemma 1, $Q^{false} \in \mathcal{Q}^{\xi}$ and $Q^{true} \in \mathcal{Q}^{\xi}$. Since $i \in C$, by self cross-monotonicity and by the definition of $P^i(\cdot)$ in $M(\xi)$,

$$P^i(b_C, v_{-C}) = \xi(Q^{false}, i) \geq \xi(Q^{true}, i) = P^i(v_C, v_{-C}). \tag{6}$$

Since $Q^{false} \subseteq Q^{true}$, $\sigma_i(Q^{false}) \leq \sigma_i(Q^{true})$. By contradiction, if Eq. 5 holds with '>', then we would obtain

$$v_i \cdot \sigma_i(Q^{true}) - P^i(b_C, v_{-C}) > v_i \cdot \sigma_i(Q^{true}) - P^i(v_C, v_{-C}),$$

which contradicts Eq. 6.

$Q^{false} \not\subseteq Q^{true}$. We will show that this case cannot arise. Let $s_1, \ldots, s_k$ be the sequence of users that $M_{\mathsf{A}}(\xi)$ drops on input (v_C, v_{-C}), i.e., $Q^{true} = U \setminus \{s_1, \ldots, s_k\}$. Let s_j be the first user in $s_1, \ldots, s_k$ such that $s_j \in Q^{false}$. Therefore $b_{s_j} \geq \xi(Q^{false}, s_j)$. By definition of $s_1, \ldots, s_{j-1}$, $Q^{false} \subseteq Q_{j-1} := U \setminus \{s_1, \ldots, s_{j-1}\}$. By Lemma 1 and by the self cross-monotonicity of $\xi(\cdot)$, we have $\xi(Q^{false}, s_j) \geq \xi(Q_{j-1}, s_j)$. Since s_j is dropped in Q^{true}, the definition of $M_{\mathsf{A}}(\xi)$ implies that $\xi(Q_{j-1}, s_j) > v_{s_j}$. Putting things together we obtain

$$b_{s_j} \geq \xi(Q^{false}, s_j) \geq \xi(Q_{j-1}, s_j) > v_{s_j}. \tag{7}$$

If $s_j \notin C$, $b_{s_j} = v_{s_j}$, thus contradicting the above inequalities. Otherwise, when $s_j \in C$, Eq. 5 yields $v_{s_j} - P^{s_j}(b_C, v_{-C}) \geq 0$, thus implying $v_{s_j} \geq \xi(Q^{false}, s_j)$, which contradicts Eq. 7.

Finally, since, for every $Q \subseteq U$, $C_{\mathsf{A}}(Q) \leq \sum_{i \in Q} \xi(Q, i) \leq \beta C_{\mathsf{A}}(Q) \leq \alpha\beta \cdot C_{\mathsf{opt}}(Q)$, where α is the approximation ratio of A, $M_{\mathsf{A}}(\xi)$ is $\alpha\beta$-approximate and $(\beta-1)$-surplus.

3.2 Reasonable Algorithms Is All We Need

In the remaining of this work, given an instance G and a feasible solution T_Q for it, the corresponding set of users that are serviced is denoted to as $\mathsf{Serv}(T_Q, G)$.

Definition 3. *An algorithm* A *is* reasonable *if, for every instance G, there exists a sequence $i_1, i_2, \ldots, i_n$ of users such that, denoted by $Q_j := U \setminus \{i_1, i_2, \ldots, i_j\}$, for $1 \leq j \leq n$, it holds that $\mathsf{Serv}(\mathsf{A}(Q_j), G) = Q_j$, i.e., algorithm A is able to compute a solution which serves all and only the users in Q_j, for $0 \leq j \leq n$. (We let $Q_0 := U$.)*

Theorem 3. *If A is reasonable then there exists a self cross-monotonic cost-sharing method $\xi(\cdot)$ for $C_{\mathsf{A}}(\cdot)$.*

Proof. Let Q_j be the set defined as in Def. 3. To ensure self cross-monotonicity, we define

$$\xi(Q_j, i) = \begin{cases} C_{\mathsf{A}}(Q_j) & \text{if } i = j+1, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

