Optimizing Time and Convenience in Group Scheduling

(Extended Abstract)

Hooyeon Lee
Stanford University
haden.lee@stanford.edu

Yoav Shoham
Stanford University
shoham@cs.stanford.edu

ABSTRACT

We consider the situation in which a group of agents are trying to find an agreeable time for a meeting. Each agent's availability is governed by a known prior probability distribution, but the realization of it is private information. The goal (of the convener) is to find an agreeable time slot while optimizing in two dimensions: Expected time to reach agreement and expected inconvenience caused during the process.

We take the most common web-based tool in use today – Doodle – as the starting point. We generalize Doodle to a class of B-Doodle mechanisms ("Batched Doodle") and study the trade-off between the Time-Inconvenience dimensions. In a B-Doodle mechanism, the convener iteratively asks agents about their availability for one or more time slots, until an agreement can be reached. Among (exponentially) many B-Doodle mechanisms, the main objective is to find a B-Doodle mechanism that minimizes the aggregate cost of Time and Inconvenience (such as a linear combination of the two). We provide an efficient algorithm that achieves this for a broad class of cost functions, and show in simulations that the result substantially outperforms Doodle in many realistic settings.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

Keywords

Doodle; Group Scheduling

1. INTRODUCTION

Scheduling an event for a group of people is a notoriously frustrating task; it tends to be tedious and time consuming. In this work we anchor the discussion in Doodle[1], which is today the most commonly used web-based tool for group scheduling. In Doodle, the sole convener sends out a list of time slots to a group of agents (the invitees), and the agents respond individually by checking off the agreeable slots. Based on the responses the convener selects a time slot and announces it to the agents.

Doodle forces the invitees to examine many time slots. This can be *inconvenient* because agents need to block their slots off until the schedule is announced. One alternative for the convener is to select and poll only a subset of slots, and if no suitable slot is found among them, then to repeat with another batch of time slots. We call this a *B-Doodle* mechanism (for "Batched Doodle"). Clearly, Doodle is a special case with one batch consisting of all time slots. Another extreme case is the OAAT (one-at-a-time) mechanism, in which the convener tests a single time slot at each iteration. While Doodle optimizes for time (the number of iterations needed), OAAT optimizes for convenience (the number of time slots to be examined). In-between lie many other mechanisms that trade off time against inconvenience differently.

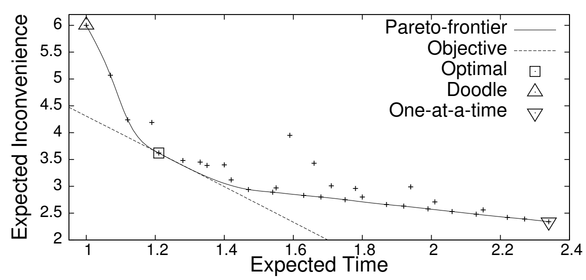

Figure 1: Pareto-frontier and objective function along with 32 B-Doodle mechanisms.

Figure 1 illustrates this. In this example there are six time slots and four agents, each of whom is available at each time slot independently with probability .8. The figure depicts a scatter plot of all 32 B-Doodle mechanisms, including Doodle and OAAT. The y-axis depicts the expected inconvenience and x-axis the expected time. The solid curve clearly shows the Pareto-frontier of Time-Inconvenience. If there is a known cost function aggregating time and inconvenience (such as 3·Time+Inconvenience), one can identify an optimal B-Doodle mechanism along this frontier by projecting the objective function (shown as the dotted line). The optimal mechanism is the "Half-n-half" mechanism that sends out 3 time slots in the first batch (and if no feasible slot is found, then it sends out the remaining 3 slots).

In this paper we investigate the difficulty of determining the optimal B-Doodle mechanism, and to what degree it improves on the simple Doodle in realistic scenarios. We will be assuming throughout the paper that the availability of agents for each time slot is independent (of their availability at other time slots, and of the availability of other agents).

[1] `http://www.doodle.com`

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.*

2. RELATED WORK

The most closely related work of which we are aware is done by Franzin et al. [1] and Garrido and Sycara [2] respectively. These papers are superficially different, as their main focus is the impact of privacy considerations (specifically, different levels of calendar sharing among agents) on time and quality, whereas we concentrate on the interaction between time spent and inconvenience caused during the scheduling process. A bigger difference lies in the mechanisms considered; we explore the space of Single-proposer Mechanisms (SPMs) in which the sole convener is trying to schedule a meeting, whereas they explore two specific mechanisms that fall outside the scope of SPMs as any agent can propose a schedule to others (this falls into Multi-proposer Mechanisms). Another crucial difference lies in the nature of the results; neither of the two papers proposes optimal solutions, and do not compare their proposed solutions to common web-based solutions such as Doodle (nor could they have, since no web-based solutions existed at the time).

3. FORMAL MODEL

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 the convener knows all such priors (i.e. all $p_{i,t}$'s), she does not know the realization of availability of agents – it is private information of each agent. The convener is trying 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 a B-Doodle mechanism (denoted by B_m where m is the number of subsets in the partition).

Definition 1 (Cost function)**.** A cost function c takes two integers j and b as arguments, and $c(j, b) > 0$ describes the aggregate cost of Time and Inconvenience that is incurred by sending out a batch of 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 $b_k = |S_k|$.

Definition 2 (Group Scheduling Problem)**.** An instance of the Group Scheduling Problem (GSP) is a tuple (N, S, f, c, P) 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 ($0 \leq f \leq 1$), c is a cost function, and P is probability distributions of availability ($P = \{p_{i,t}\}$ with $0 \leq p_{i,t} \leq 1$ for all i, t). 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).

4. ALGORITHM

Theorem 1. There exists an algorithm that runs in polynomial time, and returns the optimal B-Doodle that minimizes the expected cost, given an instance (N, S, f, c, P) of GSP.

5. EXPERIMENTAL RESULTS

Using our algorithm in simulations, we 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.

6. DISCUSSION

We formally defined two dimensions of optimality in group scheduling : Time and Inconvenience. We generalized the popular Doodle mechanism to a class of B-Doodle mechanisms that partition time slots into batches, and presented an efficient algorithm for finding the optimal B-Doodle mechanism, assuming probabilistic independence. We observed in simulations that optimal B-Doodle mechanism is superior to Doodle in many realistic situations.

While useful in and of itself, this work leaves many open questions. In realistic situations a group of agents often negotiate among themselves or respond with counter-offers. It will be interesting to extend our B-Doodle mechanisms to such multi-proposer mechanisms. Another obvious extension of our work is to relax the independence assumption and develop an adaptive mechanism that infers availability of agents based on their responses. In addition, we implicitly assumed that agents prefer all time slots equally, but people often have specific preferences. Thus it will be important to study the trade-off between optimizing the cost of scheduling process and finding a 'good' schedule. Finally, in practice an invitee may have an incentive to delay her response or to lie about her availability or preferences. Practical mechanisms should not be vulnerable to such strategic behavior of invitees, and this is yet another direction for future work.

7. 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.

8. REFERENCES

[1] M. S. Franzin, F. Rossi, E. C. Freuder, and R. Wallace. Multi-agent constraint systems with preferences: Efficiency, solution quality, and privacy loss. *Computational Intelligence*, 20(2):264–286, 2004.

[2] L. Garrido and K. Sycara. Multi-agent meeting scheduling: Preliminary experimental results. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 95–102, 1996.