Solution to Exchanges 20.1 Puzzle: Communicating to Plan Noam Nisan's 60th Birthday Workshop

ALEC SUN
Carnegie Mellon University

This is a solution to Vincent Conitzer's puzzle "Communicating to Plan Noam Nisan's 60th Birthday Workshop", which appeared as below in the July 2022 issue of SIGecom Exchanges.

Michael, Moshe, and Shahar—i.e., a constant number of organizers—are planning the workshop for Noam's 60th birthday, and are trying to predict who, out of n people, will attend. Whether a person wants to attend is a function of who else attends. "The more the merrier," so for each person i, if i would attend when S is the set of other attendees, and $S \subseteq S'$, then i would attend when S' is the set of other attendees. Let $\mathcal{S}_i$ be the set of sets S of other people for which i would attend (so, $\mathcal{S}_i$ is upward closed).

To split the work, the organizers partition the set of n people among themselves. Subsequently, each of them figures out, for every player i in his own part, what $\mathcal{S}_i$ is. (Note that each organizer thus still needs to think about how much "his" people like the people in the other parts. But each organizer knows $\mathcal{S}_i$ only for people i in his own part.) At this point, the organizers, who of course want the workshop to be successful, must communicate with each other to find the *largest* possible set of people S^* that can consistently attend (i.e., the largest set with the property such that every person in it will attend given that everyone else in the set attends: i.e., for each $i \in S^*$, we have $S^* \setminus \{i\} \in \mathcal{S}_i$, and S^* is the largest set with that property).

Up to a constant factor, how many bits of communication do the organizers need to figure this out?

We claim that the organizers need $\Theta(n \log n)$ bits of communication.

1. UPPER BOUND

We begin by asking the following question. When would someone not attend the birthday celebration? Certainly if $\mathcal{S}_i = \emptyset$ then person i would not attend. It turns out that this is the only possible restriction preventing someone from attending.

Claim 1.1. *If $\mathcal{S}_i \neq \emptyset$ for all i, that is, for every person i there is at least one set S of other attendees for which i would attend, then $S^* = [n]$.*

Claim 1.1 follows directly from "the more the merrier" and implies the following recursive algorithm for the birthday problem:

—If an organizer sees that $\mathcal{S}_i = \emptyset$ for a person i in their part, they communicate the identity of i to the other organizers using $O(\log n)$ bits and then remove person i from consideration.

—Since person i provably cannot attend, all of the organizers remove all sets containing i from each $\mathcal{S}_j$ in their parts. Note that the $\mathcal{S}_j$ remain upward-closed and hence we have reduced to an instance of the same problem with $n - 1$ people.

—Repeat the above procedure until among the people currently in consideration, which is possibly the empty set, there is no person i for which $\mathcal{S}_i \neq \emptyset$.

Authors' addresses: `sundogx@gmail.com`

Claim 1.1 shows that S^* is precisely the set of people remaining after the above algorithm is run. Since the identity of each person is communicated at most once in the above procedure, the total communication is $O(n \log n)$.

We remark that the above procedure is reminiscent of Moulin's mechanism for cost-sharing in which a designer decides on which players to be served and what cost to charge them through an iterative process. As long as there exists at least one player that has a cost-share strictly greater than their bid, that player is removed from consideration and new cost-shares are computed with the remaining players. This process is continued until all remaining players' cost-shares are at most their bids, at which point the mechanism terminates and serves this remaining set of players. See [Moulin 1999] and [Moulin and Shenker 2001] for more details on Moulin's mechanism.

2. LOWER BOUND

Now we prove that $\Omega(n \log n)$ bits of communication are required to compute S^*. For simplicity, suppose there are only two organizers, Michael and Moshe, and that the number n of people is even. Number the people $1, 2, \ldots, n$ and suppose that Michael knows $\mathcal{S}_i$ for all odd i, which we denote by Michael's input x to the problem, and Moshe knows $\mathcal{S}_i$ for all even i, which we denote by Moshe's input y. We call the pair (x, y), which is the aggregated set of preferences, the *input* to the problem, and we call the resulting set S^* the *answer*.

Consider the *communication transcript* of the organizers, which consists of all bits communicated between them as well as the final answer. Without loss of generality, assume that the two players alternate in communicating bits, with Michael communicating first. We want to show that the communication transcript must have size $\Omega(n \log n)$. We use the *fooling set* method, a lower bound technique in communication complexity that appears in Nisan's own book, co-authored with Kushilevitz, on the subject [Kushilevitz and Nisan 1996]. The idea of the fooling set method is that if two distinct input pairs have the same communication transcript, then we can find two other pairs that also have this same transcript.

Claim 2.1. *Let (x, y) and (x', y') be two inputs to Michael and Moshe that have the same communication transcript. Then the inputs (x', y) and (x, y') also have this same communication transcript.*

Proof. Each bit communicated by a player is a deterministic function of that player's input and the bits seen so far. Since (x, y) and (x', y') have the same transcript, the first bit b_1 sent, which depends only on Michael's input, is the same whether Michael has input x or x', so the first bits of the transcripts of (x', y) and (x, y') are also b_1. The second bit b_2 sent is a function of b_1, which we already argued is the same in all four transcripts, and Moshe's input. Since the transcripts of (x, y) and (x', y') have the same second bit b_2, (x', y) and (x, y') also have b_2 as their second bits. Continuing inductively, we deduce that (x, y), (x', y'), (x', y), (x, y') all produce identical transcripts. $\square$

Our strategy will be to find many input pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ all with the same answer S^*. By the Pigeonhole Principle, if m is large enough and the amount of communication is limited, then two of these input pairs $(x_i, y_i), (x_j, y_j)$ will have the same communication transcripts. By Claim 2.1, the input pairs (x_i, y_j) and (x_j, y_i) will also have this same communication transcript and in particular the same answer. However, if we had constructed these inputs from the start in such a way that all inputs (x_i, y_j) for $i \neq j$ actually had answers that are different than S^*, then this would yield a contradiction.