We first show that $\mathcal{Q}_j^{\xi} = Q_j$. Indeed, at round j of $M_{\mathsf{A}}(\xi)$, the only user which can be dropped is $j+1$, for $0 \leq j \leq n$. Consider $Q, Q' \in \mathcal{Q}^{\xi}$ with $Q \subset Q'$. Then it must be the case $Q = Q_a$ and $Q' = Q_b$, for some $a > b$. Let $i \in Q$, with $\xi(Q, i) > 0$ (otherwise the theorem holds). Then $i = i_a$, thus implying $\xi(Q_b, i) = 0 = \xi(Q', i) < \xi(Q, i)$. Finally, $\xi(\cdot)$ can be easily extended outside $\mathcal{Q}^{\xi}$ so to enforce Eq.s 2-3 for every $Q \subseteq U$.

3.3 Steiner Tree Game

Consider a graph $G = (U \cup \{s\}, E, c)$ where the set of terminals coincides with the set of users U. Consider the execution of Prim's algorithm on graph G, starting from node $a_0 := s$. Let a_j be the j-th node added it the j-th iteration: a_j is the closest, among all nodes in $U \setminus \{a_1, \ldots, a_{j-1}\}$, to the connected component built so far, i.e., $S_{j-1} := \{s\} \cup \{a_1, \ldots, a_{j-1}\}$. Let T_j be the tree containing S_j. Then, for every $j \geq 0$, $\mathsf{COST}(T_j) = \mathsf{COST}(\mathsf{MST}(S_j))$.

We next strengthen this result and prove that $\mathsf{COST}(T_j)$ is also the optimal cost for the Steiner tree of S_j:

Theorem 4. *For every $j \geq 0$, let $ST^*(S_j)$ be an optimal Steiner tree in G with terminal set S_j and possibly using Steiner points in $U \setminus S_j$. Then, it holds that $\mathsf{COST}(ST^*(S_j)) = \mathsf{COST}(T_j)$.*

Proof. The proof is by induction on $r := n - j$, i.e., $S_j = S_{n-r}$ and $0 \leq r \leq n$.

Base step ($r = 0$). For $S_n = U$ there are no Steiner points, thus implying that $ST^*(S_n)$ must be a MST of G.

Inductive step (from $r = n - j - 1$ to $r + 1 = n - j$). Let $j + 1 = n - r$ and let (a_k, a_{j+1}) be the edge added at step $j+1$ to connect a_{j+1} to S_j. By contradiction, assume $\mathsf{COST}(T_j) \neq \mathsf{COST}(ST^*(S_j))$. Since $ST^*(S_j)$ is optimal for S_j, it must hold that $\mathsf{COST}(T_j) > \mathsf{COST}(ST^*(S_j))$. If a_{j+1} is not a node of $ST^*(S_j)$, then we let $T'(S_{j+1}) := ST^*(S_j) \cup (a_k, a_{j+1})$; otherwise, we let $T'(S_{j+1}) := ST^*(S_j)$. Since $a_k \in S_j$, then $T'(S_{j+1})$ is a tree spanning S_{j+1}. By definition, $T_{j+1} = T_j \cup (a_k, a_{j+1})$, thus implying

$$\begin{aligned}\mathsf{COST}(T'(S_{j+1})) &\leq \mathsf{COST}(ST^*(S_j)) + c_{(a_k,a_{j+1})}\\ &< \mathsf{COST}(T_j) + c_{(a_k,a_{j+1})} = \mathsf{COST}(T_{j+1}).\end{aligned}$$

By the inductive hypothesis $\mathsf{COST}(ST^*(S_{j+1})) = \mathsf{COST}(T_{j+1})$, and the above inequality contradicts the optimality of $ST^*(S_{j+1})$.

This completes the proof.

