Lecture Notes: Cost Sharing in Network Design Problems

Berthold Vöcking

Department of Computer Science
RWTH Aachen
Germany

07/14/2005

Suppose ...

... a town should be connected to cable TV. As usual, different residents have different valuations for being connected. These valuations are only known to the individual residents. We do not necessarily want to connect all agents to the service but maybe only a subset depending on the valuations of the agents and the cost for building the network that connects them to the service. The cost for connecting a subset of the agents is described by a known function $C : 2^N \to \mathbb{R}^+$, where N is the set of all residents.

For which subset of the agents should we build the service (to achieve which objective)? – How can we motivate the agents to tell the true valuation? – We need more details ...

General setup

- there is a set N of n agents suitable for a service
- each agent $i \in N$ can receive a service or not ($x_i = 1$ or $x_i = 0$)
- for agent i, receiving the service has value $v_i \geq 0$, receiving no service has value 0 (v_i is a secret of agent i)
- the cost for building the service for a subset $S \subseteq N$ of the agents is $C(S)$, where $C : 2^N \to \mathbb{R}^+$ with $C(\emptyset) = 0$ and $C(S) \leq C(T)$ for $S \subseteq T \subseteq N$.

Agents have to pay for the service, that is, if agent i receives the service (i.e., $x_i = 1$) then his payment is p_i, else 0. Given a valuation vector v, a *mechanism* computes an allocation (x, p). Agents aim at maximizing their *individual welfare (utility)*

$$w_i \;=\; x_i(v_i - p_i) \;\; .$$

Our goal is to design mechanisms in such a way that the dominant strategy for each agent is truthtelling.

Strategyproofness

A mechanism M is *strategyproof* if is satisfies the following condition for every $i \in N$. Let $v = v_1, \ldots, v_N$ denote the valuation vector, and let $v' = v'_1, \ldots, v'_N$ be any vector with $v'_j = v_j$ for all $j \neq i$. Let (x, p) and (x', p') denote the allocations computed by M for v and v', respectively. Strategyproofness requires

$$v_i x_i - p_i \geq v_i x'_i - p'_i \quad .$$

(In other contexts, *strategyproofness* is also called *truthfulness* or *incentive compatibility*.)

A stricter variant of strategyproofness requires that agents do not even have an incentive to lie if they form coalitions.

Group-strategyproofness

A mechanism M is *group-strategyproof* if it satisfies the following condition for every subset $T \in N$. Let $v = v_1, \ldots, v_N$ denote the valuation vector, and let $v' = v'_1, \ldots, v'_N$ be any vector with $v'_j = v_j$ for all $j \neq T$. Let (x, p) and (x', p') denote the allocations computed by M for v and v', respectively. Group-strategyproofness requires that if the inequality

$$v_i x_i - p_i \leq v_i x'_i - p'_i$$

holds for all $i \in T$ then it must be an equality for all $i \in T$ as well.

In words, if no member of a coalition T is made worse off by changing from v to v', then no member of T is made better off.

Conditions that a mechanism should satisfy

- **No Positive Transfers (NPT):** $p_i \geq 0$, for all $i \in N$
- **Voluntary Participation(VP):** $w_i \geq 0$, for all $i \in N$
- **Consumer Sovereignity (CS):** regardless which valuations the other agents report, for every agent i there is a value $\delta \in \mathbb{R}_{\geq 0}$ such that she would get the service when reporting that her valuation is δ

An important consequence of CS is that the naive mechanism that never serves any agent (i.e., always returns $x = 0$) is not permitted.

Objectives

When designing a mechanism, we could follow the following objectives:

- **Efficiency:** maximize social welfare $\sum_{i \in N} x_i v_i - C(x)$.[1]
- **Budget Balance:** cover the cost, i.e., $C(x) = \sum_{i \in N} p_i$.

It is known that a strategyproof mechanism cannot achieve both of these objectives simultaneously. In the following, we present the *marginal cost mechanism* that achieves efficiency and a family of *cross monotone cost sharing mechanisms* that yield budget balance.

[1] $C(x)$ abbreviates $C(\{i \in N | x_i = 1\})$.

