On a Theory of Computing Symposia *

Avrim Blum
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3891
avrim@cs.cmu.edu

Prabhakar Raghavan
IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95120
pragh@almaden.ibm.com

Abstract

We consider the problem of scheduling a conference to achieve the benefits in time-compression of parallel sessions, but without the associated high degree of conflict between talks. We describe a randomized construction meeting these goals that we analyze based on an expansion property of an associated graph. We also give algorithms for attendees scheduling their time within such a conference, and algorithms for verifying a proposed conference schedule. Finally, we present simulation results for typical conference sizes.

1 Introduction

Single sessions or parallel sessions? It is a continuing debate in the scheduling of our premier research conferences. On the one hand, parallel sessions allow for more talks to be presented in a given time period. On the other hand, parallel sessions result in many attendees being frustrated by conflicts. For instance, if an attendee is interested in a constant fraction of the talks, then on average (say the schedule is determined randomly) this attendee will miss a constant fraction of those (s)he wants to see. Besides the annoyance to attendees, this forces choices to be made that encourage fragmentation of the field.

In this paper we discuss a proposal for achieving the compression of parallel sessions but without the associated high rate of conflict. In particular, suppose instead of parallel sessions we have *four* sessions, with *each talk given twice*.[1] Sessions 1 and 2 each have half the talks and are designed by the Program Committee as usual. (In other words, the conference

 These ideas were formulated following a discussion on the topic of parallel sessions at the business meeting of the 1997 Symposium on Theory of Computing.

[1]This sort of proposal has come up from time to time in the past, but we believe this is the first time its behavior with respect to resolving conflicts has been analyzed.

restricted to just Sessions 1 and 2 looks like a standard parallel-session conference.) Session 3 is a random permutation of session 1, and session 4 is a random permutation of session 2. So, the n talks are still being given over a period of $n/2$ time slots. In this case, we show that for any polynomial number of attendees, each wishing to see up to αn talks (for $\alpha > 0$ a sufficiently small constant), with probability $1 - o(1)$, *every* attendee will be able to see *all* their desired talks. This holds even for the case of an adversarial program committee that designs sessions 1 and 2 with the express purpose of frustrating the attendees. Simulation results for a typical conference size are given at the end of this paper.

We consider also several related issues. For instance, given a conference schedule in our 4-session format, we show that the "attendee time-scheduling problem" (given a conference schedule, how can you plan your time so that you see the talks that you want) can be solved in linear time. We also discuss the algorithmic problem of verifying the "goodness" of a conference schedule. In particular, we show that one can check whether all sets of talks of size up to s are conflict-free in time $n \cdot 2^{O(s)}$, rather than the naive $O(n^s)$ time.

2 Theoretical Results

Formally, we consider the problem of scheduling n *talks* in a conference of t *time-slots*, in which each talk is given b times, and there are nb/t parallel sessions. We will focus mostly on the case of $t = n/2$ and $b = 2$, so there are 4 parallel tracks.

A convenient way to view a conference schedule is as a bipartite graph. Each talk is represented by a node on the left, each time-slot is represented by a node on the right, and there is an edge between a talk and a time-slot if that talk is being presented in that time-slot. Using this representation, it is easy to see that the "attendee time-scheduling problem" (given a list of talks you want to see, how can you plan your time so that you see as many as possible) is a maximum matching problem and so can be solved in polynomial time. In fact, for the case in which each talk is given at most twice, a linear-time algorithm exists, and is described below in Section 2.1.

We will say that a set of talks S suffers a *compression* if $|N(S)| < |S|$. We say that a set of talks S has a *conflict* if it is not possible to see all the talks in S given the conference schedule. By Hall's Theorem, a set of talks S has no conflicts if and only if no $S' \subseteq S$ suffers a compression.

