An overview of online mechanism design

Course project for CS532a (Multi-agent systems)

Matt Hoffman
hoffmanm@cs.ubc.ca

December 27, 2006

Abstract

Online mechanism design is a generalization of traditional mechanism design which allows for dynamically changing sets of agents able to interact with the mechanism over some period of time. Aside from dealing with agents coming and going, the mechanism must make individual decisions as time progresses. This paper gives a brief overview of the online nature of this topic, and delves more deeply into the use of Markov decision processes for choosing outcomes that maximize the social welfare of all the agents.

1 Introduction

Classically, the study of mechanism design (MD) focuses on modeling situations in which all agents participate in a one-time decision made over some set of outcomes. This assumes that all agents are present when the mechanism (an auction for example) begins, and that all agents await the mechanism's decision. This characterization is slightly misleading in that these "one-time" decisions can involve an iterative process (e.g. indirect mechanisms) during which bids are made by different agents over successive iterations. For the purposes of this paper we will view iterative mechanisms as single time-step models, where this time-step just happens to be broken across several different iterations. This distinction still fits well with our notion of "classic" MD since all agents must be present when the iterative mechanism begins and must await its conclusion.

Online mechanisms generalize this model by introducing the notion of time dependency. Under the online model agents can arrive and depart at any discrete point in time, and the mechanism must make decisions at each time-step. In the next section we will formalize this model.

2 Formalizing the problem of online MD

For this paper we will consider finite time-horizon problems wherein decisions must be made at discrete time points $\mathcal{T} = \{1, \dots, T\}$. We will let $o = (o_1, \dots, o_T)$

denote a possible sequence of outcomes, or decisions, such that each decision o_t is drawn from the set O_t representing all feasible outcomes at time t. It might for example hold that $O_T \subseteq \cdots \subseteq O_1$, i.e. the number of feasible outcomes will decrease over time. Intuitively this sequence of sets might correspond to a repeated auction over a total of k items; here the sum over items sold at all time-steps cannot exceed k. It is worth noting, however, that this formulation allows for more general models.

For each agent $i \in \mathcal{I}$ interacting with the mechanism we will let $a_i, d_i \in \mathcal{T}$ denote the arrival and departure times respectively and let $v_i(o) \geq 0$ denote the valuation each agent has for the sequence of decisions o. It is assumed that agents have no valuation for decisions made outside of the interval $[a_i, d_i]$. This assumption is not overly restrictive since agents can have no affect on outcomes outside this interval that are not somehow reflected in their valuations inside the interval. We also assume that agents care only about maximizing their own utility, and have no valuation for other agents' utilities.

We can combine the previously introduced information by defining the type of each agent as $\theta_i = (a_i, d_i, v_i)$ which is drawn from some type-space Θ. We assume that each agent's type is drawn independently and identically distributed (iid) according to some probability distribution $f(\theta)$ and that this distribution is known to all agents.

Each agent can take actions by reporting some $\hat{\theta}_i \in \Theta$, and we will utilize a quasi-linear utility model over the course of this paper. In other words, the utility for each agent, parametrized by their type θ_i, depends on the sequence of outcomes o and some payment p; i.e. $u_i(o, p; \theta_i) = v_i(o) - p$. Together the set of agents $\mathcal{I}$, utility functions u_i, and types $\theta \in \Theta^n$ (for n agents) defines an "online" Bayesian game. The problem of online MD is then to make decisions o_t and require payments p_i from each agent in order to ensure some desired properties occur in an equilibrium of the induced Bayesian game (e.g. budget balance, efficiency, etc.).

Using this representation it is tempting to view this problem as just another form of classical MD in which each agent declares their type $\hat{\theta}_i$, makes payment p_i, and the mechanism decides on some set of outcomes $o = (o_1, \ldots, o_T)$. This is not the case for two major reasons that arise because the mechanism must function *online*. The simplest reason is that each decision o_t must be made before any outcomes can be chosen at time $t' > t$. The more difficult reason is that the mechanism must choose outcome o_t only having seen the declarations made by agents that have already arrived, i.e. all $i \in \mathcal{I}$ for which $a_i \leq t$. This distinction is slightly obscured by the notational use of $\theta \in \Theta^n$, which seems to imply that we know of all agents ahead of time. This is just used for notational convenience, however, and the mechanism must not rely on any such assumption (outside of distributional assumptions via $f(\theta)$).

