

Scheduling Virtual Conferences Fairly: Achieving Equitable Participant and Speaker Satisfaction

Gourab K Patro
IIT Kharagpur, India
L3S Research Center, Germany

Prithwish Jana
IIT Kharagpur, India

Abhijnan Chakraborty
IIT Delhi, India

Krishna P. Gummadi
MPI-SWS, Germany

Niloy Ganguly
IIT Kharagpur, India
L3S Research Center, Germany

ABSTRACT

In recent year, almost all conferences have moved to virtual mode due to (COVID-19) pandemic-induced restrictions on travel and social gathering. Contrary to the in-person conferences, virtual conferences face the challenge of efficiently scheduling talks, accounting for the availability of participants from different timezones and their interests in attending different talks. A natural objective for conference organizers is to maximize some efficiency measure, e.g., the total expected audience participation across all talks. However, we show that in virtual conference setting, optimizing for efficiency alone can result in an unfair schedule, i.e., individual utilities for participants and speakers can be highly unequal. To address such unfairness concerns, we formally define fairness notions for participants and speakers, and subsequently derive suitable objectives to account for them. As the efficiency and fairness objectives can be in conflict with each other, there is a need to maintain balance between these objectives. Thus, we propose a joint optimization framework that allows conference organizers to design schedules that balance (i.e., allow trade-offs) among efficiency, participant fairness and speaker fairness objectives. While the optimization problem can be solved using integer programming to schedule smaller conferences, we provide two techniques to significantly scale up the joint optimization framework to cater to bigger conferences. Extensive evaluations over multiple synthetic and real-world datasets show the efficacy and flexibility of our proposed approaches.

CCS CONCEPTS

• **Theory of computation** → *Scheduling algorithms*.

KEYWORDS

Fair Conference Scheduling, Virtual Conference Scheduling

ACM Reference Format:

Gourab K Patro, Prithwish Jana, Abhijnan Chakraborty, Krishna P. Gummadi, and Niloy Ganguly. 2022. Scheduling Virtual Conferences Fairly: Achieving Equitable Participant and Speaker Satisfaction. In *Proceedings of*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9096-5/22/04...\$15.00

<https://doi.org/10.1145/3485447.3512136>

the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3485447.3512136>

1 INTRODUCTION

Restrictions on travel and social gatherings to tackle the COVID-19 pandemic have forced almost all conferences to move online, and some of them may remain online in future due to the benefits online conferences offer. They are hugely economical due to reduced organizational costs, and they foster inclusivity by significantly improving the scale and outreach [2]. However, online conferences have their own set of challenges, such as, scaling up participation depends on stable and high-speed Internet in different regions; sometimes participants and speakers need to be trained on different conferencing tools for efficient participation [29]. A big challenge in organizing an online conference is *optimal scheduling of the conference talks*. Online conferences usually have participants from all around the globe, unlike the physical conferences where participants assemble at a single place. Thus, traditional timezone-specific conference schedules (based on the timezone of the venue) are no longer suitable for online conferences, as the participants from the other parts of the globe will find it hard to attend ([9, 26]). This demands for conference schedules to be *timezone-aware* instead of *timezone-specific*, and may stretch beyond the usual 7-8 hours a day, to cater to participants from different timezones. In this paper, we focus on this conference scheduling problem and associated concerns about efficiency and fairness.

In conference scheduling, a natural objective for organizers would be to maximize an efficiency measure, such as the *total expected audience participation across all talks*—similar to the participation metrics used in prior literature on optimal meeting scheduling [5, 6, 13, 24, 27]. While earlier works have focused on optimizing efficiency, optimizing for such objective alone in online conference settings can result in a schedule where the level of satisfaction of individual participants may vary widely, and the expected exposure (audience size) at different talks can be highly skewed—leading to disparity in speaker satisfactions. Note that the prior works on meeting scheduling [5, 6, 24] model only the participant satisfaction, but have no concept of speaker satisfaction. Intuitively, a participant would be less satisfied if her favorite talks are scheduled in timeslots unfavorable for her, and similarly a speaker would be less satisfied if her talk is scheduled in a timeslot that adversely limits the expected audience or crowd at her talk. Thus the organizers need to consider fairness along with efficiency.

We formally define the conference scheduling problem in §3 alongside suitable measures of participant satisfaction, speaker satisfaction, and efficiency. Intuitively, a schedule would be fair if it ensures equity of satisfaction among individual participants as well as among the speakers. We formally define the fairness notions in sections 3.4.1 and 3.4.2. We further show that it may be impossible to maximize efficiency, participant and speaker fairness objectives simultaneously as there are fundamental tensions among these objectives (more details in §4.1) — optimizing one objective could cause losses in other objectives. Thus, we propose a joint optimization framework (§4.2) which allows conference organizers to design schedules that balance (i.e., allow trade-offs) among the efficiency, participant fairness and speaker fairness objectives. We show that it can be solved using integer programming.

While the integer program solution can be used for scheduling small conferences (§5.2.1, §5.2.2), they may not scale well for scheduling bigger conferences because of the hardness of the objectives. Thus, we propose two techniques (a repeated rounding technique in §4.3.1 and participant clustering in §4.3.2) to significantly scale up the joint optimization framework. We use data from three real conferences: FATREC ([piret.gitlab.io/fatrec](https://github.com/fatrec)), RecSys (recsys.acm.org/recsys17) and ICML (icml.cc/Conferences/2017), respectively covering small, medium and large conference categories, and test the efficacy of our proposed method. Extensive evaluations over these real-world datasets (alongside synthetic datasets) show that the proposed approaches are effective in providing a nice balance between efficiency and fairness objectives (§5). To the best of our knowledge, this is the first work to consider fairness in virtual conference scheduling. We hope that our proposal will not only help conference organizers, but also spawn future research on fine-tuning solutions to specific type of virtual conferences.

2 RELATED WORK

We briefly review related research efforts in the following two directions: job scheduling and event scheduling.

Job and Network Scheduling: The most commonly studied scheduling problem in computing research is job/network scheduling. The general setting of the problem is as follows: there are multiple agents (e.g., system processes, computing jobs, data packets, networked users or machines) who have shared access to common resource(s) (e.g., fixed number of processors, limited internet bandwidth), and the agents raise requests for using the common resource(s) from time to time; now the goal is to allocate the resource(s) to the agents in a fair and optimal manner. Examples include fair-share scheduling for system processes [18, 21, 22], scheduling packet transfers to ensure fair sharing of network channels [33], fair scheduling of computing jobs on computing clusters [17, 23], scheduling for devices in shared wireless charging systems [10], and fair scheduling of retrieval queries for databases [16]. Our problem setup for fair conference scheduling is very different from a typical job scheduling setup. While conference scheduling has two types of stakeholders—participants and speakers—whose functions and fairness requirements are different from each other, job scheduling problems are usually modeled only for the agents who use the shared resource.