Consider all *parity-alternating* permutations $\sigma = \sigma_1 \sigma_2 \cdots \sigma_n$ of the n people such that σ_1 is odd, σ_2 is even, and so on. Note that there are $\frac{n}{2}$ ways to choose σ_1 since there are

$\frac{n}{2}$ odd people, $\frac{n}{2}$ ways to choose σ_2 since there are $\frac{n}{2}$ even people, $\frac{n}{2}-1$ ways to choose σ_3 since there are $\frac{n}{2}-1$ remaining odd people, and so on, for a total of $\left(\left(\frac{n}{2}\right)!\right)^2$ such permutations.

We say that each parity-alternating permutation $\sigma = \sigma_1 \ldots \sigma_n$ *induces* an input to the birthday problem as follows. Let $\mathcal{S}_{\sigma_i} = \{S : \exists j < i, \sigma_j \in S\}$, that is, the preference of person σ_i is "I would attend only if at least one of $\sigma_1, \ldots, \sigma_{i-1}$ attends." For $i = 1$ this simply means that person σ_1 would not attend. We note that the $\mathcal{S}_{\sigma_i}$ are indeed upward-closed. For all inputs (x, y) induced by such permutations σ, we have $S^* = \emptyset$ since σ_1 would not attend, which prohibits σ_2 from attending, and so on.

Claim 2.2. *For any two distinct parity-alternating permutations σ and σ', which induce preferences (x, y) and (x', y') respectively as inputs to Michael and Moshe, the answers to the inputs (x', y) and (x, y') are not $S^* = \emptyset$.*

Example 2.3. *Consider the permutations $\sigma = 1234$ and $\sigma' = 1432$, which induce the inputs*

$$\begin{aligned} x &= \text{"1 would not attend", "3 would only attend if 1 or 2 does"} \\ y &= \text{"2 would attend only if 1 does", "4 would attend only if 1 or 2 or 3 does"} \\ x' &= \text{"1 would not attend", "3 would attend only if 1 or 4 does"} \\ y' &= \text{"2 would attend only if 1 or 4 or 3 does", "4 would attend only if 1 does".} \end{aligned}$$

Note that if the input was (x', y) then 3 and 4 can attend together, and if the input was (x, y') then 2 and 3 can attend together.

Proof of Claim 2.2. By symmetry, we can consider only the pair (x, y'). We have

$$\begin{aligned} x =& \text{"}\sigma_1 \text{ would not attend",} \\ & \text{"}\sigma_3 \text{ would only attend if } \sigma_1 \text{ or } \sigma_2 \text{ does"}, \ldots \\ y' =& \text{"}\sigma'_2 \text{ would attend only if } \sigma'_1 \text{ does",} \\ & \text{"}\sigma'_4 \text{ would attend only if } \sigma'_1 \text{ or } \sigma'_2 \text{ or } \sigma'_3 \text{ does"}, \ldots \end{aligned}$$

If $\sigma'_1 \neq \sigma_1$, then removing person σ_1 from consideration according to the recursive algorithm in the upper bound does not make any other $\mathcal{S}_i$ empty. This is because σ'_2 could attend if σ'_1 does, and everyone else has at least two people that could allow them to attend. Hence $\sigma'_1 \neq \sigma_1$ implies $S^* = [n] \setminus \{\sigma_1\}$, so we can assume $\sigma'_1 = \sigma_1$, which means that σ'_1 and σ'_2 would not attend. If $\sigma'_2 \neq \sigma_2$, then removing σ'_1 and σ'_2 from consideration does not make any other $\mathcal{S}_i$ empty. This is because σ_3 could attend if σ_2 does, and everyone else has at least three people that could allow them to attend. Hence $\sigma'_2 \neq \sigma_2$ implies $S^* = [n] \setminus \{\sigma'_1, \sigma'_2\}$, so we can assume $\sigma'_2 = \sigma_2$. Continuing inductively, at each step we have either $S^* \neq \emptyset$ or $\sigma'_i = \sigma_i$. We conclude that either $S^* \neq \emptyset$ or $\sigma' = \sigma$. □

By Claim 2.2, the set of $m = \left(\left(\frac{n}{2}\right)!\right)^2$ parity-alternating permutations induce m input pairs $(x_1, y_1), \ldots, (x_m, y_m)$ all with the same answer $S^* = \emptyset$ but such that all inputs (x_i, y_j) for $i \neq j$ have answers $S^* \neq \emptyset$. If the number of bits of communication is less than $\log_2 m = \Omega(n \log n)$, then by the Pigeonhole Principle two parity-inducing permutations generate the same communication transcript, which is impossible by Claim 2.1. We conclude that the communication complexity of the birthday problem is $\Omega(n \log n)$.

REFERENCES

Kushilevitz, E. and Nisan, N. 1996. *Communication Complexity.* Cambridge University Press.

Moulin, H. 1999. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare 16,* 2, 279–320.

Moulin, H. and Shenker, S. 2001. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory 18,* 3, 511–533.