Theorem 4 implies that MST is reasonable and optimal for all sets $Q_j := S_{n-j}$, $0 \leq j \leq n$. Theorems 2 and 3 thus yield the following:

Corollary 1. *The Steiner tree game admits a mechanism $M_{\mathsf{MST}}(\xi)$ running in polynomial time which is group strategyproof, budget balance and satisfies NPT, VP and CS.*

4 Wireless Multicast and Limits of Cross-Monotonic Methods

Wireless multicast game. In *wireless* multicast routing, a feasible solution is a *directed tree* T containing a path from s to all of its nodes (i.e., T must be rooted at s and directed downwards). The cost of T is the total energy consumption required to implement all of its edges, which is equal to $\mathsf{COST}(T) := \sum_{i \in U} \max_{(i,j) \in T} c_{(i,j)}$. In the d-dimensional Euclidean version, $c_{(i,j)} = d(i,j)^\gamma$, for some $\gamma > 1$ and $d(i,j)$ being the Euclidean distance between i and j, and the instance G is a complete graph with nodes U. We assume $\gamma \geq d$ as in [1]. Fig. 1 shows a 2-dimensional Euclidean instance G:[4] the cost of the tree $T = \{(s,x_1),(s,x_2),(x_1,q_1),(x_2,q_2)\}$ is equal to $\epsilon + 2$. Interestingly, $T = \mathsf{MST}(G)$, which is not the optimal one:[5] the tree T^* connecting s directly to every other node has cost $(1+\epsilon)^\gamma$, which is better for sufficiently small ϵ. Observe that, $\mathsf{COST}(T) = C_{\mathsf{MST}}(U) < \sum_{(i,j) \in T} c_{(i,j)}$.

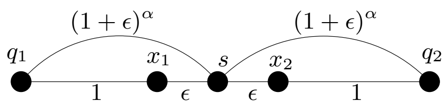

Fig. 1. The "bad" graph B_2

[4] For the sake of readability we do not draw all edges of the complete weighted graph G.
[5] For this problem, algorithm MST builds a MST of G and then orients it downwards s.

Theorem 5. *There exists a polynomial-time mechanism for the wireless multicast game which, for d-dimensional Euclidean networks, is group strategyproof,* $(3^d - 1)$*-approximate BB,* 0*-surplus and meets NPT, VP and CS.*

We next argue that graph B_2 in Fig. 1 constitutes an example of a "bad" graph for the MST algorithm in that, under certain hypothesis, it forces certain mechanisms $M_{\mathsf{MST}}(\xi)$ (the one by Biló *et al* [1] being one of them) to generate some surplus.

The two main ideas can be summarized as follows: (i) the two users $\{q_1, q_2\}$ must always pay at least the marginal cost $C_{\mathsf{MST}}(U) - C_{\mathsf{MST}}(U \setminus \{q_1, q_2\}) = (\epsilon + 2) - \epsilon = 2$; (ii) the MST algorithm, on input $U \setminus \{x_1, x_2\} = \{q_1, q_2\}$ yields a solution of cost $(1+\epsilon)^\gamma$ which is less than the above mentioned payment provided by $\{q_1, q_2\}$. Hence, some surplus is created if $Q(b) = \{q_1, q_2\}$.

Instead of proving the result for the graph B_2, we first generalized the above example to a wide class of graphs for which it is possible to prove that certain algorithms must create some surplus. Towards this end we first introduce some notation.

Notation. For any tree T, let $c(i, T) := \max_{(i,j) \in T} c_{(i,j)}$. Also let $\mathsf{pay}(T, i)$ be true if and only if $i = \arg\max\{l \mid (j, l) \in T \wedge c_{(j,l)} = c(j, T)\}$. Given an algorithm A, we let $\mathsf{A}(Q)$ denote the tree returned by A on input the set of receivers Q.[6] For every $Q \subseteq U$, we define the following two quantities:

$$C_{\mathsf{A}}(Q, i) := \begin{cases} c_{(j,i)} & \text{if } (j, i) \in \mathsf{A}(Q) \wedge \mathsf{pay}(\mathsf{A}(Q), i), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

$$\forall X \subseteq U,\ C_{\mathsf{A}}(Q, X) := \sum_{i \in X} C_{\mathsf{A}}(Q, i). \tag{10}$$

In particular, $C_{\mathsf{A}}(Q) = C_{\mathsf{A}}(Q, Q)$.