Meeting/Event Scheduling: The problem which is closely related to the conference scheduling problem is meeting or event scheduling where there are multiple agents with different availability in different time intervals, and the goal is to find an optimal schedule for meeting(s). Some works [6, 27] also capture participants' personal preferences for over the set of events as it can affect their participation. We consider both the availability and preferences/interests of the participants while modeling their satisfaction in §3. The optimality of a schedule has been predominantly associated with its efficiency in bringing more participation [5, 13, 24]. We, too, model this as the efficiency metric which captures the total expected participation given a schedule (as in §3.3). While most of the works have relied on centralized scheduling architecture, a line of works [11, 20, 25, 31, 34] explore the decentralized scheduling due to privacy concerns from the participants' side. We model the conference scheduling problem using the former one. In meeting scheduling, utility/satisfaction is modeled only for the participants, and there has been no consideration of satisfaction from the side of the event (i.e., no concept of speakers as individuals with self interests). In sharp contrast, the speakers in a conference also have satisfaction attached to them (§3.2). Meeting scheduling has focused more on optimizing participation, fairness has received little attention in such settings (except for Baum et al. [4] dealing with a very different context). Besides there have been very few works on conference scheduling, but they focus on in-person settings [28, 32], efficiency objectives [3, 7]. Our FairConf framework not only accommodates the basic concepts of satisfaction of participants and speakers, but also cares for both efficiency and fairness in conference scheduling.

3 PRELIMINARIES

Problem Setup: In a conference, let \mathcal{P} , \mathcal{T} , and \mathcal{S} represent the sets of participants, planned talks, and available slots (non-overlapping) respectively; $|\mathcal{P}| = m$, $|\mathcal{T}| = n$, $|\mathcal{S}| = l$; let $p \in \mathcal{P}$, $t \in \mathcal{T}$, and $s \in \mathcal{S}$ be instances of participant, talk, and slot respectively. Assuming a talk to be scheduled only once, a conference schedule Γ is a mapping $\Gamma : \mathcal{T} \rightarrow \mathcal{S}$. Note that, in this paper, we limit ourselves to the case with no parallel or overlapping time slots; this implies that each slot refers to a unique time interval. Thus, the conference schedule Γ is a one-to-one mapping with $n \leq l$. The goal of a conference scheduling problem is to find a schedule Γ which satisfies some specified constraint(s) or optimizes some specified objective(s). Further on, we use **VCS** as abbreviation for **Virtual Conference Scheduling**.

Interest Scores [$V_p(\cdot)$]: The participants may have different preference levels over the set of talks. We model this phenomenon using participant-specific interest scores. Let $V(t|p) = V_p(t)$ represent p 's interest score for talk t . Note that the interest score represents the probability of satisfaction of the participant on attending the corresponding talk; i.e., $V_p(t) \in [0, 1]$, $\forall p \in \mathcal{P}, t \in \mathcal{T}$.

Ease of Availability [$A_p(\cdot)$]: In a virtual conference setting, the participants are located in different parts of the world which makes it convenient for them to attend talks only in specific times of the day (usually during the day time of their timezone). Note that participants from same timezone may also have different ease of availability throughout the 24-hour period. Thus, we model this phenomenon using participant-specific availability scores. Let $A(s|p) = A_p(s)$

represent the ease of availability score or the probability of p making herself available in slot s ; i.e., $A_p(s) \in [0, 1]$, $\forall p \in \mathcal{P}, s \in \mathcal{S}$.

3.1 Participant Satisfaction (NCG)

In virtual conferences, a participant's satisfaction depends on both her interest for the talks and her ease of availability in the time slots when the talks are scheduled. For simplicity, we assume $V_p(\cdot)$ and $A_p(\cdot)$ to be independent of each other, i.e., the interest score of a talk does not affect the ease of availability of a participant in a slot and vice-versa. However, the expected gain of a participant p from a talk t in slot s will depend on the joint probability of p making herself available in s and getting satisfied after attending t : i.e., $V_p(t) \times A_p(s)$; which in turn represents the probability of p attending talk t in slot s . Thus, given a conference schedule Γ , we define the cumulative gain (CG) of participant p as below.

$$CG(p|\Gamma) = CG_p(\Gamma) = \sum_{t \in \mathcal{T}} V_p(t) \times A_p(\Gamma(t)) \quad (1)$$

Now, let's imagine a situation wherein the participant p is asked to choose the conference schedule Γ . Assuming p to be a selfish and rational agent, she would choose the schedule which benefits her the most; i.e., the one which gives her the highest cumulative gain. Here, the best conference schedule for p would be the one in which the talk with the highest V_p is scheduled in the slot with the highest A_p , the talk with second highest V_p is scheduled in the slot with the second highest A_p , and so on (this also follows from the Rearrangement inequality [15]); let Γ_p^* be that best conference schedule for p . We call the cumulative gain of p from schedule Γ_p^* as her ideal cumulative gain (ICG): $ICG(p) = ICG_p = \max_{\Gamma} CG_p(\Gamma) = CG_p(\Gamma_p^*)$. We now define the overall satisfaction of a participant as her normalized cumulative gain (NCG) as below.

$$NCG(p|\Gamma) = NCG_p(\Gamma) = \frac{CG_p(\Gamma)}{ICG_p} \quad (2)$$

Since the denominator ICG_p is the maximum possible cumulative gain for the participant p , $NCG_p(\Gamma) \in [0, 1]$, $\forall p \in \mathcal{P}, \forall \Gamma$.

3.2 Speaker Satisfaction (NEC)

We model the satisfaction of a speaker using the expected participation or crowd at her talk. To have high speaker participation, the talk needs to be scheduled in a slot with high ease of availability of the interested participants. Thus, given a schedule Γ , we define expected crowd (EC) at talk t as below.

$$EC(t|\Gamma) = EC_t(\Gamma) = \sum_{p \in \mathcal{P}} V_p(t) \times A_p(\Gamma(t)) \quad (3)$$

Now if the speaker of talk t is asked to prepare the conference schedule, she would try to maximize her expected crowd (assuming that similar to participants, speakers are also selfish and rational agents). Such a schedule can be easily constructed by searching through the set of available slots to find the slot with the highest expected crowd for t , and then randomly allocating the remaining talks to the remaining slots. Let that best schedule for t be denoted as Γ_t^* . We call the expected crowd at talk t with schedule Γ_t^* as the ideal expected crowd (IEC) of t : $IEC(t) = IEC_t = \max_{\Gamma} EC_t(\Gamma) = EC_t(\Gamma_t^*)$. IEC represents the maximum value for expected crowd at the talk. We now define the overall satisfaction of a speaker as the

normalized expected crowd (NEC) at her talk as below.

$$NEC(t|\Gamma) = NEC_t(\Gamma) = \frac{EC_t(\Gamma)}{IEC_t} \quad (4)$$

Since the denominator IEC_t is the maximum possible value of the expected crowd at talk t , thus, $NEC_t(\Gamma) \in [0, 1]$, $\forall t \in \mathcal{T}, \forall \Gamma$.

3.3 Efficiency Objective (TEP)

From a mechanism design perspective, a natural objective for the conference organizers is to maximize the efficiency, i.e., the total participation in the conference – similar to the participation metrics used in prior literature on optimal meeting scheduling [5, 6, 13, 24, 27]. Given a talk t scheduled in slot s , the probability of participant p attending it, can be written as $V_p(t) \times A_p(s)$. Thus, the total expected participation (TEP) in the conference with schedule Γ can be written as below.

$$TEP(\Gamma) = \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} V_p(t) \times A_p(\Gamma(t)) \quad (5)$$

It is worth noting that the efficiency TEP, here, is same as: (i) the sum of cumulative gains of all the participants; (ii) the sum of expected crowd at all the talks; both are in the form of a utilitarian social welfare function [30]. We use Γ^{EM} to represent the schedule which maximizes the efficiency; i.e., $\Gamma^{\text{EM}} = \text{argmax}_{\Gamma} TEP(\Gamma)$.

