Theoretical Computer Science 562 (2015) 419–435

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Bounding the payment of approximate truthful mechanisms ☆

Gruia Calinescu [1]

Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA

ARTICLE INFO

Article history:
Received 4 June 2011
Received in revised form 26 July 2012
Accepted 7 October 2014
Available online 13 October 2014
Communicated by G. Ausiello

Keywords:
Approximation algorithm
Mechanism design
Set cover
Polymatroid

ABSTRACT

In a STACS 2003 paper, Talwar analyzes the overpayment the *VCG* mechanism incurs for ensuring truthfulness in auctions. Among other results, he studies k-Set Cover (given a universe U and a collection of sets $S_1, S_2, \ldots, S_m$, each having a cost $c(S_i)$ and at most k elements of U, find a minimum cost subcollection – a *cover* – whose union equals U) and shows that the payment of the *VCG* mechanism is at most $k \cdot c(\mathcal{OPT}')$, where $\mathcal{OPT}'$ is the best cover disjoint from the optimum cover $\mathcal{OPT}$. The *VCG* mechanism requires finding an optimum cover. For $k \geq 3$, k-Set Cover is known to be NP-hard, and thus truthful mechanisms based on approximation algorithms are desirable. We show that the payment incurred by two mechanisms based on approximation algorithms (including the Greedy algorithm) is bounded by $(k-1)c(\mathcal{OPT}) + k \cdot c(\mathcal{OPT}')$. The same approximation algorithms have payment bounded by $k(c(OPT) + c(OPT'))$ when applied to more general set systems, which include k-Polymatroid Cover, a problem related to Steiner Tree computations. If q is such that an element in a k-Set Cover instance appears in at most q sets, we show that the total payment based on our algorithms is bounded by $q \cdot k^2$ times the total payment of the *VCG* mechanism.

1. Introduction

There has been a surge of recent interest in the intersection area of economic sciences and computer science (see [41] for a survey). Inside this area is the field of *mechanism design*, which we describe below in the particular setting we are going to analyze in this paper. The seminal paper of Nisan and Ronen [40] introduced the computational issues of mechanism design and gives a general overview in a setting more general than ours. A reader with economics background will recognize the scenario in our simplified case. A reader with computer science background will find below all the definitions used in this paper, and is warned that more definitions and notations are required in the general setting.

We are given a *ground* set $E \neq \emptyset$ of elements and a family $\mathcal{F} \neq \emptyset$ of *feasible* subsets of E, together comprising a *set system*. We consider only the case when the set system $(E, \mathcal{F})$ is closed upwards, which means that for any $J \in \mathcal{F}$ and any superset $J' : J \subseteq J' \subseteq E$, we have $J' \in \mathcal{F}$. This assumption is valid in certain circumstances, as in [46,3,20] and specific examples are given later in this paper. Each element $e \in E$ is controlled by a different selfish (economic) agent, which we call the agent of e. The agent of e has a private (unknown to anyone else) cost $c(e) \geq 0$ for providing e to a feasible subset. We consider only *single parameter* agents – their only parameter being the cost. Finding feasible subsets is sometimes called *team selection* in the literature [3,20] dealing with mechanisms.

☆ An extended abstract appeared in Proc. ISAAC 2004.

E-mail address: calinescu@iit.edu.

[1] Research supported in part by NSF Grant CCF-0515088.

http://dx.doi.org/10.1016/j.tcs.2014.10.019
0304-3975/

A *direct revelation mechanism* is a protocol which asks each agent to provide a bid $b(e)$ (giving an $|E|$-dimensional vector b), and then computes a feasible subset $N = N(b)$ and a payment function $p(e, b) \geq 0$. If $e \in N$, the agent of e must allow e to be used by N in exchange for the payment $p(e, b)$.

The agents are selfish and could misrepresent their cost in order to increase their payments. The field of *mechanism design* deals with the design of protocols which ensure the designer's goals are achieved by giving incentives to agents. The revelation principle [44,38] states that in order to prove or disprove the existence of a "good" (as defined in [44] – meaning it satisfies certain desirable properties which we do not mention) mechanism for a given problem, one can consider only truthful direct revelation mechanism. Thus we only use direct revelation mechanisms, which are fully characterized by the computation of the feasible subset N and the payment function p.

In our context, the economics notion of maximizing social welfare is minimizing the cost of N [40]. Minimizing the cost of N could be the goal of the designer of the mechanism. Another reasonable goal is minimizing the sum of payments. A mechanism is called *truthful* (also called strategy-proof) if every agent has as her best interest to bid her cost (by setting $b(e) = c(e)$) regardless of the other agents' bids.