For every $i \in Q$, let Q_i^{A} be the subset of nodes that are reachable through i in $\mathsf{A}(Q)$ (i.e., those nodes that have i as an ancestor in $\mathsf{A}(Q)$). Let $\mathsf{A}_i(Q)$ be set of edges connecting i to Q_i^{A} in $\mathsf{A}(Q)$ (i.e., the edges in the subtree of $\mathsf{A}(Q)$ rooted at i). Notice that $\mathsf{A}_i(Q)$ does not contain i.

Definition 4. *A communication graph* $G = (U \cup \{s\}, E, w)$ *is* A-bad *if there exist* $Q \subseteq U$, $X \subset Q$ *and* $Y \subset Q$ *such that the following hold:*

$$\mathsf{A}(Q \setminus Q_X) = \mathsf{A}(Q) \setminus \bigcup_{i \in X} \mathsf{A}_i(Q) \tag{11}$$

$$C_{\mathsf{A}}(Q \setminus Y) < C_{\mathsf{A}}(Q, Q_X) \tag{12}$$

with $Y \cap Q_X = \emptyset$.

[6] We assume the algorithm A to return a tree connecting the source s to all and only the nodes in Q.

Theorem 6. *If G is* A*-bad, then there is no cross-monotonic cost-sharing method $\xi(\cdot)$ for $C_{\mathsf{A}}(\cdot)$.*

The mechanism by Biló *et al* [1] employs the cross-monotonic methods $\xi_F(\cdot)$ for the MST game by Jain and Vazirani [3]: given a family $F = \{f_1, \ldots, f_n\}$ of functions $f_i : \mathbf{R}^+ \to \mathbf{R}^+$, the function $\xi_F(\cdot)$ is a β-cost-sharing method for the wireless multicast cost function yielded by algorithm MST.

In the sequel, we will show that this kind of approach must always create some surplus. Intuitively, their mechanism $M_{\mathsf{MST}}(\xi_F)$ can potentially output every subset $Q \subseteq U$, which requires the method $\xi_F(\cdot)$ to be cross-monotonic. Theorem 6 thus implies that $\beta > 1$.

Definition 5. *A function $\xi : 2^U \times U \to \mathbf{R}^+ \cup \{0\}$ is Y-*critical *if, for all $j \in Y$, $\xi(U, j) > 0$, where $Y \subseteq U$.*

Theorem 7. *Let $G = (U \cup \{s\}, E, c)$ be a* A*-bad graph. If $\xi(\cdot)$ is a cross-monotonic β-cost-sharing method for $C_{\mathsf{A}}(\cdot)$ which is Y-critical, where Y is the set in Def. 4, then the mechanism $M_{\mathsf{A}}(\xi)$ is not 0-surplus (on the instance G).*

The above result can be applied to a family of graphs B_k generalizing graph B_2 in Fig. 1:

Definition 6. *For every integer $k \geq 2$, the graph $B_k = (U_k \cup s, E_k, c)$ is defined as follows: $U_k := \{q_l\}_{1 \leq l \leq k} \cup \{x_l\}_{1 \leq l \leq k}$, $E_k := \{(s, i) | i \in U_k\} \cup \{(q_l, x_l)\}_{1 \leq l \leq k}$. Moreover, $c_{(s,x_l)} = \epsilon$, $c_{(x_l,q_l)} = 1$ and $c_{(s,q_l)} = (1+\epsilon)^\gamma$.*

For B_k graphs, we can strengthen Theorem 7 and provide a lower bound on the surplus that all mechanisms using a U-critical function $\xi(\cdot)$ must generate. It is easy to verify that, for every F, $\xi_F(\cdot)$ is U-critical for *every* weighted graph G with non-zero edge weights. We thus obtain the following result on the mechanism $M_{\mathsf{MST}}(\xi_F)$ proposed by Biló *et al* [1]:

Corollary 2. *Let $\xi(\cdot)$ be cross-monotonic and U-critical. Then, for every graph B_k, mechanism $M_{\mathsf{MST}}(\xi)$ cannot be β-surplus, for any $\beta < k - 1$. Moreover, for d-dimensional Euclidean instances, $M_{\mathsf{MST}}(\xi_F)$ cannot be less than $(\tau_d - 1)$-surplus, with $\tau_1 = 2$, $\tau_2 = 6$, and τ_d increasing exponentially in d. These results apply to $M_{\mathsf{MST}}(\xi_F)$, for every F.*

The next result states that no A-bad graph exists if A is an optimal algorithm.

