



When Analysis Fails: Heuristic Mechanism Design via Self-Correcting Procedures

Citation

Parkes, David C. 2009. When Analysis Fails: Heuristic Mechanism Design via Self-Correcting Procedures. For an invited talk given at the 35th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'09), January 24-30, Czech Republic.

Published Version

http://dx.doi.org/10.1007/978-3-540-95891-8_9

Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:3160486>

Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

When Analysis Fails: Heuristic Mechanism Design via Self-Correcting Procedures^{*}

David C. Parkes

School of Engineering and Applied Sciences
Harvard University
`parkes@eecs.harvard.edu`

Abstract. Computational mechanism design (CMD) seeks to understand how to design game forms that induce desirable outcomes in multi-agent systems despite private information, self-interest and limited computational resources. CMD finds application in many settings, in the public sector for wireless spectrum and airport landing rights, to Internet advertising, to expressive sourcing in the supply chain, to allocating computational resources. In meeting the demands for CMD in these rich domains, we often need to bridge from the theory of economic mechanism design to the practice of deployable, computational mechanisms. A compelling example of this need arises in dynamic combinatorial environments, where classic analytic approaches fail and heuristic, computational approaches are required. In this talk I outline the direction of *self-correcting* mechanisms, which dynamically modify decisions via “output ironing” to ensure truthfulness and provide a fully computational approach to mechanism design. For an application, I suggest heuristic mechanisms for dynamic auctions in which bids arrive over time and supply may also be uncertain.

1 Introduction

Mechanism design theory from microeconomics holds appeal within computer science as providing a principled method to develop optimized mechanisms for resource and task allocation problems of all kinds, ranging from problems in electronic commerce such as sponsored search to problems in networked systems such as allocation of wireless bandwidth and compute time on shared computational grids.¹

The essential problem addressed in mechanism design theory is that of *implementation*: given a set of $N = \{1, \dots, n\}$ agents each with private information $v_i : X \rightarrow \mathbb{R}$ on their value for outcomes X , with $v_i \in V$, design a game form

^{*} This abstract is for an invited talk given at the 35th International Conference on *Current trends in theory and practice of computer science* (SOFSEM’09), January 24-30, 2009, Czech Republic.

¹ See Jackson [1], Nisan [2] and Parkes [3, chapter2] for introductory surveys.

in which agents send messages that represent reports about their valuations, so that the outcome selected in the equilibrium of the game satisfies some desired social choice function $f : V^n \rightarrow X$. For example, perhaps the goal is to allocate a resource to the agent with the highest value, in which case a second-price sealed-bid [4] auction will implement the outcome in a dominant-strategy equilibrium. See Varian [5] for a simple exposition of this result and its generalization to the celebrated Vickrey-Clarke-Groves [4, 6, 7] mechanism.

It is often useful in mechanism design to focus on *direct-revelation* mechanisms, simultaneous-move games in which each agent sends a message that corresponds to a report about its valuation. A direct-revelation mechanism is defined by a pair (g, t) of an outcome rule $g : V^n \rightarrow X$ and payment rule $t : V^n \rightarrow \mathbb{R}^n$, where $t_i(v)$ for $v \in V^n$ is the payment by agent i to the mechanism. The standard analytical approach is to impose constraints on (g, t) and find the mechanism that best-satisfies a set of desiderata (e.g., efficiency, revenue optimality, fairness, budget balance, and so forth) across the space of incentive-compatible (= non-manipulable) mechanisms. This is best illustrated by Myerson’s derivation of the revenue-optimal mechanism for selling a resource when the seller has a distribution function F_i to characterize the valuation of each buyer $i \in N$. Myerson was able to show an equivalence between the problem of finding the optimal incentive-compatible mechanism and the comparatively simple problem of maximizing expected “virtual surplus” subject to monotonicity constraints,² and further reduce this to the problem of maximizing “ironed” virtual surplus. Ironing, for Myerson, is a process by which the virtual valuation functions are transformed into (weakly) non-decreasing functions of reported value by to “iron out” any such failure of monotonicity of virtual valuation as a function of valuation.