Example 1. An example introduced by Friedman and Parkes [3] involves the pricing of WiFi at Starbucks in order to maximize the social welfare of all users. Agents announce their arrival time and how long they will be using the service, as well as their valuation for this time. The wireless bandwidth is limited and as a result the mechanism must decide which users to allow onto the network and what prices to charge these users.

Example 2. Consider an auction setting in which k goods are to be sold over some time period. Agents arrive at the auction one-by-one and place a bid $\hat{b}_i$ for some quantity of the goods[1]. The mechanism must then decide whether to accept the bid before moving on to the next bid (from another agent).

These two examples encompass a number of interesting settings, and in fact the auction setting can be seen as a generalization of the of which can actually be seen as a generalization of the second. In fact Example 2 is a restriction of Example 1 where only one agent is allowed to arrive at each time-step and each agent departs after the same time-step. In Section 4 we will introduce a truthful mechanism that is able to make allocations for the WiFi example that maximize the social welfare of all agents.

3 Lying strategically

As in classical mechanism design agents interacting with an online mechanism have the ability to lie in order to improve their expected utility under the induced game. In the online setting agents can also misreport their arrival or departure times strategically. For instance, in the WiFi allocation mechanism of Example 1 agents can delay reporting their arrival time in the hopes that other agents leave and the price drops. We further see that agents cannot declare an arrival time $\hat{a}_i < a_i$, since agents cannot make declarations before their arrival.

In the online auction setting (i.e. Example 2) Lavi and Nisan utilize a notion of *supply curves* in order to induce truthfulness. A supply curve is some function $\tilde{p}_i(k)$ that is fixed before receiving a bid $b_i(k)$ from agent i. Here $\tilde{p}_i(k)$ denotes the marginal payment for the kth item, and $b_i(k)$ the agent's marginal valuation for that item (as long as i is reporting truthfully). Since we have assumed that agents arrive one-by-one, i denotes both the agent and the time-increment. Given this fixed price, agents will purchase k_i items at price p_i such that

$$k_i = \arg\max_q \textstyle\sum_{j=1}^{q} (b_i(j) - \tilde{p}_i(j)), \qquad p_i = \textstyle\sum_{j=1}^{k} \tilde{p}_i(j).$$

If we require that the marginal price for each additional item does not decrease between iterations, we can see that delaying an agent's arrival time will be weakly worse for the agent since the price will never decrease. Agents can do no better lying about their valuation as well, since the price does not depend on this valuation. The sole remaining problem is to choose $\tilde{p}_i$, and this will generally be done in order to maximize the social efficiency or revenue of the seller. (Refer to the cited paper for more details.)

[1] Lavi and Nisan assume in [5] that this bid is some arbitrary function of k such that the marginal utility for each additional item is non-increasing. In this way each agent can bid on all available items.

In general problems of online MD, however, there are multiple agents arriving at each time-step where each agent will interact with the mechanism for a time period consisting of $d_i - a_i$. We cannot just assume that every agent arrives and departs on the same time step. Further, this allows the agent to lie about their departure time, which the previously discussed notion of supply-curves does not take into account.

While this greatly complicates our discussion of online mechanisms, we can instead restrict ourselves to direct-revelation incentive-compatible (truthful) online mechanisms. A direct-revelation online mechanism is one in which, as noted earlier, the only actions available to an agent are in announcing $\hat{\theta}_i \in \Theta$ where $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{v}_i)$. A truthful online mechanism is thus straightforward: one in which the agent immediately announces his/her type upon arrival. This relies upon an online variation of the Revelation Principle as noted in [3].

Theorem 1. *If a dominant-strategy or Bayes-Nash equilibrium of some social choice function F can be implemented by some online mechanism $\mathcal{M}$, then F can be truthfully implemented by some direct-revelation online mechanism $\mathcal{M}'$.*

The proof of this theorem follows the standard proof of the Revelation Principle in which the mechanism $\mathcal{M}'$ will *optimally lie* for each agent i when given the agent's true type. The next section will describe in more detail one particular truthful direct mechanisms in the online domain.