Consider the simple case in which each one-element subset of E is feasible, a situation which occurs frequently in real life. Say the government needs a task to be performed and invites sealed bids from agents (contractors). If the government chooses the lowest bidder and pays to this agent her bid, then the agents may bid higher than their cost with the goal of making a profit. The government can however use the *VCG mechanism* [47,16,27]: select the lowest bidder and pay her an amount equal to the second lowest bid. Assuming agents don't collude, it can be shown that in this case each agent has as her best interest to make a bid equal to her cost. The *VCG* mechanism is truthful, and minimizes cost (maximizes social welfare). However the payment may be much larger than the cost.

The *VCG* mechanism can be applied to our upwards closed set system setting as well [46,3,6], and selfish agents would bid their cost: for all e, we have $b(e) = c(e)$. The mechanism selects *OPT*, a feasible subset of minimum cost, and computes payments with a formula we present later.

The upwards closed set systems we consider are instances of problems. For example, the elements can be the edges of a graph and feasible subsets consist of connected spanning subgraphs: the Minimum Spanning Tree problem. Of particular interest in routing protocols is the Shortest s–t Path problem [3,20].

Define $p(I, Mec)$ to be the total payment of truthful mechanism *Mec* on the instance I, that is the sum of the payments given by the mechanism to the agents. Note that, since the mechanism is truthful and the agents are assumed to be acting in their own interest, the bids $b(e)$ are equal to costs $c(e)$ and thus the payment is a function of the instance, for a given mechanism. Talwar [46] analyzes the *frugality ratio* of problems, which is defined as the maximum over instances I of $p(I, VCG)/c(OPT'(I))$, where $OPT'(I)$ is the best feasible subset disjoint from *OPT* for instance I. [46] characterizes the problems with frugality ratio 1. For example, Minimum Spanning Tree has frugality ratio 1 [6,23,46], and Shortest s–t Path has frugality ratio $\Theta(n)$ [3,46], where n is the number of vertices in the graph.

One interesting problem analyzed by Talwar [46] is the k-Set Cover problem: given a universe U and a collection of sets $S_1, S_2, \ldots, S_m$, each having a cost $c(S_i)$ and at most k elements of U, find a minimum cost subcollection (called *cover*) whose union equals U. k-Set Cover fits the general framework as follows: each set S_i is an element of E, and covers are the feasible subsets. [46] has proved that the frugality ratio of k-Set Cover is exactly k. Thus for k-Set Cover, the mechanism *VCG*, besides optimizing the cost (social welfare), guarantees a bound on the payment.

For $k \geq 3$, k-Set Cover is known to be NP-hard, and thus truthful mechanisms based on approximation algorithms are desirable. Lehmann et al. [34] prove and [3,1] mention that it is known that a direct revelation mechanism with single parameter agents is truthful if it is based on a *monotone* algorithm (we define later monotone algorithms, and how mechanisms are based on such algorithms); we call such a mechanism *monotone*. For k-Set Cover, we analyze two monotone mechanisms based on approximation algorithms: the mechanism *Greedy* based on the Greedy algorithm, and the *WOG* mechanism based on the Worst-Out-Greedy algorithm (defined later). For both mechanisms we show that their payment is bounded by $(k-1)c(\mathcal{OPT}) + k \cdot c(\mathcal{OPT}')$. This bound is at most twice worse than the bound of the *VCG* mechanism, which must find the optimum solution. For the mechanism *WOG*, we construct instances where our bound on the payment is tight. We present an instance of k-Set Cover where the payments of both mechanisms *WOG* and *Greedy* are lower than the payment of mechanism *VCG*.

We call (k, q)-Set Cover the Set Cover problem where each set has size at most k and every element appears in at most q sets. We prove that for all (k, q)-Set Cover instances I, we have $p(I, Greedy) \leq k^2 \cdot q \cdot p(I, VCG)$ and $p(I, WOG) \leq k^2 \cdot q \cdot p(I, VCG)$. The (k, q)-Set Cover problem is the only problem in our setting we are aware of where there is a nontrivial bound on the ratio of the payment of an approximation algorithm to the payment of the exact algorithm. Previous approximate truthful mechanisms [1,2] either are for maximization problems and look at revenue (instead of payment) or look at a different setting in which costs are public and the agents have some other private data.

In the k-Polymatroid Cover problem, the upwards closed set system is defined by a monotone submodular rank function $r : \mathcal{P}(E) \to \mathbb{N}$ with $r(e) \leq k$ for all $e \in E$: a feasible set J is a set satisfying $r(J) = r(E)$. The k-Polymatroid Cover problem is related to the full-component approach for Steiner Tree computation [42,50,5,45,43,9], and is a generalization of k-Set Cover. The k-Polymatroid Cover problem is NP-hard for $k \geq 3$, polynomial for $k = 1$ (when the problem becomes finding a minimum-cost basis in a matroid), while the complexity for $k = 2$ is unknown, but pseudopolynomial algorithms are known for linear polymatroids [10,42]. Wolsey [49] has shown that the greedy algorithm for k-Polymatroid Cover has approximation