Given Hall's Theorem, the ideal bipartite graph for the conference is an expander: if *every* set S of at most αn talks has the property that $|N(S)| \geq |S|$, then every possible set of at most αn desired talks can be seen without conflict. (Note this is a bit different from the usual notion of an expander which requires $|N(S)| \geq (1 + \epsilon)|S|$, but is roughly equivalent since the right-hand side of our graph is only half the size of the left-hand side.) Unfortunately, to produce an expander requires degree 3 on the left; i.e., each talk to be given 3 times. If each talk is given "only" twice, then *any* schedule of n talks into $n/2$ time slots will result in the existence of a set of $O(\log n)$ talks that cause a conflict (see Theorem 2). Nonetheless, we show that giving each talk twice *is* sufficient, with high probability, so long as there are only a polynomial number of attendees.

As mentioned above we consider the following method for scheduling the talks. First, sessions 1 and 2 are an arbitrary partition of the n talks into two lists of length $n/2$; that is, they are an arbitrary legal parallel-session schedule. Session 3 is a *random* permutation of session 1, and session 4 is a *random* permutation of session 2. We show the following:

Theorem 1 *For any polynomial $p(n)$ there exists a constant $\alpha > 0$ such that if $p(n)$ attendees each want to see αn talks, then with probability $1 - o(1)$, the randomized scheduling method described above allows all attendees to see all their desired talks.*

The analysis proceeds in two steps. We first consider small sets of talks, showing that with "reasonably" high probability, *all* sets of at most $\frac{1}{6}\ln n$ talks can be seen without conflict. We then consider large sets, and show that for any *fixed* set S of at most αn talks, with high probability no non-small subset (a subset of size $\geq \frac{1}{6}\ln n$) of S suffers a compression. These together give our desired result.

Lemma 1 *Let B_k be the event that some set of at most k talks suffers a compression. Then* $\Pr[B_k] \leq \frac{1}{n}[e^{2k}2^{3k+1}]$.

Proof. Consider a fixed set S of k talks, with k_1 talks in session 1 and $k_2 = k - k_1$ talks in session 2. Let k_3 be the number of time-slots occupied by these talks in sessions 1 and 2 combined. (So, $k_1 + k_2 \geq k_3 \geq \max(k_1, k_2)$). Then,

$$\begin{aligned}
\Pr(S \text{ is compressed}) &\leq \frac{\binom{n}{k-1-k_3}\binom{k-1}{k_1}\binom{k-1}{k_2}}{\binom{n}{k_1}\binom{n}{k_2}} \\
&\leq \frac{(ne/(k-1-k_3))^{k-1-k_3}((k-1)e/k_1)^{k_1}((k-1)e/k_2)^{k_2}}{(n/k_1)^{k_1}(n/k_2)^{k_2}} \\
&= \frac{1}{n^{k_3+1}}\left[\frac{e^{2k-1-k_3}(k-1)^k}{(k-1-k_3)^{k-1-k_3}}\right]. \qquad (1)
\end{aligned}$$

The number of different sets of talks S occupying k_3 time-slots in sessions 1 and 2 is at most $\binom{n}{k_3}2^{k_3}2^{k_3} \leq (4ne/k_3)^{k_3}$. Therefore (using the fact that for a given k_3 our bound is increasing with k),

$$\begin{aligned}
\Pr[B_k] &\leq \sum_{k_3 \leq k} \frac{e^{2k-1}2^{2k_3}}{n}\left[\frac{(k-1)^k}{(k-1-k_3)^{k-1-k_3}k_3^{k_3}}\right] \\
&\leq \sum_{k_3 \leq k} \frac{e^{2k-1}2^{k+2k_3}}{n},
\end{aligned}$$

where the last step uses the inequality $a^a b^b \geq ((a+b)/2)^{a+b}$. This gives us our desired bound. □

Lemma 2 *For a fixed set S of αn talks, the probability that some subset of S of size* **at least** *k suffers a compression is at most $\frac{1}{n}(16\alpha e^4)^{k/2}\left(\frac{1}{1-16\alpha e^4}\right)$.*