Theorem 8. *If $G = (U \cup \{s\}, E, c)$ is* A*-bad, then* A *is not optimal.*

Proof. Let $Q \subseteq U$ and $X, Y \subseteq Q$ be the sets as in Def. 4. Then Eq.s 11-12 imply respectively

$$C_{\mathsf{A}}(Q) = C_{\mathsf{A}}(Q \setminus Q_X) + C_{\mathsf{A}}(Q, Q_X) > C_{\mathsf{A}}(Q \setminus Q_X) + C_{\mathsf{A}}(Q \setminus Y). \tag{13}$$

Since $Y \cap Q_X = \emptyset$, $\{Q \setminus Q_X\} \cup \{Q \setminus Y\} = Q$. Hence, the tree $T := A(Q \setminus Q_X) \cup A(Q \setminus Y)$ reaches all nodes in Q and its cost satisfies

$$\mathsf{COST}(T) \leq C_{\mathsf{A}}(Q \setminus Q_X) + C_{\mathsf{A}}(Q \setminus Y) < C_{\mathsf{A}}(Q),$$

thus implying that A was not optimal on input Q.

One could try to prove that no BB mechanism employing algorithm A exists by showing that (i) there exists an A-bad instance and (ii) the function $C_{\mathsf{A}}(\cdot)$ is submodular. Unfortunately, this never happens:

Theorem 9. *If $G = (U \cup \{s\}, E, c)$ is A-bad, then C_{A} is not submodular.*

Notice that, the above theorem also implies that, if A-bad instances exist, it is still possible to have BB mechanisms which are not based on cross-monotonic cost-sharing functions for $C_{\mathsf{A}}(\cdot)$. In order to prove Theorem 9, we first need the following two intermediate results.

Lemma 2. *For every A-bad graph it holds that*

$$C_{\mathsf{A}}(Q \setminus Q_X) = C_{\mathsf{A}}(Q) - C_{\mathsf{A}}(Q, Q_X),$$

where X is the same as in Def. 4.

Proof. Eq. 11 implies that $\mathsf{A}(Q \setminus Q_X) = \mathsf{A}(Q) \setminus \{(i,j) | \ (i,j) \in \mathsf{A}(Q) \wedge j \in Q_X\}$. Hence, since $\mathsf{A}(Q)$ is a tree, we have

$$C_{\mathsf{A}}(Q \setminus Q_X) = \sum_{i \in Q} C_{\mathsf{A}}(Q, i) - \sum_{i \in Q_X} C_{\mathsf{A}}(Q, i) = C_{\mathsf{A}}(Q) - C_{\mathsf{A}}(Q, Q_X).$$

Lemma 3. *If $C_{\mathsf{A}}(\cdot)$ is submodular, then for any $Q', Q, A \subseteq U$, with $Q' \subset Q$ and $A \cap Q' = \emptyset$, it holds that*

$$C_{\mathsf{A}}(Q' \cup A) - C_{\mathsf{A}}(Q') \geq C_{\mathsf{A}}(Q \cup A) - C_{\mathsf{A}}(Q). \tag{14}$$

Proof. Since $C_{\mathsf{A}}(\cdot)$ is submodular, then for any $Q', Q, A \subseteq U$, with $Q' \subset Q$, and any $a \notin Q'$, it holds that

$$C_{\mathsf{A}}(Q) - C_{\mathsf{A}}(Q') \geq C_{\mathsf{A}}(Q \cup \{a\}) - C_{\mathsf{A}}(Q' \cup \{a\}). \tag{15}$$

