Mechanism Design for Resource Allocation – with Applications to Centralized Multi-Commodity Routing*

(Extended Abstract)

Qipeng Liu
Institute of interdisciplinary information sciences
Tsinghua University, China
lqp1831@gmail.com

Yicheng Liu
Institute of interdisciplinary information sciences
Tsinghua University, China
61c@live.cn

Pingzhong Tang
Institute of interdisciplinary information sciences
Tsinghua University, China
kenshinping@gmail.com

ABSTRACT

We formulate and study the *algorithmic mechanism design* problem for a general class of resource allocation settings, where the center redistributes the private resources brought by individuals. Money transfer is forbidden. Distinct from the standard literature, which assumes the amount of resources brought by an individual to be public information, we consider this amount as an agent's private, possibly multi-dimensional type. Our goal is to design truthful mechanisms that achieve two objectives: maxmin and Pareto efficiency.

For each objective, we provide a reduction that converts *any optimal algorithm* into a *strategy-proof mechanism* that achieves the *same objective*. Our reductions do not inspect the input algorithms but only query these algorithms as oracles.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; J.4 [**Social and Behavioral Sciences**]: Economics; C.2.2 [**Computer-Communication Networks**]: Network protocols

Keywords

mechanism design, strategyproof, resource allocation, network routing

1. INTRODUCTION

One of the most important problems at the intersection of economics and computation is *algorithmic mechanism design*, which dates back to the seminal work of Nisan and Ronen [6]. The basic problem asks:

Given an algorithmic optimization problem, is it possible to efficiently produce a truthful mechanism that (approximately) achieves the optimal value of the original problem?

Over the past decade, there has been a number of breakthroughs regarding this problem in settings where money transfer is allowed, including the rich literature on truthful welfare-maximizing mechanism design [8, 4, 3], and Bayesian incentive compatible (BIC) mechanism design [5]. Recently, the same problem has been investigated under the context of revenue optimal mechanism design [1, 2].

Distinct from the above literature, we study algorithmic mechanism design . In particular, we study a general class of resource allocation problems, where each agent brings a certain amount of resources and the mechanism distributes these resources to achieve certain objectives. It is important to note that our setting differs from the standard resource allocation literature (such as one-sided matching, hedonic games, etc) in that each agent's type is the amount of resources she brings, rather than her preference over allocations. Our goal is to design strategy-proof mechanisms that achieve two objectives: maxmin and Pareto efficiency.

Our framework is rich enough to encompass, or at least heavily intersect with, a variety of applications, such as facility location [10], fair division [9] and network route allocation.

We make the following contributions. For any resource allocation problem, we provide two reductions, one for maxmin and the other for Pareto efficiency, that automatically convert any optimal algorithm into a strategy-proof mechanism that achieves the same objective. Our reductions do not make use of any algorithm but only assume black-box access to the algorithm.

- For the maxmin objective, we construct a polynomial time algorithm that, for any input, generates the optimal group-strategy-proof mechanism by calling the optimal algorithm only once.
- For Pareto efficiency, we show that, if there is an algorithm that *serially optimizes* the utility profile, the algorithm *per se* is strategy-proof.

2. OUR CONTRIBUTIONS

We formulate the resource allocation problem. An environment specifies the parameters for the mechanism designer to operate.

Definition 1. An environment is a tuple $\{\mathcal{N}, \mathcal{S}, \mathcal{P}, \mathcal{O}, u\}$, where

- $\mathcal{N}$ denotes the set of n agents,

*This work was supported by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the Natural Science Foundation of China Grant 61033001, 61361136003, 61303077, a Tsinghua Initiative Scientific Research Grant and a China Youth 1000-talent program.

Appears in: *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), Lomuscio, Scerri, Bazzan, Huhns (eds.), May, 5–9, 2014, Paris, France.*

- $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$, where each $\mathcal{S}_i$ is the private type set of agent i,
- $\mathcal{P}$ is a set of public information and resources shared by all agents,
- $\mathcal{O}$ is the set of outcomes.
- $u_i : \mathcal{O} \times \mathcal{P} \to \mathcal{R}$ is the utility function of agent i.

By revelation principle, one can without loss restrict attentions to the set of direct revelation mechanisms, which can be regarded as functions that maps agents' reported type profile and public information into an outcome.

Each agent i in the environment comes with a private amount of resources $s_i \in \mathcal{S}_i$ and is asked by the mechanism to report this quantity. Upon receiving all inputs, the mechanism returns an outcome, i.e., an allocation of resources. A resource allocation environment imposes certain feasibility constraints on any mechanism defined on it.

Definition 2. For any mechanism input $s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2, \cdots, s_n \in \mathcal{S}_n, p \in \mathcal{P}$, define $FEA(s_1, s_2, \cdots, s_n, p) \subseteq \mathcal{O}$ as a set of *feasible outcomes* under $(s_1, s_2, \cdots, s_n, p)$.

We consider environments and feasibility constraints that satisfy the following resource monotone property.