LEMMA 1. *Efficiency maximization in VCS can be mapped to a min cost bipartite matching problem with polynomial time solution.*

3.4 Fairness in Conference Scheduling

3.4.1 Participant Fairness. We postulate that disparity in normalized satisfactions may cause participant unfairness, and to ensure fairness for participants, the conference schedule should equally satisfy all the participants. However, such a hard constraint might become infeasible in real-world cases. Thus, we define a relaxed unfairness measure for participants as below.

DEFINITION 1. **Participant Unfairness ($\Psi^P(\Gamma)$):** The participant unfairness caused by a schedule Γ , is the maximum difference between the satisfactions of any two participants.

$$\Psi^P(\Gamma) = \left\{ \max_{p_i \in \mathcal{P}} NCG(p_i|\Gamma) \right\} - \left\{ \min_{p_j \in \mathcal{P}} NCG(p_j|\Gamma) \right\} \quad (6)$$

Using the above unfairness measure, the **fairness objective for participants** can be defined as finding the schedule Γ which minimizes $\Psi^P(\Gamma)$.

$$\text{argmin}_{\Gamma} \Psi^P(\Gamma) \equiv \text{argmin}_{\Gamma} \left\{ \left\{ \max_{p_i \in \mathcal{P}} NCG(p_i|\Gamma) \right\} - \left\{ \min_{p_j \in \mathcal{P}} NCG(p_j|\Gamma) \right\} \right\} \quad (7)$$

We now give the decision variant of the participant fairness objective which is NP-complete (as in definition 2 and theorem 3.4.1; proof is given in the supplementary material).

DEFINITION 2. **Decision variant of participant fairness:** Given $\mathcal{P}, \mathcal{T}, \mathcal{S}$, and $V_p(t), A_p(s) \forall p, t, s \in \mathcal{P}, \mathcal{T}, \mathcal{S}$, and $\epsilon \in \mathbb{R}^{\geq 0}$, does there exist a schedule Γ (a mapping from \mathcal{T} to \mathcal{S}) such that $\Psi^P(\Gamma) \leq \epsilon$?

THEOREM 3.4.1. *The participant fairness problem (as given in definition 2) is NP-complete.*

3.4.2 Speaker Fairness. Similar to participant fairness, we define the unfairness measure for speakers and then the corresponding fairness objective.

DEFINITION 3. Speaker Unfairness ($\Psi^S(\Gamma)$): The speaker unfairness caused by a schedule Γ , is the maximum difference between the satisfactions of any two speakers.

$$\Psi^S(\Gamma) = \left\{ \max_{t_i \in \mathcal{T}} NEC(t_i | \Gamma) \right\} - \left\{ \min_{t_j \in \mathcal{T}} NEC(t_j | \Gamma) \right\} \quad (8)$$

Now, the **fairness objective for speakers** can be defined as finding the schedule Γ which minimizes $\Psi^S(\Gamma)$.

$$\argmin_{\Gamma} \Psi^S(\Gamma) \equiv \argmin_{\Gamma} \left\{ \left\{ \max_{t_i \in \mathcal{T}} NEC(t_i | \Gamma) \right\} - \left\{ \min_{t_j \in \mathcal{T}} NEC(t_j | \Gamma) \right\} \right\} \quad (9)$$

4 BALANCING EFFICIENCY AND FAIRNESS

In VCS (as defined in §3), the ultimate goal is to find a schedule Γ that optimizes efficiency while minimizing participant unfairness and speaker unfairness (as defined in eq. (5), eq. (7), eq. (9) respectively). However, the individual objectives may be at loggerheads with each other, and simultaneous optimization of the three objectives may not be possible. Thus, first in §4.1, we highlight the potential conflicts in simultaneously ensuring efficiency and fairness. Then, in §4.2, we propose a joint optimization framework for the problem, and in §4.3, we propose two techniques to scale up the joint optimization for scheduling bigger conferences.

4.1 Tension between Efficiency and Fairness

In this section, through a set of claims, we illustrate some fundamental tensions between efficiency and fairness in VCS. Due to space constraints, proofs are omitted and given in the appendix.

CLAIM 1. In VCS, it is not always possible to gain participant fairness without losing efficiency.

CLAIM 2. In VCS, it is not always possible to gain speaker fairness without losing efficiency.

CLAIM 3. In VCS, it is not always possible to gain speaker fairness without losing participant fairness and vice-versa.

From the above three claims, we can say that improving on one of the three objectives might cause losses in the other two objectives. Thus, while designing a conference schedule, a suitable balance among the two fairness objectives and the efficiency objective should be maintained. Hence to attain such a balance, next, we propose a framework to jointly optimize these objectives.

4.2 Joint Optimization for Efficiency and Fairness (FairConf)

We propose the following joint optimization framework FairConf, which combines participant and speaker fairness objectives with efficiency.

$$\begin{aligned} \argmax_{\Gamma} \frac{TEP(\Gamma)}{mn} + \lambda_1 \times & \left\{ \left\{ \min_{p_j \in \mathcal{P}} NCG(p_j | \Gamma) \right\} - \left\{ \max_{p_i \in \mathcal{P}} NCG(p_i | \Gamma) \right\} \right\} \\ + \lambda_2 \times & \left\{ \left\{ \min_{t_j \in \mathcal{T}} NEC(t_j | \Gamma) \right\} - \left\{ \max_{t_i \in \mathcal{T}} NEC(t_i | \Gamma) \right\} \right\} \quad (10) \end{aligned}$$

Here we normalize the efficiency objective to bring all the three components to similar scales; i.e., TEP is divided by $|\mathcal{P}| \cdot |\mathcal{T}| = mn$ (it is the maximum possible value for TEP —occurs when $V_p(t) = A_p(s) = 1, \forall p, t, s$). We also reverse the fairness objective functions from eq. (7) and eq. (9) while inserting them in eq. (10) as it features argmax instead of argmin, and use λ_1, λ_2 as weights for participant fairness and speaker fairness respectively.

We take a matrix X of dimensions $|\mathcal{T}| \times |\mathcal{S}|$. Each element of X : $X_{t,s}$ is a binary indicator variable for talk $t \in \mathcal{T}$ being scheduled in slot $s \in \mathcal{S}$, i.e., $X_{t,s} = 1$ if t is scheduled in s and 0 otherwise. Now to operationalize the joint optimization objective in eq. (10), we express it as the following integer program.

$$\begin{aligned} \argmax_X \quad & \frac{1}{mn} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} V_p(t) \cdot A_p(s) \cdot X_{t,s} \\ + \lambda_1 \left[\min_{p_j \in \mathcal{P}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}} \frac{V_{p_j}(t) A_{p_j}(s)}{ICG(p_j)} X_{t,s} - \max_{p_i \in \mathcal{P}} \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}} \frac{V_{p_i}(t) A_{p_i}(s)}{ICG(p_i)} X_{t,s} \right] \\ + \lambda_2 \left[\min_{t_j \in \mathcal{T}} \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} \frac{V_p(t_j) A_p(s)}{IEC(t_j)} X_{t_j,s} - \max_{t_i \in \mathcal{T}} \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} \frac{V_p(t_i) A_p(s)}{IEC(t_i)} X_{t_i,s} \right] \\ \text{s.t. } & X_{t,s} \in \{0, 1\} \quad \forall t \in \mathcal{T}, s \in \mathcal{S} \\ & \sum_{s \in \mathcal{S}} X_{t,s} = 1, \quad \forall t \in \mathcal{T} \\ & \sum_{t \in \mathcal{T}} X_{t,s} \leq 1, \quad \forall s \in \mathcal{S} \end{aligned} \quad (11)$$

Here the first constraint is the integrality constraint. Second constraint ensures that, each talk gets scheduled exactly once. On the other hand, one slot can be allocated to atmost one talk which is ensured by the third constraint.