The marginal cost mechanism

Given a vector of reported valuations v, let $W(v)$ denote the maximum welfare that can be achieved by serving any subset of the agents. A subset of the agents is called *efficient* if it yields social welfare $W(v)$. The marginal cost mechanism (MC) chooses an efficient set S and sets

$$x_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

$$p_i = \begin{cases} W(v|^i 0) - (W(v) - v_i) & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

where $v|^i\delta = (v_1, \ldots, v_{i-1}, \delta, v_{i+1}, \ldots, v_n)$, for any $\delta \in \mathbb{R}_{\geq 0}$.

MC satisfies NPT, VP, and CS

- NPT can be seen as follows. Let $i \in S$. We need to show that the payment $p_i = v_i x_i - (W(v) - W(v|^i 0))$ is non-negative. This follows from

$$W(v) = \sum_{i \in S} v_i - C(S) \leq v_i + \sum_{j \in S \setminus \{i\}} v_j - C(S \setminus \{i\}) \leq v_i + W(v|^i 0)$$

  because $W(v|^i 0)$ corresponds to the maximum welfare achievable over the set of agents $N \setminus \{i\}$.
- VP follows from $w_i = v_i - p_i = W(v) - W(v|^i 0) \geq 0$, for $i \in S$.
- CS follows as MC selects the agents in an efficient set S. Fix all declarations v_j, $j \neq i$. There is a threshold t such that every efficient solution contains i when $v_i > t$.

Strategyproofness of MC

Theorem 1: MC is strategyproof.

Proof: First, assume $i \in S$ when bidder i reports her true valuation v_i.

- In this case, reporting a higher value does neither affect x_i nor p_i, and hence the welfare of i does not change.
- When decreasing the reported valuation, x_i might be set to 0. In this case the welfare of the agent is 0. However, when telling the true valuation the value is at least 0 because of VP.

Strategyproofness of MC – cont.

Now assume $i \notin S$ when bidder i reports her true valuation v_i. In this case, decreasing the reported valuation can obviously not help. In order to achieve $x_i = 1$ bidder i must increase its reported valuation. Let $t \geq v_i$ denote the threshold such that reporting a valuation less than t implies $x_i = 0$ and reporting a valuation larger than t implies $x_i = 1$. Observe that $W(v|^i t) = W(v|^i 0)$ $(*)$. If bidder i increases her reported valuation to $\delta > t$, then her payment is

$$\begin{aligned} p_i' &= \delta - (W(v|^i \delta) - W(v|^i 0)) \\ &= \delta - (W(v|^i t) + (\delta - t) - W(v|^i 0)) \overset{(*)}{=} t \ . \end{aligned}$$

Hence, the welfare of agent i is $v_i - p_i' = v_i - t \leq 0$.

□

Remark about MC

MC is the only strategyproof mechanism that yields efficiency and satisfies the conditions NPT, VP, and CS. This has some interesting consequences.

- It is not difficult to construct an example in which MC is not budget balanced. Thus, efficiency and budget balance cannot be achieved simultaneously.
- Furthermore, one can also construct an example for which MC is not group-strategyproof. Hence, group-strategyproofness and efficiency can also not be achieved simultaneously.

Example: Multicast cost sharing

Some nodes in a computer network (e.g. the Internet) should be connected to a multicast tree (e.g. for a video transmission).

- The network is described by a directed graph $G = (V, E)$ with a source $s \in V$ and terminals $t_1, \ldots, t_n \in V$.
- Each terminal t_i is managed by a selfish agent $i \in N$ whose secret valuation for being connected with the source s is denoted v_i.
- An efficient subset of the terminals should be connected to the source.

There are different alternatives for defining the cost function depending on how the multicast trees between the source and the terminals are build.

Example: Multicast cost sharing – cont.

Steiner tree model:

- Each edge $e \in E$ comes with some cost $c(e)$.
- For $S \subseteq N$, let $T(S)$ denote the min-cost Steiner tree connecting the source s with the terminals of the agents in S.
- The cost for serving a subset of agents S is thus $C(S) = c(T(S)) = \sum_{e \in T(S)} c(e)$.

MC computes a set S maximizing social welfare $\sum_{i \in S} v_i - c(T(S))$. Then, for each $i \in S$, it computes a set S_i maximizing social welfare over the agents $N \setminus \{i\}$. The payment charged to agent $i \in S$ is

$$p_i = v_i - \left(\sum_{i \in S} v_i - c(T(S)) \right) + \left(\sum_{i \in S_i} v_i - c(T(S_i)) \right) \quad .$$

Example: Multicast cost sharing – cont.