4 Online mechanisms as MDPs

In the last section we briefly described a particular mechanism for online auctions; in this section we will describe a mechanism applied to the more general domain of online MD. In particular, we will look at a formulation introduced by Parkes and Singh in [8] that views the computation performed by the mechanism as a Markov decision process (MDP).

An MDP comes equipped with a state space describing the state of the world at any point in time, a set of actions which can be taken from the current state, and a reward function which maps from state/action pairs into some real-valued reward. We also have a stochastic transition model which gives the probability of transitioning to some state x_{t+1} by taking some action a_t from state x_t. This formalism is usually used as a decision process for a single-agent acting in some noisy environment, and the solution to an MDP is a deterministic policy π which tells us which action to take from each state.

If we ignore for the moment incentive issues among agents we can formulate an MDP whose state space represents the history of the online mechanism and whose actions are the decisions or outcomes chosen by the mechanism (i.e. the center). We are then left to define the reward function. For the rest of this paper we will assume a reward function that is constructed such that the optimal policy will maximize the social welfare of all agents. This need not be the case however, and other interesting options might be a reward function that leads to a so-called "optimal" mechanism which maximizes the mechanism's revenue.

Following the notation of [8] we will denote the state of the MDP at time t as $h_t = (\theta_{\leq t}, o_{<t})$, the history of all reported types and all decisions made up to time t. Building upon the notation introduced in Section 2, we denote the outcomes possible in state h_t as O_t and all possible states at time t as H_t; this allows us to define the full state space $H = \bigcup_t H_t$. We can now introduce the state-transition model, a probability distribution $P(h_{t+1} \mid h_t, o_t)$. This distribution can be obtained using the type distribution $f(\theta)$, in particular by restricting this distribution to types θ_i such that $a_i = t{+}1$ and re-normalizing.

Before defining the reward function, we will let $R_i(h_t, o_t) = v_i(o_{\leq t}) - v_i(o_{<t})$ denote the marginal valuation of agent i attained when the mechanism makes decision o_t. We can then define the MDP reward function $R(h_t, o_t) = \sum_i R_i(h_t, o_t)$. Thus the rewards for this MDP directly align with the social welfare of all agents up to time t, bearing in mind that this is for the moment assuming that all agents are reporting their types truthfully.

The solution to this MDP is some policy $\pi = (\pi_1, \dots, \pi_T)$ such that each sub-policy $\pi_t : H_t \to O_t$ determines which outcome to choose given the agents currently interacting with the mechanism. This policy induces a value function $V^\pi(h_t)$ which represents the expected value of being in state h_t and choosing outcomes according to π, i.e.

$$V^\pi(h_t) = \mathbb{E}_\pi[R(h_t, \pi(h_t)) + \cdots + R(h_T, \pi(h_T))] \,.$$

The optimal value function V^* is one that maximizes the expected return for every state in H. We can first compute $V^*(h) = \max_{o \in O_T} R(h, o)$ for every state $h \in H_T$ occurring at the last time instant. Given the value function computed for all $h' \in H_{t+1}$ we can utilize the recursion

$$V^*(h) = \max_{o \in O_t} \Big\{ R(h, o) + \textstyle\sum_{h' \in H_{t+1}} P(h' \mid h, o) V^*(h') \Big\}$$

for all $h \in H_t$. This recursion is known as the *value iteration* algorithm, and has time-complexity polynomial in the size of the MDP and the maximum time-period T (see [13]). Finally, we can see that the optimal value function V^* is induced by the optimal policy π^*, where we can rewrite the above recurrence such that for all $h \in H_t$

$$\pi^*(h) = \arg\max_{o \in O_t} \Big\{ R(h, o) + \textstyle\sum_{h' \in H_{t+1}} P(h' \mid h, o) V^*(h') \Big\} \,. \tag{1}$$

It is well known in the reinforcement learning literature (e.g. [11, 13]) that a deterministic optimal policy always exists for any MDP.