By contradiction, assume that there exists $A = \{a_1, \ldots . a_k\}$, with $A \cap Q' = \emptyset$ such that $C_{\mathsf{A}}(Q) - C_{\mathsf{A}}(Q') < C_{\mathsf{A}}(Q \cup A) - C_{\mathsf{A}}(Q' \cup A)$. By repeatedly applying Eq. 15, with $a = a_1, a = a_2, \ldots, a = a_k$, we obtain

$$\begin{aligned} C_{\mathsf{A}}(Q \cup A) - C_{\mathsf{A}}(Q' \cup A) &> C_{\mathsf{A}}(Q) - C_{\mathsf{A}}(Q') \\ &\geq C_{\mathsf{A}}(Q \cup \{a_1\}) - C_{\mathsf{A}}(Q' \cup \{a_1\}) \\ &\geq C_{\mathsf{A}}(Q \cup \{a_1, a_2\}) - C_{\mathsf{A}}(Q' \cup \{a_1, a_2\}) \\ &\vdots \\ &\geq C_{\mathsf{A}}(Q \cup A) - C_{\mathsf{A}}(Q' \cup A), \end{aligned}$$

thus a contradiction.

We are now in a position to prove Theorem 9.

Proof of Theorem 9. From Def. 4 and Lemma 2 there exist $Q \subseteq U$ and $X, Y \subseteq Q$ such that

$$C_{\mathsf{A}}(Q, Q_X) = C_{\mathsf{A}}(Q) - C_{\mathsf{A}}(Q \setminus Q_X) > C_{\mathsf{A}}(Q \setminus Y). \tag{16}$$

By contradiction, assume that $C_{\mathsf{A}}(\cdot)$ is submodular. The fact that $C_{\mathsf{A}}(\cdot) \geq 0$, Lemma 3 (with $A = Q_X$) and Eq. 16 imply the following inequalities, respectively:

$$C_{\mathsf{A}}(Q \setminus Y) \geq C_{\mathsf{A}}(Q \setminus Y) - C_{\mathsf{A}}(Q \setminus \{Y \cup Q_X\} \geq C_{\mathsf{A}}(Q) - C_{\mathsf{A}}(Q \setminus Q_X) > C_{\mathsf{A}}(Q \setminus Y).$$

The above contradiction implies that $C_{\mathsf{A}}(\cdot)$ is not submodular.

The following result states that the converse of the above theorem does not hold. Hence, there is no equivalence between "non submodularity" and "badness" of cost functions.

Theorem 10. *There exists a two-dimensional Euclidean instance $G = (U \cup \{s\}, E, c)$ for which G is* not MST*-bad and $C_{\mathsf{MST}}(\cdot)$, restricted to G, is not submodular.*

Acknowledgements. We wish to thank the authors of [1] for providing us with a copy of their work.

References

1. V. Biló, C. Di Francescomarino, M. Flammini, and G. Melideo. Sharing the cost of muticast transmissions in wireless networks. In *Proceedings of SPAA*, pages 180–187, 2004.
2. B. Dutta and D. Ray. A concept of egalitarism under participation constraints. *Econometrica*, 57:615–635, 1989.
3. K. Jain and V.V. Vazirani. Applications of approximation algorithms to cooperative games. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 364–372, 2001.
4. K. Kent and D. Skorin-Kapov. Population monotonic cost allocation on MST's. In *Operational Research Proceedings KOI*, volume 43-48, 1996.
5. S. Leonardi and G. Schäfer. Cross-monotonic cost-sharing methods for connected facility location games. Technical Report MPI-I-2003-1-017, Max-Plankt-Institut für Informatik, September 2003.
6. N. Megiddo. Cost allocation for steiner trees. *Networks*, 8:1–6, 1978.
7. H. Moulin. Incremental cost sharing: characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16:279–320, 1999.
8. H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. 1997. http://www.aciri.org/shenker/cost.ps.
9. M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings of FOCS*, 2003.
10. P. Penna and C. Ventre. More powerful and simpler cost-sharing methods. TR of the European Project CRESCCO, http://www.ceid.upatras.gr/crescco/, 2004.
11. L.S. Shapley. A value of n-person games. Princeton University Press, 1953. H. Kuhn and W. Tucker editors.