The obvious computational bottleneck in the Steiner tree cost model is that MC has to solve several instances of the NP-hard Steiner tree problem. In practice, however, multicasts are often not build in form of general Steiner trees but the terminals can be only connected using paths that belong to the shortest path tree rooted at the source, which can be computed by a simple BFS. If the set of allowed multicast trees is restricted to subtrees of a given tree then the computational complexity for MC decreases significantly.

Example: Multicast cost sharing – cont.

Shortest Paths Tree Model:

- We are given a rooted tree $T = (V, E)$ with cost function $c : E \to \mathbb{R}_{\geq 0}$.
- The root of T is the source s and, w.l.o.g., the leaves correspond to the terminals $t_1, \ldots, t_n$. For simplicity in notation, we identify the set of leaves with the set of agents N.
- For a subset $S \subseteq N$, the cost for connecting S is the cost of the unique minimal subtree connecting s with the leaves in S.

The social welfare of a given set S can now be easily computed in linear time. But how can we efficiently identify the set S that maximizes the social welfare?

Example: Multicast cost sharing – cont.

The optimal social welfare W_v of an internal node $v \in V \setminus N$ can be computed recursively by the following formula

$$W_v = \sum_{v' \in \Gamma^+(v)} \left(W_{v'} - c((v, v')) \right) \ ,$$

where $\Gamma^+(v)$ denotes the children of v with positive contribution to the social welfare, that is,

$$\Gamma^+(v) = \{ v' \in V | v' \text{ is child of } v \text{ and } W_{v'} > c((v, v') \} \ .$$

The maximum social welfare that can be achieved is thus W_s. This value can computed by one bottom-up pass through the tree T. The set of welfare maximizing terminals S (and their payments) can then be computed by another top-down pass. (How?)

Cross monotone cost sharing mechanisms

Cost Sharing Method

A *cost sharing method* is a formula ξ associating to each subset $S \subseteq N$ of agents an allocation of total cost among these agents in the form of nonnegative cost shares $\xi_i(S)$, that is,

- $\xi_i(S) \geq 0$, for all $S \subseteq N$ and $i \in S$,
- $\xi_i(S) = 0$, for all $S \subseteq N$ and $i \notin S$, and
- $\xi_S(S) = \sum_{i \in S} \xi_i(S) = C(S)$.

The budget balanced mechanism that we build is based on a suitable cost sharing method ξ: The mechanism selects a set S of agents receiving the service. For $i \in S$, $\xi_i(S)$ is the cost share of i, i.e., the payment of i for the service.

Cross monotone cost sharing mechanisms – cont.

Cross Monotonicity

A cost sharing method ξ is called *cross monotone* if agent i's cost share cannot increase when the set of agents receiving service expands, that is,

$$S \subseteq T, i \in S \;\Rightarrow\; \xi_i(T) \leq \xi_i(S) \;.$$

Cross monotone cost sharing mechanisms – cont.

Given a cross monotone cost sharing method ξ and a valuation vector v, we can compute a Nash equilbrium allocation using an iterative tatonnement process as follows.

Tatonnement process

Set $S_0 = N$ and, for $t = 1, 2, \ldots$, compute

$$S_t = \{i | v_i \geq \xi_i(S_{t-1})\} \ .$$

The process terminates when reaching a set $S = S_t$ such that, for all $i \in S$, $v_i \geq \xi_i(S_t)$, that is, $S_t = S_{t+1}$.

Equilibrium property of the mechanism

Lemma 2: The set S found by the tatonnement process is a Nash eqilibrium.

Beweis:

- No agent $i \in S$ wants to leave S because $v_i \geq \xi_i(S_t)$.
- No agent $i \notin S$ wants to enter $S = S_t$ because it left a set $S_{t'}$ with $t' < t$ so that

$$v_i < \xi_i(S_{t'}) \leq \xi_i(S_t \cup \{i\}) \ ,$$

where the latter equation follows from $S_t \cup \{i\} \subseteq S_{t'}$ and the cross monotonicity of ξ. $\square$

Group strategyproofness of the mechanism

A closer look at the tatonnement process shows that S is even a *strong equilibrium*, i.e., there is no group of agents that can deviate in a manner which is profitable for at least one member and not disadvantageous for the others. As a direct consequence, we obtain the following result.