Definition 3. For an environment $\{\mathcal{N}, \mathcal{S}, \mathcal{P}, \mathcal{O}\}$, a feasibility constraint function FEA is *resource monotone* if $\forall i$, there is a partial order $\leq_i$ on $\mathcal{S}_i$, such that given any input profile $s_1, s_2, \cdots, s_n$ and $p \in \mathcal{P}$, for any $s'_i \leq s_i$, we have

$$FEA(s_1, \cdots, s'_i, \cdots, s_n, p) \subseteq FEA(s_1, \cdots, s_i, \cdots, s_n, p).$$

The goal of resource allocation mechanism design is to optimize a certain real-valued function $w : \mathcal{O} \to \mathbb{R}$ for equilibrium outcomes. Here w can be different functions according to different application scenarios. For example, in some cases w denotes social welfare, i.e., $w(o) = \sum_{i=1}^{n} u_i(o, p)$; while in some other cases, w denotes the minimum utility among all agents', i.e., $w(o) = \min_i u_i(o, p)$. The solution concept in this abstract is the dominant strategy equilibrium.

Consider the problem of designing a strategy-proof mechanism that maximizes the minimal utility among all agents, i.e., $w(o) = \min_{i \in \mathcal{N}} u_i(o, p)$. We show that, given an algorithm that computes the $\arg\max_o w(o)$, we can construct a strategy-proof mechanism with output o' such that $w(o') = w(o)$ for each input.

THEOREM 1. *If there exists an algorithm $\mathcal{A}$ that optimizes $w(o) = \min_i u_i(o, p)$ for some continuous, resource monotone environment, one can efficiently construct a strategy-proof mechanism $\mathcal{M}$ that optimizes $w(o)$.*

Given an algorithm $\mathcal{A}$, we can construct strategy-proof mechanism $\mathcal{M}$ as the mechanism 1. In other words, we find an outcome o_n that brings down all agents' utilities to u^*, the maxmin value computed by $\mathcal{A}$.

Applying the same technique, one can produce a strategy-proof mechanism to maximize the minimal utility subject to the individual rationality constraints. Also, we can convert any optimal algorithm into a strategy-proof mechanism that acheieves Pareto efficiency.

THEOREM 2. *If there exists an algorithm $\mathcal{A}$ that serially optimizes $w(o) = (u_1(o, p), u_2(o, p), \cdots, u_n(o, p))$ for monotone utility functions, resource monotone environment and $u_i(o, p) \geq r_i(p)$, and for each player i, her utility, which is a function of s_i, $u_i(\mathcal{A}(s_1 \ldots s_i \ldots s_n, p), p)$ is continuous, the algorithm itself is an IR and SP mechanism that computes a serially optimal outcome.*

Mechanism 1 A group strategy-proof mechanism via $\mathcal{A}$

1. on inputs $s_1, s_2, \cdots, s_n, p$, run $\mathcal{A}(s_1, \cdots, s_n, p)$ to get the outcome o_0;
2. let $u^* \leftarrow \min_i u_i(o_0, p)$;
3. for $i = 1, 2, \cdots, n$:

   find o_i such that for all $j \neq i$, $u_j(o_i, p) = u_j(o_{i-1}, p)$ and $u_i(o_i, p) = u^* \leq u_i(o_{i-1}, p)$;
4. outputs o_n;

3. CONCLUSION

In this abstract, we studied truthful mechanism design for a general class of resource allocation settings, where the center redistributes the private resources brought by individuals. We designed truthful mechanisms that achieve two objectives: maxmin and Pareto efficiency. For each objective, we provided a reduction that converts *any optimal algorithm* into a *strategy-proof mechanism* that achieves the *same objective*.

4. REFERENCES

[1] Y. Cai, C. Daskalakis, and S. M. Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *FOCS*, pages 130–139, 2012.

[2] Y. Cai, C. Daskalakis, and S. M. Weinberg. Understanding incentives: Mechanism design becomes algorithm design. In *FOCS*, pages 618–627, 2013.

[3] S. Dobzinski and J. Vondrák. The computational complexity of truthfulness in combinatorial auctions. In *ACM Conference on Electronic Commerce*, pages 405–422, 2012.

[4] S. Dughmi and T. Roughgarden. Black-box randomized reductions in algorithmic mechanism design. In *FOCS*, pages 775–784, 2010.

[5] J. D. Hartline and B. Lucier. Bayesian algorithmic mechanism design. In *STOC*, pages 301–310, 2010.

[6] N. Nisan and A. Ronen. Algorithmic mechanism design (extended abstract). In *Proceedings of STOC*, pages 129–140, New York, NY, USA, 1999. ACM.

[7] N. Nisan, T. Roughgarden, Éva Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[8] C. H. Papadimitriou, M. Schapira, and Y. Singer. On the hardness of being truthful. In *FOCS*, pages 250–259, 2008.

[9] A. D. Procaccia. Cake cutting: not just child's play. *Commun. ACM*, 56(7):78–87, 2013.

[10] A. D. Procaccia and M. Tennenholtz. Approximate mechanism design without money. In *ACM Conference on Electronic Commerce*, pages 177–186, 2009.