Algorithmic and Game-theoretic Approaches to Group Scheduling

Hooyeon Lee
Stanford University
haden.lee@stanford.edu

1. INTRODUCTION

Scheduling an event for a group of people is a notoriously frustrating task; it tends to be tedious, time consuming, and often leads to suboptimal schedules. One reason the problem has resisted a solution is that it has many dimensions, some of them easy to put one's finger on, but others more subtle such as game-theoretic manipulability of the procedure.

The first part of this work seeks algorithmic solutions to group scheduling problem. Assuming prompt and truth-telling agents, we can aim to design an efficient algorithm that finds the optimal scheduling process with respect to a given cost function. In Section 3 we formally define this as an optimization problem, and investigate the difficulty of the problem. We propose an efficient algorithm for finding the optimal solution, and then use the proposed algorithm in simulations to show that the result substantially outperforms the baseline approach in many realistic settings.

The second part of this work puts more weight on the strategic behavior of agents. Imagine that agents have preferences over both the time and the number of attendees at the event. Given a set of invitees, the set is said to be *stable* if all invitees prefer attending to not attending, and if no uninvited person wishes she had been invited. Stability is obviously desirable, but in general a stable schedule may not exist; therefore it is interesting to know how hard it is to determine whether it exists (and to find the one that maximizes attendance). These questions take an extra meaning if agents can strategically misreport their preferences, and we answer these questions in Section 4.

2. RELATED WORK

Early work in AI had a heuristic, systems-oriented flavor to it. Jennings et al. [4] proposed the design of an agent-based meeting scheduling system that negotiates with others on behalf of human users. More recently, Crawford and Veloso [1] approached to the group scheduling problem by training agents to learn about the negotiation strategies of others. The work by Darmann et al. [2] is closely related to ours. There agents are assumed to have preferences over activities as well as number-of-participants. We inherit from the work of Darmann et al. both the preference structure and the stability criterion. In their work, though, agents can be assigned to one of any number of activities, but in our problem a single time slot must be selected for the event. More dramatically, Darmann et al. only consider non-strategic agents, while we consider both the non-strategic and the strategic cases. Ephrati et al. [3] tackled incentive issues, by extending the VCG mechanisms into scheduling systems, assuming that availability of the agents is known.

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

3. OPTIMAL GROUP SCHEDULING

3.1 Problem Setting

We consider a class of Single-proposer Mechanisms (SPMs) in which the sole convener tries to find an agreeable time slot for a group of agents. Consider a set N of n invitees and a set S of s time slots. For each agent a_i and each time slot t, there is a known prior probability, $p_{i,t}$, such that the agent is available at time t with probability $p_{i,t}$ and unavailable with probability $(1-p_{i,t})$. While $p_{i,t}$'s are known to the convener as a prior, the realization of availability of each agent is private information. The convener aims to find a *feasible* time slot by asking invitees to reveal their availability; a feasible time slot requires at least $\lceil f \cdot n \rceil$ agents be available (f is called the feasibility threshold). The convener iteratively sends out a group of time slots until a feasible one is found. In our setting Time spent by the scheduling process is measured by the number of iterations and Inconvenience caused by the total number of time slots that have been sent out by the convener. Any valid (ordered) partition of S is called a B-Doodle mechanism, denoted by B, which describes the way the convener is polling availability of agents.

Definition 1 (Group Scheduling Problem)**.** An instance of the Group Scheduling Problem (GSP) is a tuple (N, S, f, P, c) where $N = \{a_1, a_2, \ldots, a_n\}$ is a set of n agents , $S = \{1, 2, \ldots, s\}$ is a set of s time slots, f is the feasibility threshold ($0 \leq f \leq 1$), $P = \{p_{i,t}\}$ is probability distributions of availability (with $0 \leq p_{i,t} \leq 1$ for all i, t), and c is a cost function (where $c(j, b) > 0$ describes the aggregate cost of Time and Inconvenience that is incurred by sending out b time slots during the j-th iteration). We assume that cost is additive so that the overall cost of executing the first j iterations of $B_m = \langle S_1, S_2, \ldots, S_m \rangle$ is simply the sum, $\sum_{k=1}^{j} c(k, b_k)$ where $b_k = |S_k|$. The objective in GSP is to find an optimal partition, B^*, of S such that B^* minimizes the expected cost (expectation with respect to P), given (N, S, f, c, P).

3.2 Preliminary Results

In this idealized setting where agents are prompt and truth-telling, we are able to design an efficient algorithm that runs in polynomial time and finds the optimal B-Doodle mechanism that minimizes the expected cost, given an instance of (N, S, f, P, c) of GSP. Using our algorithm in simulations, we also observed that Doodle is substantially inefficient, including (but not limited to) when one or more of the following conditions hold:

- There is a relatively small number of agents ($n \leq 10$).
- There is a large number of time slots ($s \geq 15$).
- Agents are relatively free ($p > .5$).
- Feasibility threshold is relaxed ($f < .8$).
- c places more weight on Inconvenience than Time.

First four conditions affect the probability of feasibility of a given time slot in the same way; if the probability is high, then Doodle ought to be more inefficient. While the last condition is independent of this probability, c determines what is being optimized, and Doodle becomes more inefficient if c favors reducing Inconvenience over reducing Time.