One thing that we haven't mentioned, though, are the payments p_i that each agent is required to pay. The above policy or sequence of decision functions utilized by the mechanism have been calculated assuming that each agent is truthfully reporting their type $\hat{\theta}_i = \theta_i$

4.1 Delayed payments in online mechanisms

Friedman and Parkes [3] introduced a direct online variant of the Vickrey-Clarke-Groves (VCG) mechanism that brings truth telling into an equilibrium of the induced game. This delayed VCG mechanism was later extended to the previously discussed MDP framework by Parkes and Singh [8], and it is this notation that we will follow most closely. The mechanism $\mathcal{M}_D = (\Theta, \pi, p^D)$ chooses some sequence of outcomes $o = (o_1, \ldots, o_T)$ such that $o_t = \pi(h_t)$ for some history $h \in H$. The decision-policy is defined as in (1), where again agents are assumed to be reporting truthfully. Payments of

$$p_i^D(\hat{\theta}) = \left\{ \tilde{R}(\hat{\theta}_{-i}) - \tilde{R}(\hat{\theta}) \right\} - \tilde{R}_i(\hat{\theta})$$

are then collected from each agent i, but this collection is *delayed* until the final time period T. Here $\hat{\theta}_{-i}$ denotes the reported types of all agents except i and $\tilde{R}$ denotes the total reported social welfare over all involved agents over all time periods.

We can now show that truth-telling is a Bayes-Nash equilibrium of the induced game. Assume that all other agents report their valuations truthfully while agent i reports some type $\hat{\theta}_i$. Let $\theta_{>i}$ denote all agents who arrive strictly later than agent i. Here $\theta_{>i}$ is treated as a random variable when the decision is made, because at this point in the decision process those agents have not yet arrived and reported their types. We can then write the expected utility for agent i as

$$\mathbb{E}_{\theta_{>i}} \Big[v_i(\pi(\theta_{-i}, \hat{\theta}_i)) - \textstyle\sum_{j \neq i} \tilde{R}_j(\theta_{-i}, \hat{\theta}_i) + \tilde{R}(\theta_{-i}) \Big] .$$

After some algebraic manipulation we can see that this is similar to the standard VCG mechanism, and agent i is paying his *expected* social cost. This is not dominant strategy truthful because we must still assume that all agents are reporting truthfully (i.e. it's a Nash equilibrium of the Bayesian game), and we must also take into account the expectation under later agents. Finally, we must also assume that the policy π can simulate *delaying* reporting by the agent, in other words reporting some $\hat{a}_i > a_i$. This is not terribly restrictive though, in that the mechanism will make the best expected decision for the agent in terms of utility, whereby no decisions concerning the agent will be made if delaying would be more profitable.

There are two final problems with this mechanism. The first of which is that it is not readily apparent that the mechanism is individual-rational. It seems important that no agent is made worse off by taking part in the mechanism, at least under expectation. In order to show this we will make one other assumption about the induced MDP, namely that it exhibits value-monotonicity, i.e. for all history states h_t and some additional agent i it holds that

$$V^*(h_t(\hat{\theta} \cup \theta_i)) - V^*(h_t(\hat{\theta})) \geq 0.$$

In other words, it must hold that the addition of one more player to the induced MDP will make the optimal value function no worse. If this holds we can see that the expected utility to agent i can be calculated as

$$\mathbb{E}_{\theta_{>i}} \Big[\tilde{R}(\theta) - R(\theta_{-i}) \Big] \geq 0$$

and as a result the mechanism is *ex interim* individual rational.

The final problem with this mechanism is that it is not strictly an online algorithm. Decisions are made online, but agents are required to wait until the final time period T in order to make payments. In order to remedy this shortcoming Singh et al. [8] introduced an online VCG mechanism $\mathcal{M}_o = (\Theta, \pi, p^o)$ which calculates prices online. Each agent makes payments

$$p_i^o(\hat{\theta}) = \left\{ \tilde{V}^\pi(h_{\hat{a}_i}(\hat{\theta}_{-i})) - \tilde{V}^\pi(h_{\hat{a}_i}(\hat{\theta})) \right\} - \tilde{R}^i_{\leq m_i}(\hat{\theta}),$$

where $R^i_{\leq m_i}$ represents the social welfare of agent i for time periods up to m_i, the *commitment period* for agent i. The commitment period is some time-step $m_i \in [a_i, d_i]$ at which point the mechanism makes a decision affecting agent i. In the WiFi mechanism of Example 1 this might be the point at which the mechanism allocates bandwidth to the agent. This allows for the mechanism to simulate a delayed announcement for the agent, as noted earlier.