But in many problems of practical interest, the rules of a mechanism must be specified by a computational procedure rather than an analytic formula because no tractable procedure exists to implement the “ideal” mechanism, and these problems present a need to address two kinds of bottlenecks:

- The *scalability bottleneck*: mechanism rules must be realized by computationally tractable procedures.
- The *analytic bottleneck*: analytic methods to derive useful mechanisms fail in discrete, multidimensional and combinatorial problems.

Indeed, it is notable that very little progress has been made on extending Myerson’s analysis of revenue optimizing auctions to selling more than two goods.

Computational mechanism design seeks to combine economic and computational thinking, and has made good progress on addressing the scalability bottleneck. Early progress came by working with existing mechanisms, while making

² The *virtual valuation* of agent i given value v_i is $\phi_i(v_i) = v_i - [1 - F_i(v_i)]/F'_i(v_i)$, the *virtual surplus* of an allocation is the virtual valuation of the winner if the resource is allocated or the value of the seller for the resource otherwise, and monotonicity requires that the probability that an agent is allocated is (weakly) non-decreasing in its reported value, for all values of other agents.

significant advances through concise and expressive bidding languages, methods for efficient preference elicitation, and with fast algorithms for determining the outcome of a mechanism given reports. Much of this progress was made in the important domain of *combinatorial auctions*, i.e. auctions in which agents have non-linear valuations on bundles of items [8].

A second wave of progress involved reconciling tensions between computational and economic constraints, and developing novel mechanisms, for instance through “prior-free” mechanisms [9], approximate mechanisms [10, 11] and often working in problem domains that are particular to computer science such as those of network routing and job scheduling. But progress here was limited to addressing the scalability bottleneck and not the analytic bottleneck, in that where new mechanisms were derived the incentive-compatibility analysis was still performed by hand through mathematical analysis and this has limited progress.

A new, third wave of progress in computational mechanism design must seek to address both the scalability and the analytic bottlenecks by leveraging computational methods also for the *design* of mechanisms. This includes automated mechanism design [12] and parameterized mechanism design [13]. But I do not find these approaches entirely satisfactory, and argue in this talk the following point, that

a problem of computational mechanism design can be considered solved when we can take a state-of-the-art computational procedure for solving a centralized, cooperative problem and apply “with small modification” to solve the mechanism design problem.

Clearly a lot is buried in the meaning of “with small modification,” but I will offer an illustrative example in the form of *self-correcting mechanisms* for dynamic auctions [14, 15]. One of the underlying observations that drives work on self-correcting mechanisms is that rather than have a complete description of the rules of a mechanism, it is sufficient that mechanism properties be guaranteed only to the extent necessary to ensure equilibrium properties, so that self-interested participants know how to best represent their own interests. Given this observation, the basic idea in self-correcting mechanisms is to achieve these sufficient properties through a computational procedure that acts as a “wrapper” to augment an underlying (possibly heuristic) computational method.

A good example of a problem that exhibits both the scalability and analytic bottlenecks is that of *dynamic multi-item auctions* where bids arrive over time for multiple units of a resource, that may itself be in uncertain supply. Given a probabilistic model of bid dynamics, the underlying decision problem is one for which compelling computational methodologies exist but for which there are no closed-form solutions. Moreover, a complete characterization of truthful mechanisms is available that requires that the mechanism policy is *monotonic*, in the sense that if an agent is allocated for some bid then it is allocated for all “higher” bids (= larger value, smaller quantity, more relaxed time constraints) [16]. In the absence of a computational methodology for mechanism design, mechanisms are

available for this problem only in the case of simple probabilistic environments for which the underlying decision problem has an analytic solution [17–22].

In illustrating the idea of self-correcting mechanisms I will show how the computational procedure of *online stochastic combinatorial optimization* [23] can be combined with a procedure to automatically perform sensitivity analysis of the decision policy “on the fly,” in order to modify allocation decisions as necessary to ensure monotonicity properties and thus truthfulness. The result is a scalable, dynamic auction for buyers with multi-unit demand that is demonstrated for both expiring goods (e.g. time on a compute server) non-expiring goods with a deadline (e.g. theater tickets). The sensitivity analysis procedure is a form of “ironing” in the sense of Myerson, except that it irons away failure of monotonicity in the outcome of the mechanism rather than in the inputs to a mechanism, but to the same end of providing a truthful equilibrium. Thus I think of this as *output ironing*, whereas Myerson’s procedure is one of input ironing.