4.3 Scaling Up the Joint Optimization

As the joint objective is NP-hard (theorem 3.4.1), scaling the integer program (as given in eq. (11)) to big conferences with large number of talks and participants, would need huge computing resources. Thus, we provide a rounding heuristic (in §4.3.1), and a clustering approach (in §4.3.2) which can significantly reduce the time and computing resources needed for fair VCS.

4.3.1 Repeated Rounding of Fractional Solutions (RRFS):

Here, we propose a repeated rounding heuristic to approximately solve the joint optimization in eq. (11). We first ignore the integrality constraint ($X_{t,s} \in \{0, 1\}$ in eq. (11)), and replace it with a general non-negativity constraint ($X_{t,s} \geq 0$) to get a fractional solution for decision variable X . Note that even though finding an integer solution is NP-hard, finding a fractional solution is polynomial time solvable ($O(l^3)$, where l is number of slots) [19]. We, then, find the maximum element from the fractional solution (let it be $X_{t,s}$), schedule talk t in slot s , and replace all other elements in the same row as t and column as s with 0. Then, we move to the next maximum from the remaining elements and repeat the same process till every other element become zero or all talks are scheduled. If some talks remain unscheduled when there is no non-zero element, we filter the unscheduled talks and slots, and repeat the

same process of finding a fractional solution and rounding. This method is detailed in algorithm 1 in the appendix, and has the worst case time complexity $O(nl(l^2 + mn))$ when $l \leq 2^{n^2}$.

4.3.2 Participant Clustering (PC): In big conferences, even though the number of talks and slots stay limited or may not grow too much, the number of participants could become very high which could lead to high memory complexity even to get a fractional solution. Thus, for such cases, we propose to group similar participants into clusters as a pre-processing strategy. We concatenate the interest and availability scores of a participant to create a vector, and call it as the participant’s profile vector; i.e., participant p ’s profile is $[V_p(t) \forall t \in \mathcal{T} : A_p(s) \forall s \in \mathcal{S}]$. We, then, apply k -means clustering to group similar participants, and use the cluster centroids as the representative participant profiles in the joint optimization framework while adding multiplicative weights—same as the size of the corresponding clusters—only in the efficiency and speaker fairness terms of the objective (as these two depend on the true audience participation values).

5 EXPERIMENTAL EVALUATION

5.1 Experimental Setup

5.1.1 Datasets: We test FairConf on a combination of multiple real-world datasets and a synthetic dataset.

Real-world Datasets: We consider three computer science conferences: Workshop on Responsible Recommendation (FATREC), ACM conference on Recommender Systems (RecSys), and International Conference on Machine Learning (ICML). Note that these conferences fall in small, medium and large conference categories respectively, and they help us to evaluate not only FairConf, but also the proposed scaling up approaches on medium and large conferences. While we gather the data on participant availabilities and timezones from publicly available sources (released by the organizers of 2020 conference), true participant interests are not available. Thus, we consider the list of talks (published papers) from 2017 edition of the conferences as the talk set of the 2020 edition, and then rely on randomized signals from the total number of citations or views of the papers to sample participant interest scores. Next, we describe each dataset in more details.

i. RECSYS: We collect the participant data (timezone data of 1112 participants) released in the welcome note of 2020 edition of the conference. We consider each participant to be completely available only during their working time (i.e., $\forall p, A_p(s) = 1$ if s is in between 9AM to 5PM in local timezone) of a day, and otherwise not available at all. We gather the published papers (26 papers) in the 2017 edition (<https://recsys.acm.org/recsys17/>) along with their citation counts which we use as a proxy for overall participant interest for a talk (i.e., a paper). For each participant-talk pair, we sample the interest scores from the a Bernoulli distribution $V_p(t) \sim \text{Bernoulli}(\text{prob} = \frac{\#cites(t)}{\max_{t' \in \mathcal{T}} \#cites(t')})$, $\forall p, t$. We consider 48 half hour slots over a 24 hour period (starting from 00hours in UTC), and try to schedule the 26 talks.

ii. FATREC: Here, we consider the workshop on responsible recommendation which is organized in conjunction with the RecSys conference. We collect the accepted papers (11 papers) from 2017 edition of the workshop (<https://piret.gitlab.io/fatrec/>) along

with their citation counts. We use the citation counts to sample the participants’ interest scores just as we do in case of RECSYS, and separately also from a normal distribution $V_p(t) \sim \text{Normal}(\text{mean} = \frac{\#cites(t)}{\max_{t' \in \mathcal{T}} \#cites(t')}, \text{std} = \frac{\text{mean}}{4})$, $\forall p, t$; however due to space constraints we present results only on the dataset derived from Normal distribution. For availability scores, we downsample 40 participants from the RECSYS participants set. We consider 96 fifteen-minute-slots (slot-size is same as in the 2017 schedule) over a 24 hour period (starting from 00hours in UTC), and try to schedule the 11 talks.

iii. ICML: We use the publicly available survey responses [8] from the participants of the International Conference on Machine Learning for the participant availability scores. We keep the number of participants same as the number of respondents in the survey, which is 2722. We, then, gather all papers from the 2017 edition of the conference (<https://icml.cc/Conferences/2017>) which are listed in ACM digital library [1] along with the number of times each paper is downloaded. Similar to RECSYS, for each participant-talk pair, we sample the interest score from a Bernoulli distribution. For the 209 listed papers, we consider 240 half hour slots over a period of 5 days (starting from 00hours in UTC), and schedule the talks.

Synthetic Dataset: We synthesize a dataset typically mimicking a small conference, and use it to understand and illustrate the dynamics of different methods. We take number of participants ($|\mathcal{P}| = 10$), number of talks ($|\mathcal{T}| = 10$), number of slots ($|\mathcal{S}| = 10$), and generate the synthetic dataset where the slots represent non-overlapping equal-sized time intervals available for scheduling. The interest scores and availability scores are then sampled from a uniform random distribution in $[0, 1]$; i.e., $V_p(t) \sim \text{Uniform}([0, 1])$ and $A_p(s) \sim \text{Uniform}([0, 1])$, $\forall p, t, s$.

5.1.2 Baselines: We use the following baselines and empirically compare them with our approach FairConf from §4.2.

A. Efficiency Maximization (EM): Following the trend in prior works on optimal meeting scheduling [5, 6, 13, 24, 27], here, we just optimize the schedule for efficiency; i.e., Γ^{EM} or $\text{argmax}_{\Gamma} \text{TEP}(\Gamma)$ without any fairness consideration.

B. Participant Fairness Maximization (PFair): Here, we just optimize for participant fairness; i.e., minimize participant unfairness $[\text{argmin}_{\Gamma} \Psi^{\text{P}}(\Gamma)]$ as defined in eq. (7).

C. Speaker Fairness Maximization (SFair): Here, we just optimize for speaker fairness; i.e., minimize speaker unfairness $[\text{argmin}_{\Gamma} \Psi^{\text{S}}(\Gamma)]$ as defined in eq. (9).