Although we use the notation here for the MDP value function, this is again a variant of the VCG mechanism wherein agents are paying their expected social cost. Here however the calculation can be done only with regards to those agents that can be affected by i's existence, i.e. those arriving before m_i. In order to save space the complete proof is omitted here, but this mechanism can be shown to be Bayes-Nash incentive compatible and ex-post individual rational. Interested readers are referred[2] to [8].

5 Conclusion and pointers for further reading

Online mechanism design presents an important generalization of classic mechanism design that allows for sensitivity to timing constraints and dynamically arriving agents. This is becoming especially important as many more electronic commerce applications spring up, including trade between non-human, or software agents [12, 2]. Another key application involves the negotiation of shared computing or network resources, especially those negotiated without human intervention [7, 4].

In this paper I've presented a very abbreviated overview of online mechanism design, and a slightly more in-depth look at the use of MDPs in making these online decisions. One topic that I was unable to touch on deals with the complexity of these models. In many situations the use of a history vector h as state in the induced MDP can be avoided using some sufficient statistics of this history. Even when this can be done, as the time T and MDP size increase the value-iteration algorithm becomes increasing more complex. For very large T, or infinite T with a discount factor[3], it may be more efficient to utilize the policy-iteration algorithm. A tight worst-case bound on the complexity of policy-iteration is still an open problem, however—see [13] for more details.

[2] It should be noted that the referenced work utilizes a negative payment, or a payment *made to each agent*. As a result the payments here differ by a factor of -1.

[3] The discount factor must be the same for all agents in order for this to work.

Another interesting approach taken by Singh et al. [9] attempts to reduce the complexity of the MDP-based approach by utilizing an ε-Bayes-Nash equilibrium. This uses the assumption that agents are indifferent between valuation changes less than ε.

Other approaches have extended upon the basic model presented in this overview. Work by Porter [10] applies online mechanism to the problem of real-time scheduling of computational resources. Specifically this paper looks at problems with continuous-valued time-spaces. Bredin and Parks [1] also apply this analysis to online double-auctions which have more applications to buyer/seller markets. Finally, a recent paper by Lavi and Nisan [6] looks at online auctions for expiring goods, which presents another interesting application of the time dynamics touched on here.

References

[1] J. Bredin and D. Parkes. Models for truthful online double auctions. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI'2005)*, pages 50–59, 2005.

[2] E. Ephrati and J. Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *In Proc.. of the National Conference on Artificial Intelligence*, 1991.

[3] E. Friedman and D. Parkes. Pricing WiFi at Starbucks—issues in online mechanism design. In *Proc. Fourth ACM Conf. on Electronic Commerce*, 2003.

[4] Y. A. Korilis and A. A. Lazar. On the existence of equilibria in noncooperative optimal flow control. *Journal of the ACM*, 42(3):584–613, 1995.

[5] R. Lavi and N. Nisan. Competitive analysis of online auctions. In *Proc. ACM Conference on Electronic Commerce*, 2000.

[6] R. Lavi and N. Nisan. Online ascending auctions for gradually expiring items. Technical report, The Hebrew University, 2004.

[7] A. Lazar and N. Semret. The progressive second price auction mechanism for network resource sharing, July 1998. In 8th Int. Symp. on Dynamic Games and Applications, Maastricht.

[8] D. Parkes and S. Singh. An MDP-based approach to online mechanism design. In *Proc. of Neural Information Processing Systems*, volume 17, 2003.

[9] D. Parkes, S. Singh, and D. Yanovsky. Approximately efficient online mechanism design. In *Proc. 18th Annual Conf. on Neural Information Processing Systems*, 2004.

[10] R. Porter. Mechanism design for online real-time scheduling. In *In Proc. of the ACM Conference on Electronic Commerce*, 2004.

[11] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, New York, 1994.

[12] J. Rosenschein and G. Zlotkin. *Rules of encounter: designing conventions for automated negotiation among computers.* MIT Press, 1994.

[13] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.