Theorem 3: The tatonnement process based on a cross monotone cost sharing method ξ is budget balanced and group-strategyproof.

It is easy to check that the conditions NPT, VP, and CS are satisfied as well.

Submodularity of the cost function

The question that we have not answered until now is how to derive a cross monotone cost sharing method ξ. To do this, we need that the given cost function satisfies the following property.

Submodular cost functions

A cost function $C : 2^N \to \mathbb{R}$ is *submodular* if for all $S, T \subseteq N$

$$C(S \cap T) + C(S \cup T) \leq C(S) + C(T) \ .$$

At this point, let us remark that the negative results about efficient mechanisms on page 13 also hold when assuming submodular cost functions.

Submodularity of the cost function – cont.

A more intuitive description of submodularity is that the marginal cost $C(S \cup \{i\}) - C(S)$ of adding agent i to a set S of other agents does not increase when the set expands. More formally:

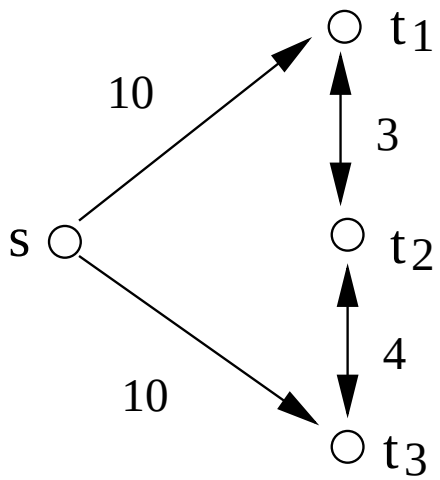
Lemma 4: A cost function C is submodular if and only if for all $S \subseteq T \subseteq N$ and $i \in N \setminus T$

$$C(S \cup \{i\}) - C(S) \geq C(T \cup \{i\}) - C(T) \ .$$

We leave the proof of this lemma as an exercise.

Submodularity of the Multicast-Problem

Submodularity seems to be a quite natural condition. Nevertheless, not all problems satisfy this conditions.



For example, the multicast problem in the Steiner tree model does not necessarily have a submodular cost function. The picture on the left shows a counterexample with $C(\{t_2\} \cup \{t_1\}) - C(\{t_2\}) = 0$ and $C(\{t_2, t_3\} \cup \{t_1\}) - C(\{t_2, t_3\}) = 3$.

In the shortest path tree model, however, the cost function is submodular. (Why?)

Building a cross monotone cost sharing method

Fix any permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$. For $S \subseteq N$ and $i \in N$, let

$$S_\pi(i) \;=\; \{j \in S | \pi^{-1}(j) \leq \pi^{-1}(i)\} \;\; .$$

In words, $S_\pi(i)$ contains i and all elements from S that π lists before i. We set

$$\xi_i(S) = C(S_\pi(i)) - C(S_\pi(i) \setminus \{i\}) \;\; .$$

In words, the cost share of agent i in S corresponds to her marginal cost assuming that the agents are added in the order described by π. Lemma 3 directly implies:

Observation 5: If the cost function C is submodular, then the constructed cost sharing method ξ is cross motonone.

The Shapley value mechanism

Instead of defining the cost share wrt a particular permutation, the Shapley value mechanism (SV) defines the cost shares by averaging over all permutations, that is,

$$\xi_i(S) \quad = \quad \frac{1}{n!}\sum_{\pi} C(S_\pi(i)) - C(S_\pi(i) \setminus \{i\})$$

SV has various remarkable properties like fairness and efficiency. For example, it is the most efficient budget balanced and group-strategyproof mechanisms, i.e., it achieves the largest efficiency among all budget balanced and group-strategyproof mechanisms.

SV for multicast cost sharing

The formula for the SV cost shares looks quite forbidding. For the multicast problem in the shortest path tree model, however, there is a much simpler and intuitive description of the SV cost shares: Consider a subset S of the terminals that should be served. The SV cost shares can be computed by distributing the cost of every edge e evenly among the terminals from S below e. (Why?)

Obviously these cost shares can be computed in polynomial time. Consequently, the tatonnement process runs in polynomial time as well. Thus, the SV mechanism has an efficient implementation for the problem of multicast cost sharing in the shortest path tree model.