D. Interest-Availability Matching (IAM): Here, we sort the talks in descending order of the overall interest scores received by them, i.e., $\sum_{p \in \mathcal{P}} V_p(t)$, and the slots in descending order of the overall availability scores received by them, i.e., $\sum_{p \in \mathcal{P}} A_p(s)$. Now, we assign the talk with the highest overall interest score to the slot with the highest overall availability score, the talk with the second highest overall interest score to the slot with the second highest overall availability score, and so on (with random tie-breaks). IAM is one of the naive alternatives when scheduling is done manually (as natural objectives like EM usually need computing resources). It is also worth noting that, in the usual physical conference settings (i.e., all participants have identical ease of availability over all available slots $A_p(s) = A(s)$, $\forall s \in \mathcal{S}, p \in \mathcal{P}$), IAM yields a conference schedule which maximizes efficiency. It is a special case of lemma 2 (refer to case (a) of lemma 2 for the proof).

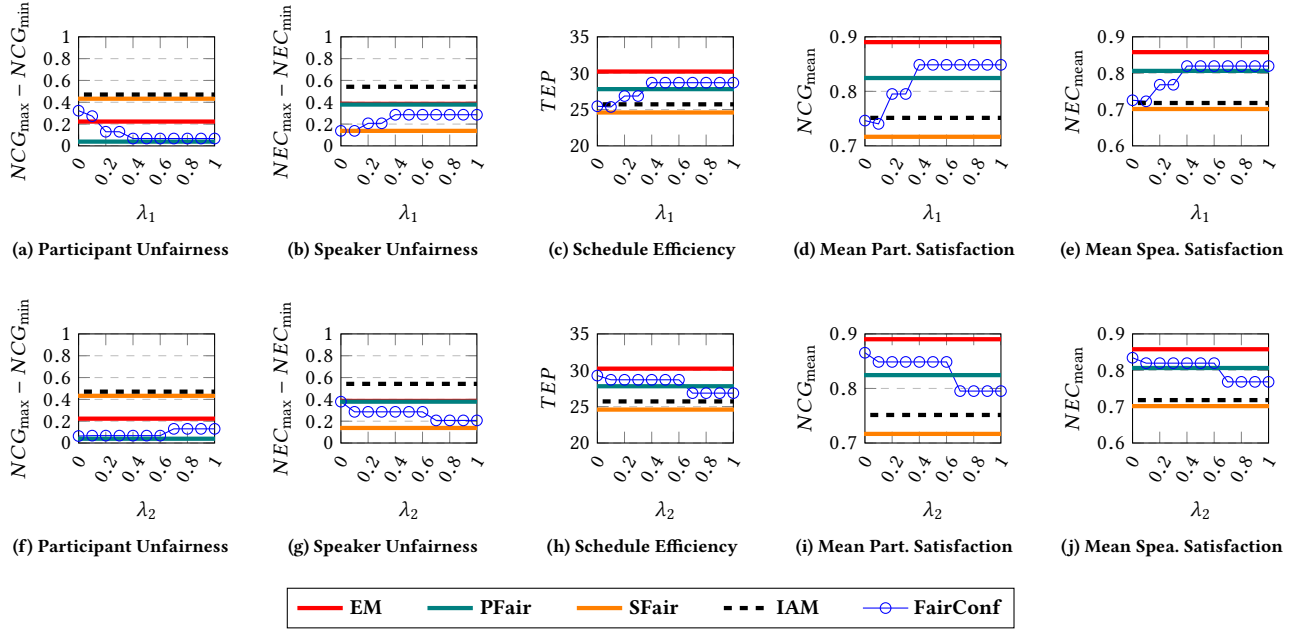


Figure 1: Results on synthetic dataset. For the plots in the first row, λ_2 is fixed at 0.5, and λ_1 is varied. For the plots in second row, λ_1 is fixed at 0.5, and λ_2 is varied.

LEMMA 2. *IAM maximizes efficiency, if the participants are identical either in terms of their interests in the talks or in terms of their ease of availability over the available slots, or both.*

5.1.3 Evaluation Metrics: Apart from the fairness metrics ($\Psi^P(\Gamma) = NCG_{\max} - NCG_{\min}$ as in definition 1, $\Psi^S(\Gamma) = NEC_{\max} - NEC_{\min}$ as in definition 3), we also measure the mean satisfaction of participants and speakers (NCG_{mean} and NEC_{mean}), and **efficiency** (TEP as in §3.3) as indicators of **efficiency**. Even though our unfairness metrics ($\Psi^P(\Gamma)$, $\Psi^S(\Gamma)$) are based on max-min differences for simplicity in modeling, they are quite vulnerable to participants or speakers with niche profiles especially in big conferences. Thus, we also use gini index [14] to measure the overall inequality in individual participant and speaker satisfactions (NCG_{gini}) as a measure of overall unfairness in bigger datasets (RECSYS and ICML). We use cvxpy (<https://www.cvxpy.org/>) paired with Gurobi (<https://www.gurobi.com/>) solver for the optimization. System details are: Debian GNU/Linux 10 on AMD64 architecture, Python 2.7.16, cvxpy 1.0.21, gurobipy 9.1.1, numpy 1.16.2, scikit-learn 0.20.3.

5.2 Experimental Results

5.2.1 Results on the Synthetic Dataset: We plot the results for synthetic dataset in fig. 1. Note that, unlike FairConf, the baseline approaches do not have hyperparameters λ_1, λ_2 ; thus, baseline results are just horizontal straight lines while FairConf’s results vary with hyperparameter settings.

Baseline Results: EM achieves the highest expected participation TEP (by definition it should), the highest mean participant satisfaction and mean speaker satisfaction (refer figs. 1c to 1e) while performing poorly on participant and speaker fairness (figs. 1a

and 1b). On the other hand, the naive IAM performs poorly in all the metrics. As PFair optimizes only for participant fairness, it has the highest participant fairness (least unfairness in fig. 1a) while losing in all the other metrics. Similarly, SFair performs the best in speaker fairness (least unfairness in fig. 1b) while losing in all other metrics as SFair optimizes only for speaker fairness.

FairConf Results: Note that, for the plots in first row (fig. 1a to fig. 1e), we fix $\lambda_2 = 0.5$ and vary λ_1 from 0 to 1; for the plots in second row (fig. 1f to fig. 1j), we fix $\lambda_1 = 0.5$ and vary λ_2 from 0 to 1. The general trends observed in FairConf’s results are: with increase in the weight for participant fairness (λ_1), FairConf achieves better participant fairness (fig. 1a) but worse speaker fairness (fig. 1b); with increase in the weight for speaker fairness (λ_2), FairConf achieves better speaker fairness (fig. 1g) but worse participant fairness (increase in participant unfairness in fig. 1f). We find that FairConf with the setting of $\lambda_1 = \lambda_2 = 0.5$ gives a balanced performance across all the metrics; it performs good in both participant fairness (very small unfairness in fig. 1a– close to PFair) and speaker fairness (very small unfairness in fig. 1b– slightly higher than SFair) while causing only marginal losses in mean participant satisfaction (fig. 1d), mean speaker satisfaction (fig. 1e), and the efficiency (fig. 1c). Note that, here, we do not imply that $\lambda_1 = \lambda_2 = 0.5$ will always give a balanced performance from FairConf; instead such a hyperparameter setting will be dataset-specific, and one needs to find it out through exploration. Moreover, a conference organizer may not always want a fully balanced schedule; she may even set the hyperparameters as per her relative priorities towards participants and speakers.