Proof. The probability that a fixed set $S' \subseteq S$ of k talks suffers a compression, given that the talks of S' use up k_3 time-slots in sessions 1 and 2, is at most the quantity given in Equation (1). The *number* of sets $S' \subseteq S$ using k_3 time-slots in sessions 1 and 2 is at most $\binom{\alpha n}{k_3} 2^{2k_3}$. Therefore, the probability that some set $S' \subseteq S$ using k_3 time-slots in sessions 1 and 2 (and having at most $2k_3$ talks total) suffers a compression is at most $\frac{1}{n}\left(\alpha \cdot 16e^4\right)^{k_3}$. Thus, the probability that *any* $S' \subseteq S$ with at least k talks suffers a compression is at most

$$\sum_{k_3=k/2}^{\alpha n} \frac{\left(16\alpha e^4\right)^{k_3}}{n} \leq \frac{\left(16\alpha e^4\right)^{k/2}}{n}\left(\frac{1}{1-16\alpha e^4}\right). \quad \square$$

Proof of Theorem 1. Lemma 1 implies that with probability $1 - o(1)$, no set of size $\leq \frac{1}{6}\ln n$ is compressed. Now, say $p(n) = O(n^\beta)$ for some constant β. Choose $\alpha = \frac{1}{16}e^{-4-12\beta}$ so that $(16\alpha e^4)^{(\ln n)/12} \leq n^{-\beta}$. Lemma 2 implies that with probability $1 - o(1)$, no subset of size $\geq \frac{1}{6}\ln n$ any of the $p(n)$ sets of desired talks suffers a compression either. $\square$

One might hope to improve on Theorem 1 (and Lemma 1) by producing a schedule such that *every* set of k talks can be seen without conflict for $k \gg \log n$. However, the following simple argument shows that this is not possible.

Theorem 2 *For any schedule of n talks into 4 sessions such that each talk is given twice, there exists a set S of $O(\log n)$ talks that suffer a compression.*

Proof. Consider a graph with a vertex for each time slot, and where a talk scheduled in time-slots i and j is represented as an edge from i to j. This graph has degree 4. Pick some arbitrary vertex in the graph and grow a breadth-first search tree from that node until at least two cross-edges are observed. (An edge from a node to itself — i.e., a talk given in only one time-slot — counts as a cross-edge.) This must occur by the time the tree has grown to depth $\lg n$ because the degree of the graph is at least 3. Consider now the two cycles induced by these two back edges. If the cycles touch (or overlap) then the union of the two cycles is our desired set S. If the cycles do not touch, then the two cycles together with the path in the tree between them (which has length at most $2\lg n$) is our desired set. $\square$

What if we allow each talk to be given 3 times? In this case, standard arguments [1, 2] show that the bipartite graph will with high probability be an expander, and therefore all sets of αn talks have the property that they can be seen without conflict.

2.1 Algorithms

We now describe two algorithms. The first is a simple linear-time algorithm for attendees to determine how to schedule their time to see as many of their desired talks as possible, given a conference schedule. The second is an algorithm for the program committee that in time $n \cdot 2^{O(k)}$ will verify whether or not in a proposed schedule all sets of k or fewer talks can be seen without conflict.

Theorem 3 *There is a linear-time algorithm that given a schedule of n talks into time-slots such that each talk is presented twice, and given a set S of desired talks, determines the maximum subset of S that can be seen without conflict.*

Proof. We use the representation described in the proof of Theorem 2. That is, there is a vertex for each time-slot and each talk is represented by an edge. Let $G(S)$ denote the subgraph induced by the talks in S, and partition $G(S)$ into its connected components. If a connected component C has more edges (talks) than vertices (time slots) then clearly these talks cannot all be seen without conflict. To see as many as possible, simply remove edges that are involved in cycles until the number of edges equals the number of vertices (at most one cycle remains), and then orient the remaining edges so that each vertex has indegree at most 1. The edges can be oriented by the following simple strategy: begin at a leaf ℓ (vertex of degree 1) in any component; orient the incident edge into ℓ, and discard ℓ and this incident edge from further consideration. When there are no leaves we must be in a component consisting only of a cycle, and can orient its edges around this cycle. $\Box$

We now consider the problem faced by a program committee that wishes to determine whether or not all possible sets of k talks can be seen without conflict. Naively this can be done in time $O(n^k)$ by checking each k-set individually. We now describe an $n \cdot 2^{O(k)}$-time algorithm, based on the fact that the graph is of constant degree. To think about the algorithm, it helps to notice that if T is a minimal "bad" set of talks, and $S = N(T)$ is the set of time slots they occupy, then each time slot in S must have at least two talks from T.