In summary, the main message of this talk is there exists a need to couple characterizations of truthful (or “incentive-compatible”) mechanisms with a fully computational methodology in which state-of-the-art computational procedures for decision making and optimization are modified in order to achieve these properties and without needing to resort to analytic methods to design mechanisms. I refer to these mechanisms as “self-correcting” mechanisms and offer applications to dynamic auctions to illustrate the approach.

Acknowledgments

Much of the content of this talk comes from collaborations with Ruggiero Cavallo, Satinder Singh and especially Florin Constantin and Quang Duong. This work is also supported in part by a Microsoft Fellowship and a Yahoo! Fellowship.

References

1. Jackson, M.O.: Mechanism theory. In Derigs, U., ed.: The Encyclopedia of Life Support Systems. EOLSS Publishers (2003)
2. Nisan, N.: Introduction to mechanism design (for computer scientists). In Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V., eds.: Algorithmic Game Theory. Cambridge University Press (2007)
3. Parkes, D.C.: Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency. PhD thesis, Department of Computer and Information Science, University of Pennsylvania (May 2001)
4. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* **16** (1961) 8–37
5. Varian, H., MacKie-Mason, J.K.: Generalized Vickrey auctions. Technical report, University of Michigan (1995)
6. Clarke, E.H.: Multipart pricing of public goods. *Public Choice* **11** (1971) 17–33
7. Groves, T.: Incentives in teams. *Econometrica* **41** (1973) 617–631
8. Cramton, P., Shoham, Y., Steinberg, R., eds.: *Combinatorial Auctions*. MIT Press (January 2006)

9. Goldberg, A., Hartline, J., Karlin, A., Saks, M., Wright, A.: Competitive auctions. *Games and Economic Behavior* **55** (2006) 242–269
10. Nisan, N., Ronen, A.: Algorithmic mechanism design. *Games and Economic Behavior* **35** (2001) 166–196
11. Lehmann, D., O’Callaghan, L.I., Shoham, Y.: Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM* **49**(5) (September 2002) 577–602
12. Conitzer, V., Sandholm, T.: Applications of automated mechanism design. In: *Proc. UAI Bayesian Modeling Applications Workshop*, Acapulco, Mexico (2003)
13. Likhodedov, A., Sandholm, T.: Approximating revenue-maximizing combinatorial auctions. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. (2005)
14. Parkes, D.C., Duong, Q.: An ironing-based approach to adaptive online mechanism design in single-valued domains. In: *Proc. 22nd National Conference on Artificial Intelligence (AAAI’07)*. (2007)
15. Constantin, F., Parkes, D.C.: Self-correcting sampling-based dynamic multi-unit auctions. Technical report, Harvard University (2008)
16. Hajiaghayi, M.T., Kleinberg, R., Mahdian, M., Parkes, D.C.: Online auctions with re-usable goods. In: *Proc. ACM Conf. on Electronic Commerce*. (2005) 165–174
17. Hajiaghayi, M.T., Kleinberg, R., Parkes, D.C.: Adaptive limited-supply online auctions. In: *Proc. ACM Conf. on Electronic Commerce*. (2004) 71–80
18. Pai, M., Vohra, R.: Optimal dynamic auctions. Technical report, Kellogg School of Management (2008)
19. Lavi, R., Nisan, N.: Competitive analysis of incentive compatible on-line auctions. In: *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*. (2000) 233–241
20. Gallien, J.: Dynamic mechanism design for online commerce. *Operations Research* (2006)
21. Gershkov, A., Moldovanu, B.: Dynamic revenue maximization with heterogeneous objects: A mechanism design approach. Technical report, University of Bonn (2008)
22. Pavan, A., Segal, I., Toikka, J.: Dynamic mechanism design: Revenue equivalence, profit maximization, and information disclosure. Technical report, Stanford University (2008)
23. Hentenryck, P.V., Bent, R.: Online Stochastic Combinatorial Optimization. MIT Press (2006)