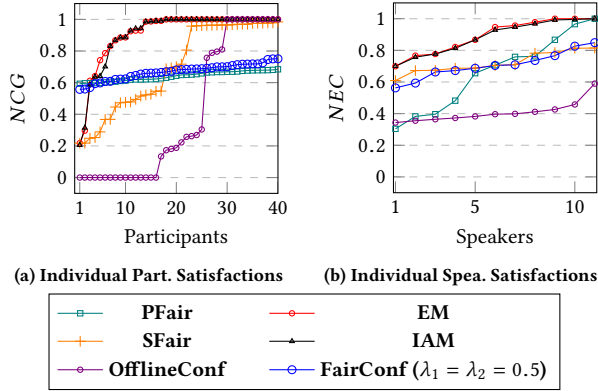


Figure 2: Individual participant and speaker satisfactions (sorted in increasing order) in FATREC dataset.

Methods	$NCG_{\max} - NCG_{\min}$	NCG_{mean}	$NEC_{\max} - NEC_{\min}$	NEC_{mean}
PFair	0.09	0.64	0.69	0.66
SFair	0.76	0.70	0.20	0.72
EM	0.78	0.91	0.30	0.89
IAM	0.79	0.90	0.30	0.88
OfflineConf	1	0.40	0.24	0.40
FairConf ($\lambda_1 = \lambda_2 = 0.5$)	0.19	0.66	0.28	0.70

Table 1: FATREC results (rounded upto two decimal points).

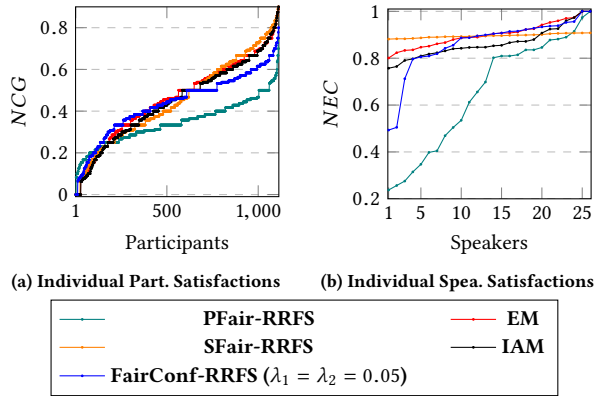


Figure 3: Individual participant and speaker satisfactions (sorted in increasing order) in RECSYS dataset.

Methods	NCG_{gini}	NCG_{mean}	NEC_{gini}	NEC_{mean}
PFair-RRFS	0.17	0.34	0.21	0.65
SFair-RRFS	0.26	0.45	0.01	0.89
EM	0.23	0.46	0.03	0.9
IAM	0.25	0.44	0.04	0.86
FairConf-RRFS ($\lambda_1 = 0.05, \lambda_2 = 0$)	0.18	0.43	0.07	0.85
FairConf-RRFS ($\lambda_1 = \lambda_2 = 0.05$)	0.19	0.43	0.06	0.87
FairConf-RRFS ($\lambda_1 = 0, \lambda_2 = 0.05$)	0.22	0.45	0.03	0.9

Table 2: RECSYS results (rounded upto two decimal places).

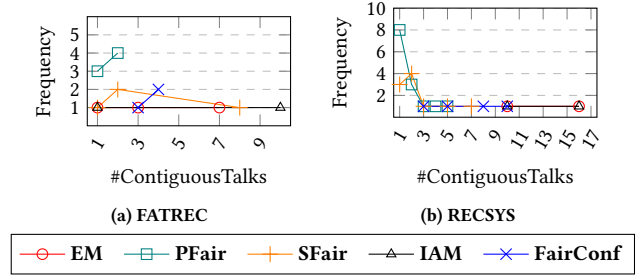


Figure 4: Schedule Contiguity. (FATREC: FairConf with $\lambda_1 = \lambda_2 = 0.5$); RECSYS: FairConf uses RRFS with $\lambda_1 = \lambda_2 = 0.05$, PFair and SFair use RRFS)

5.2.2 Results on FATREC Dataset: In fig. 2a and fig. 2b, we plot the individual participant satisfactions and speaker satisfactions—both sorted in increasing order—in FATREC dataset. We list relevant metric values in table 1. Along with the baselines and FairConf, we also evaluate the real FATREC-2017 workshop schedule (referred to as OfflineConf).

Baseline Results: While EM and IAM are able to ensure fairness on speaker-side (low inequality in NECs: table 1), they cause huge participant unfairness ($NCG_{\max} - NCG_{\min}$ in table 1). This is because, the number of talks (11) is very few in comparison to a total of 96 available slots, and there are enough number of slots favorable to crowds from either European or American timezones—covering majority of participants; so both EM and IAM are able to achieve high satisfaction for all speakers by just scheduling their talks in the slots favorable to either of the majority participant groups while undermining the minority participant group from non-European and non-American timezones. Similarly, SFair also undermines the minority participant group (fig. 2a). On the other hand, PFair significantly flattens the participant-side curve (fig. 2a) thereby being the most fair for participants; however it comes at a price—huge unfairness on the speaker-side (fig. 2b, table 1).

OfflineConf and FairConf Results: As FATREC-2017 was held in-person in Como, Italy, its schedule was local timezone-specific. If the same schedule were to be used in case of an online version, it would favour the participants mostly from nearby European timezones while severely undermining the participants from distant timezones (refer OfflineConf in fig. 2a). Such timezone-specific schedule also leaves no chance for the talks to be scheduled in slots with optimal availability of other majority participant groups, which then leads to highly suboptimal and unfair schedule for the speakers too (refer OfflineConf in table 1). While the baselines and timezone-specific schedule are proving to be less suitable for online conferences, FairConf, on the other hand, with $\lambda_1 = \lambda_2 = 0.5$ setting strikes a good balance by significantly reducing the max-min gap for the participants—thereby improving inclusivity— while still maintaining good speaker satisfaction and fairness similar to EM and SFair (table 1).

5.2.3 Results on RECSYS Dataset: While the number of talks in RECSYS is small, it has a very high number of participants. Thus, we use the proposed scalable approach (RRFS as in §4.3.1). Note

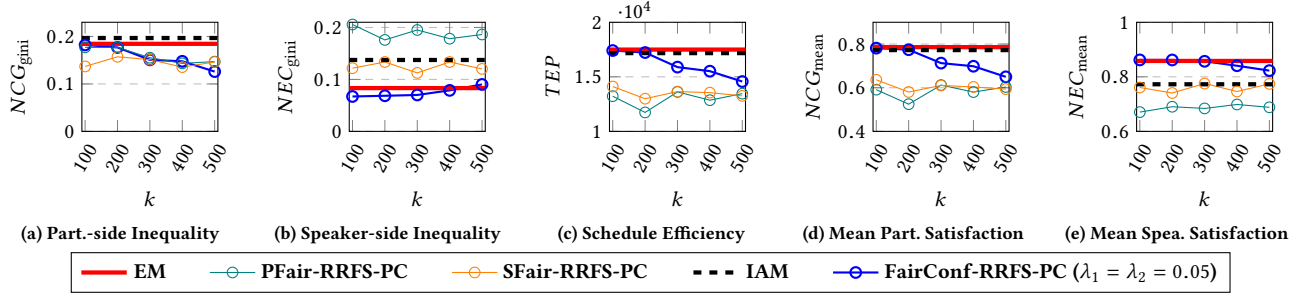


Figure 5: Results on ICML dataset.

that, due to the hardness of PFair and SFair, we compute them also through repeated rounding of fractional solutions. Similar to FATREC, here, we plot the individual participant and speaker satisfactions in fig. 3, and metric values in table 2. Both EM and IAM result in schedules with high inequality for the participants (fig. 3a), i.e., high participant unfairness (table 2). While PFair reduces the inequality on the participant-side, it increases the speaker-side inequality (fig. 3b); SFair behaves completely the opposite way. In this dataset, we find a balanced performance from FairConf with $\lambda_1 = \lambda_2 = 0.05$, i.e., reduction in max-min gaps on participants and speaker sides (also low gini index) without much degradation in overall satisfactions (table 2). This also serves as empirical evidence for the efficacies of scalable RRFS approach for big conferences.