For each talk t_0, the algorithm runs the following procedure Recursive-verify with arguments $S = \{\}$ and $T = \{t_0\}$ to see if there are any bad sets of talks containing t_0.

Algorithm Recursive-verify(S, T):

1. For each talk in T, look at the two time-slots that it occupies. If either of these is not already in S, place it into S "unmarked".
2. If $|S| \geq k$ then return "OK". If $|S| < |T|$ then return "conflict".
3. For each unmarked time-slot $s \in S$, for each possible subset T_0 of at least two talks given at time-slot s (note for each s there are only a constant number of choices for T_0),
   (a) Mark s and let $T' = T \cup T_0$.
   (b) If $|T'| \leq k$ then run Recursive-verify(S, T'). If this returns "conflict" then return "conflict".
   (c) Unmark s.
4. If we get here (i.e., if none of our recursive calls returned "conflict") then return "OK".

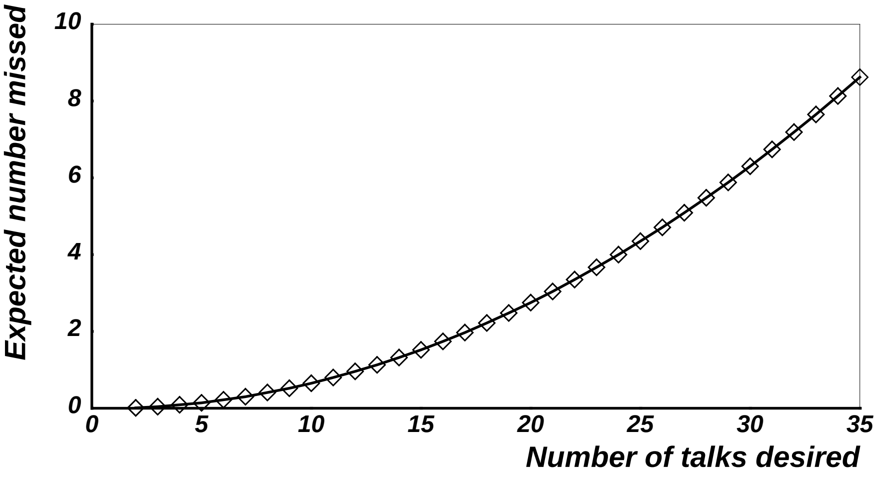

Figure 1: Expected number of talks missed in a 70-talk standard parallel-session conference for an attendee who wants to see a random subset of k talks, as a function of k.

3 Experiments

We consider a conference with 70 talks scheduled into 35 time slots. Figure 1 shows a plot for standard parallel sessions of the expected number of talks missed (conflicts) for an attendee wishing to see a random subset of k talks, as a function of k. (Alternatively, it is the expected number of conflicts for a fixed attendee, for a random assignment of the talks to time-slots.) For instance, an attendee wishing to see 21 talks would, on average, miss 3 talks; an attendee wishing to see 24 talks would, on average, miss 4; and an attendee wishing to see 27 talks would, on average, miss roughly 5. This plot is an exact calculation, as opposed to a Monte-Carlo simulation.