4. COPING WITH STRATEGIC AGENTS

4.1 Problem Setting

Definition 2 (Setting)**.** An instance of the *Stable Group Scheduling Problem* (SGSP) is a tuple (N, M, P) where $N = \{a_1, a_2, \ldots, a_n\}$ is a set of n agents, $M = \{t_1, t_2, \ldots, t_m\}$ is a set of time slots, and P is a collection of preferences of agents ($P = (P_1, P_2, \ldots, P_n)$). For each agent a_i, P_i is a total preorder ($\succeq_i$) on the set of alternatives, $X = X_0 \cup \{x_\emptyset\}$, where $X_0 = (M \times \{1, 2, \ldots, n\})$ and $x_\emptyset$ is the outside option of not attending; for any alternative $x \in X$, $(t, k) \succeq_i x$ is interpreted as agent a_i weakly preferring attending the event at time t if k attendees are present (including herself) to the alternative x (and similarly for $x \succeq_i (t, k)$).

We set $A_i = \{(t, k) \in X_0 | (t, k) \succ_i x_\emptyset\}$ and say that agent a_i approves of all alternatives in A_i. When each agent is indifferent among all alternatives in A_i, we call the problem a *simple Stable Group Scheduling Problem* (s-SGSP).

Definition 3 (Schedule)**.** A *schedule* for an instance (N, M, P) is a pair (t, S_t) such that $t \in M$ and $S_t \subseteq N$, and is interpreted as the organizer chooses time t and invites a subset of agents, S_t. Note that $S_t = \emptyset$ is allowed in our definition. A schedule (t, S_t) is said to be *individually rational* if for every agent $a_i \in S_t$ it holds that $(t, |S_t|) \in A_i$. A schedule (t, S_t) is said to be *envy-free* if for every agent $a_i \notin S_t$ it holds that $(t, |S_t \cup \{a_i\}|) \notin A_i$. A schedule is *stable* if it is both individually rational and envy-free.

Definition 4 (Single-peaked preferences)**.** Given an instance (N, M, P) of SGSP, the preferences of agent a_i are *single-peaked* (SPK) if for every fixed time slot $t \in M$ there exists an ideal number of attendees, $o_i^t \in \{1, \ldots, n\}$, such that for any $k_1 \leq k_2 \leq o_i^t$ it holds that $(t, k_2) \succeq_i (t, k_1)$ and for any $o_i^t \leq k_2 \leq k_1$ it holds that $(t, k_2) \succeq_i (t, k_1)$.

There are two important special cases of SPK-preferences: increasing preferences (INC-preferences) and decreasing preferences (DEC-preferences). Agent a_i is said to have an INC-preference with respect to time slot t if $o_i^t = n$ (this agent prefers maximizing attendees). Analogously agent a_i is said to have a DEC-preference with respect to t if $o_i^t = 1$. We assume that all agents have SPK-preferences with respect to all time slots; such an instance is called an SPK-instance of SGSP (and analogously for INC and DEC-instances).

4.2 Preliminary Results

Theorem 1. Given an SPK-instance of SGSP and non-strategic agents, there exists an algorithm that terminates in polynomial time, and decides whether a stable schedule exists (and if it exists, the algorithm finds a maximum one).

Theorem 2. It is impossible to design a strategy-proof mechanism that finds any stable schedule (provided that it exists), even if the problem space is restricted to DEC-instances of s-SGSP with $|N| = 2$ and $|M| = 1$. It is also impossible to design a strategy-proof mechanism that finds any stable schedule (provided that it exists), even if the problem space is restricted to INC-instances of SGSP with $|N| = 2$ and $|M| = 2$.

Theorem 3. There exists a (deterministic) strategy-proof mechanism that finds a maximum stable schedule in polynomial time, given an INC-instance of s-SGSP.

5. CURRENT AND FUTURE WORK

In Section 3, we investigated an optimization problem in group scheduling. While useful in and of itself, this work leaves many open questions. It will be interesting to extend our B-Doodle mechanisms to Multi-proposer Mechanisms (MPMs) in which agents may negotiate with others. We implicitly assumed that agents prefer all time slots equally, but it will be important to study the trade-off between optimizing the cost and finding a 'good' schedule.

In Section 4, we moved our focus to stability constraints and strategic behavior of agents. One direction for future work is to allow each agent to specify her preferences over time slots (in addition to the number of attendees). Another direction is to take the identities of attendees into account. In realistic situations an agent may have constraints such as "I will only attend if some agent y attends as well". We believe such extensions will make our model more realistic and applicable in multi-agent scheduling systems.

6. ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation (under grants IIS-0963478 and IIS-1347214), the U.S. Army (under grant W911NF1010250), and the Kwanjeong Educational Foundation.

7. REFERENCES

[1] E. Crawford and M. Veloso. Learning to select negotiation strategies in multi-agent meeting scheduling. *Progress in Artificial Intelligence*, pages 584–595, 2005.

[2] A. Darmann, E. Elkind, S. Kurz, J. Lang, J. Schauer, and G. Woeginger. Group activity selection problem. In *Internet and Network Economics*, pages 156–169. Springer, 2012.

[3] E. Ephrati, G. Zlotkin, and J. Rosenschein. A non-manipulable meeting scheduling system. In *Proceedings of the 13th international workshop on distributed artificial intelligence*, pages 105–125, 1994.

[4] N. R. Jennings, A. J. Jackson, and L. E. Ns. Agent-based meeting scheduling: A design and implementation, 1995.