5.2.4 Schedule Contiguity Analysis: In physical conferences, the talks are clubbed together in smaller numbers and scheduled in a contiguous manner. Such a strict contiguity need not be maintained in the virtual setting; however, if the talks are scheduled in a very segregated manner, participants might lose interest. Thus, we plot the frequencies of differently clubbed contiguous talks in fig. 4. We find that PFair results in a very segregated (high number of singular talks) schedule as it tries to fairly satisfy participants from all timezones. However, both EM and IAM give schedules where most of the talks are clustered around time slots favorable to majority of participants. SFair, however, leads to schedules with a mix of large number of contiguous talks and a few number of singular talks. FairConf, on the other hand, cares about efficiency and fairness simultaneously, thus clubs small number of talks and schedules them in the slots which are often at the intersection of availability intervals of participants from different timezones.

5.2.5 Results on ICML Dataset: As ICML has a very large number of participants, we first cluster the participants as proposed in §4.3.2, and then apply RRFS (as in §4.3.1) for FairConf joint optimization. In fig. 5, we plot the results on ICML dataset. Similar to previous datasets, here also, we observe FairConf to be performing in a more balanced manner than the baselines. We capture the changes in performance by varying k (number of participant clusters). While we observe no specific trend in the performances of the baselines by varying k , there is a clear trend in FairConf’s performance; with other settings fixed, an increase in k results in an increase participant fairness (decreasing inequality in fig. 5a), and subsequently a decrease in speaker fairness (increasing inequality in fig. 5b). This is because higher k leads to smaller participant

clusters where the centroids are able to better represent all the participants leading to better participant fairness which shifts some talks to the favorable slots of previously less-represented participants thereby decreasing speaker fairness. Note that an increase in participant fairness, here, also comes with losses in efficiency, and individual participant and speaker satisfactions (figs. 5c to 5e).

5.2.6 Priority Scheduling and Repetitions: Sometimes the conference organizers might have varying priorities towards the talks (e.g. short vs. long, main vs. special track). In such cases, multiple instances of FairConf in asynchronous setup can be used to schedule different groups of talks based on their priority levels. Top-priority talks can be scheduled first by allowing their mapping to any possible slot, and subsequently scheduling lower priority talks in the remaining slots. This allows us to bring intra-(priority)-group fairness for speakers. Moreover such an asynchronous setup can also allow for repetitions of top-priority talks once a round of scheduling is done. We test such priority scheduling and repetitions on RECSYS dataset. We group the talks into three priority levels (top, medium and low) of equal size based on their overall interest scores, and then asynchronously schedule them. The detailed results are given in table 4 in the appendix, we here highlight the main findings from the results. We observe that in comparison to a full-scale FairConf, priority scheduling achieves better participant satisfaction since the top level talks get slots with more participant availability as they do not have to compete with low priority talks any more, and it also gives better intra-group fairness for speakers. We also observe that introducing repetitions of talks, participant satisfaction slowly increases and the schedules get more fair for participants. Priority-based repetitions also increase the total expected crowd at the talks thereby increasing speaker satisfaction. In fact, with prioritized repetitions of talks, the speakers even have access to more audience (NEC more than 1) than what they would have received in their single best slot. Even though the gaps between two assigned slots for a talk can be upto 12 hours, the speakers often have incentive of getting more total audience through repetitions. Note that with repetitions, the inequality on speaker-side might increase since the set of good slots which can ensure speaker fairness would already have been allocated in the first schedule, and the remaining slots, to be allocated for repetitions, may have varying effects on speaker satisfactions.

6 DISCUSSION

In this work, we modeled a very timely and important problem of virtual conference scheduling with efficiency and fairness concerns. Apart from the formal definitions, we brought out fundamental tensions among participant fairness, speaker fairness, and efficiency. We experimentally showed that the proposed joint optimization framework, FairConf, can find balanced conference schedules and also allow the organizers to generate schedules as per their relative priorities towards participants and speakers. We note some of the limitations of the present work and possible future directions in the appendix.

Reproducibility: Our datasets, codes, and results can be accessed at <https://github.com/gourabkumarpatro/FairConf>.

Acknowledgements: G. K Patro acknowledges the support by TCS Research Fellowship. This research was supported in part by an European Research Council (ERC) Advanced Grant for the project “Foundations for Fair Social Computing” (grant agreement no. 789373), and an European Research Council (ERC) Marie Skłodowska-Curie grant for the project “NoBIAS - Artificial Intelligence without Bias” (grant agreement no. 806030), both funded under the EU’s Horizon 2020.