In Figure 2 we plot a comparison between standard parallel sessions and the 4-session proposal discussed here where each talk is given twice. This figure shows the probability of at least one conflict for an attendee wishing to see a random subset of k talks, as a function of k. The numbers for the standard parallel session conference are produced via an exact calculation. The numbers for the 4-session proposal are determined from Monte-Carlo simulation. Specifically, for each value of k, we consider 100 random conference schedules. For each of these schedules we pick 1000 random subsets of k talks. We then compute the fraction of times that the subset of talks has a conflict and average the 100 results. We also plot a slightly improved version (shown with a dotted line) in which we generate random conference schedules *subject to* the constraint that the permutations have no fixed points (i.e., each talk is given at two *different* time slots). These simulations can be viewed as empirical estimates of the probability an arbitrary set of k talks would have a conflict, for a random program committee (a random schedule of sessions 1 and 2), that perhaps also chooses its permutations for sessions 3 and 4 subject to not having fixed points. Note, however, that the theoretical results of the previous section hold even in the case of a program committee that *adversarially* schedules sessions 1 and 2 so as to maximize conflicts among your desired set of talks.

As can be seen in Figure 2, the 4-session method can dramatically reduce the probability

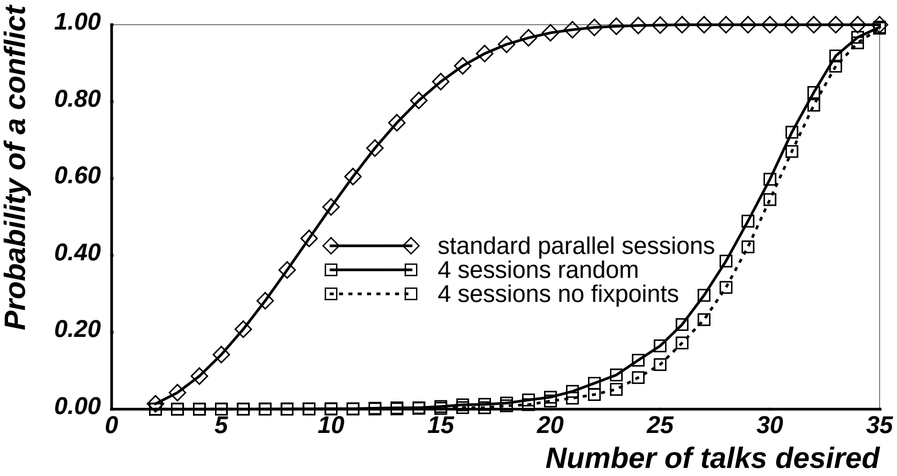

Figure 2: Probability of conflict versus desired number of talks for a 70-talk conference. The top line is for a standard parallel session conference. The middle line is for a 4-session conference in which sessions 3 and 4 are random permutations of sessions 1 and 2. The lowest (dotted) line is for a 4-session conference in which sessions 3 and 4 are random permutations of sessions 1 and 2, but subject to the constraint that the permutation has no fixed points.

of a conflict. For instance, with standard parallel sessions, a (random) attendee wishing to see 15 talks would have an 85% chance of a conflict, an attendee wishing to see 20 talks would have a 98% chance of a conflict, and an attendee wishing to see 25 talks would have a 99.9% chance of a conflict. On the other hand, with the 4-session proposal (the no-fixed point version), an attendee wishing to see 15 talks has only an 0.2% chance of a conflict, an attendee wishing to see 20 talks has a 2% chance of a conflict, and an attendee wishing to see 25 talks has an 11.6% chance of a conflict.

4 Conclusions

In this paper we have analyzed, theoretically and empirically, some beneficial features of a new scheme for organizing talks in a conference. By having each talk presented twice, we can decrease the length of a conference without substantially increasing the number of conflicts. The scheme can be generalized to the case of a conference with more parallel sessions in a fairly straightforward way. Indeed, such “massively parallel” scheduling need not be restricted to talks; we can apply the same principle to all conference events. Thus attendees can choose between “banquets” at 2pm and 8pm; or perhaps receptions at 11am or 6pm. What is interesting is that by giving each attendee just two choices per event, the amount of conflict is reduced dramatically. If we allow an increasing number of choices, then in the limit, this process converges to the following model: imagine a large room with a long table of food in the middle. Some researchers stand around the table describing their recent research. Others alternate between listening to such descriptions, giving their own descriptions, and fetching food from the table. In other words, the limiting view of our process tends to one massive reception-style meeting during which attendees exchange scientific ideas.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley, 1991.

[2] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.