REFERENCES

- [1] 2017. *ICML'17: Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia). JMLR.org. <https://dl.acm.org/doi/proceedings/10.5555/3305381>
- [2] 2020. *Conference management: 5 benefits of hosting a virtual conference*. <https://oxfordabstracts.com/blog/2020-03-30-5-benefits-of-hosting-a-virtual-conference/>
- [3] Ercan Atagün and Serdar Birol. 2020. Effectiveness of Genetic Algorithm in the Solution of Multidisciplinary Conference Scheduling Problem. In *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*. Springer, 220–230.
- [4] Richard Baum, Dimitris Bertsimas, and Nathan Kallus. 2014. Scheduling, revenue management, and fairness in an academic-hospital radiology division. *Academic radiology* 21, 10 (2014), 1322–1330.
- [5] Peter George Capek, William Grey, Paul Andrew Moskowitz, Clifford A Pickover, and Dailun Shi. 2008. Event scheduling with optimization. US Patent 7,343,312.
- [6] Andy Chun, Hon Wai, and Rebecca YM Wong. 2003. Optimizing agent-based meeting scheduling through preference estimation. *Engineering Applications of Artificial Intelligence* 16, 7-8 (2003), 727–743.
- [7] Rubens Correia, Anand Subramanian, Teobaldo Bulhões, and Puca Huachi V Penna. 2021. Scheduling the Brazilian OR Conference. *Journal of the Operational Research Society* (2021).
- [8] Hal DaumÄI III and Aarti Singh. 2020. *ICML 2020 Virtualization Survey Responses - Results*. <https://icml.cc/Conferences/2020/VirtualizationSurvey>
- [9] Alastair F Donaldson. 2020. A Report on the first virtual PLDI conference. *arXiv preprint arXiv:2007.11686* (2020).
- [10] Wen Fang, Qingqing Zhang, Qingwen Liu, Jun Wu, and Pengfei Xia. 2018. Fair scheduling in resonant beam charging for IoT devices. *IEEE Internet of Things Journal* 6, 1 (2018), 641–653.
- [11] Eugene C Freuder, Marius Minca, and Richard J Wallace. 2001. Privacy/efficiency tradeoffs in distributed meeting scheduling by constraint-based agents. In *Proc. IJCAI DCR*. Citeseer, 63–72.
- [12] Michael R Garey and David S Johnson. 1979. *Computers and intractability*. Vol. 174. freeman San Francisco.
- [13] Leonardo Garrido and Katia Sycara. 1996. Multi-agent meeting scheduling: Preliminary experimental results. In *Proceedings of the Second International Conference on Multiagent Systems*, 95–102.
- [14] Corrado Gini. 1912. Variabilità e mutabilità. *Reprinted in Memorie di metodologica statistica* (Ed. Pizetti E (1912).
- [15] GH Hardy, JE Littlewood, and G Pólya. 1967. *Inequalities*. Cambridge Mathematical Library Series.
- [16] Michael Harris, John Carrino, and Eric Wong. 2015. Fair scheduling for mixed-query loads. US Patent 9,092,482.
- [17] Michael Isard, Vijayan Prabhakaran, Jon Currey, Udi Wieder, Kunal Talwar, and Andrew Goldberg. 2009. Quincy: fair scheduling for distributed computing clusters. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, 261–276.
- [18] Judy Kay and Piers Lauder. 1988. A fair share scheduler. *Commun. ACM* 31, 1 (1988), 44–55.
- [19] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.
- [20] Hooyeon Lee and Ashish Goel. 2016. Probabilistic Matrix Inspection and Group Scheduling. In *IJCAI*, 322–328.
- [21] Tong Li, Dan Baumberger, and Scott Hahn. 2009. Efficient and scalable multi-processor fair scheduling using distributed weighted round-robin. *ACM Sigplan Notices* 44, 4 (2009), 65–74.
- [22] Jean-Pierre Lozi, Baptiste Lepers, Justin Funston, Fabien Gaud, Vivien Quéma, and Alexandra Fedorova. 2016. The Linux scheduler: a decade of wasted cores. In *Proceedings of the Eleventh European Conference on Computer Systems*, 1–16.
- [23] Kshiteej Mahajan, Arjun Singhvi, Arjun Balasubramanian, Varun Batra, Surya Teja Chavali, Shivaram Venkataraman, Aditya Akella, Amar Phanishayee, and Shuchi Chawla. 2019. Themis: Fair and efficient gpu cluster scheduling for machine learning workloads. *arXiv preprint arXiv:1907.01484* (2019).
- [24] Rajiv Maheswaran, Milind Tambe, Emma Bowring, Jonathan Pearce, and Pradeep Varakantham. 2004. Taking DCOP to the real world: Efficient complete solutions for distributed event scheduling. (2004).
- [25] Rajiv T Maheswaran, Jonathan P Pearce, Emma Bowring, Pradeep Varakantham, and Milind Tambe. 2006. Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. *Autonomous Agents and Multi-Agent Systems* 13, 1 (2006), 27–60.
- [26] Chris Misa, Dennis Guse, Oliver Hohlfeld, Ramakrishnan Durairajan, Anna Sperotto, Alberto Dainotti, and Reza Rejaie. 2020. Lessons learned organizing the PAM 2020 virtual conference. *ACM SIGCOMM Computer Communication Review* 50, 3 (2020), 46–54.
- [27] José A Pino and Hugo A Mora. 1998. Scheduling meetings using participants’ preferences. *Information Technology & People* (1998).
- [28] Jan Pisanski and Tomaz Pisanski. 2019. The use of collaboration distance in scheduling conference talks. *Informatica* 43, 4 (2019).
- [29] Michael Saliba. 2020. Getting to grips with online conferences. *Nature Energy* 5, 7 (2020), 488–490.
- [30] Amartya Sen, Bernard Arthur Owen Williams, Bernard Williams, et al. 1982. *Utilitarianism and beyond*. Cambridge University Press.
- [31] Sandip Sen and Edmund H Durfee. 1998. A formal study of distributed meeting scheduling. *Group Decision and Negotiation* 7, 3 (1998), 265–289.
- [32] Thomas Stidsen, David Pisinger, and Daniele Vigo. 2018. Scheduling EURO-k conferences. *European Journal of Operational Research* 270, 3 (2018), 1138–1147.
- [33] Nitin Vaidya, Anurag Dugar, Seema Gupta, and Paramvir Bahl. 2005. Distributed fair scheduling in a wireless LAN. *IEEE Transactions on Mobile Computing* 4, 6 (2005), 616–629.
- [34] Richard J Wallace and Eugene C Freuder. 2005. Constraint-based reasoning and privacy/efficiency tradeoffs in multi-agent problem solving. *Artificial Intelligence* 161, 1-2 (2005), 209–227.

A APPENDIX

Proof of lemma 1: Let's use matrix X of dimensions $|\mathcal{T}| \times |\mathcal{S}|$ to represent a conference schedule; i.e., the element $X_{t,s} \in \{0, 1\}$ is a binary indicator variable for talk $t \in \mathcal{T}$ being scheduled in slot $s \in \mathcal{S}$. We can now rewrite the efficiency maximization problem ($\arg\max_{\Gamma} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} V_p(t) \times A_p(\Gamma(t))$) as below.

$$\begin{aligned} & \arg\max_X \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} V_p(t) \cdot A_p(s) \cdot X_{t,s} \\ & \equiv \arg\max_X \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}} V_p(t) \cdot A_p(s) \cdot X_{t,s} \\ & \equiv \arg\max_X \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}} X_{t,s} \cdot \sum_{p \in \mathcal{P}} V_p(t) \cdot A_p(s) \\ & \equiv \arg\min_X \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}} X_{t,s} \cdot \left(|\mathcal{P}| - \sum_{p \in \mathcal{P}} V_p(t) \cdot A_p(s) \right) \\ & \equiv \arg\min_X \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{S}} X_{t,s} \cdot c_{t,s} \end{aligned}$$

As $|\mathcal{T}| \leq |\mathcal{S}|$, we can introduce $(|\mathcal{S}| - |\mathcal{T}|)$ dummy talks $(\mathcal{T}_d = \{t'_1, \dots, t'_{|\mathcal{S}|-|\mathcal{T}|}\})$ with costs: $c_{t,s} = |\mathcal{P}|$, $\forall t, s \in \mathcal{T}_d, \mathcal{S}$. As scheduling the dummy talks has a constant cost attached, we can now rewrite the above transformed problem as below.

$$\arg\min_X \sum_{t \in \mathcal{T} \cup \mathcal{T}_d} \sum_{s \in \mathcal{S}} X_{t,s} \cdot c_{t,s}$$

This is a minimum cost bipartite matching problem which can be solved in polynomial time using the Hungarian algorithm [19].

Proofsketch of theorem 3.4.1: The problem is clearly in NP. Given a schedule Γ , participant unfairness can be calculated and verified in $O(m \cdot (n \log n + l \log l))$ or $O(ml \log l)$ time (as $n \leq l$). To prove NP-hardness, we reduce the **number partitioning problem** (a well-known NP-complete problem [12]) to our participant fairness problem. An arbitrary instance of number partition problem has a multiset \mathcal{G} of integers, and the task is to decide whether \mathcal{G} can be partitioned into two disjoint subsets \mathcal{G}_1 and \mathcal{G}_2 such that the sum of numbers in \mathcal{G}_1 equals the sum of numbers in \mathcal{G}_2 . We provide a polynomial time reduction to a participant fairness instance. Let $n = |\mathcal{G}|$, $\mathcal{G} = \{g_1, g_2, \dots, g_n\}$, and $\text{sum}(\mathcal{G}) = \sum_{i=1}^n g_i$. Now let the set of participants $\mathcal{P} = \{p_1, p_2\}$, and the set of talks $\mathcal{T} = \{t_1, \dots, t_n\}$. We set the interest scores of participants as $V_{p_1}(t_i) = V_{p_2}(t_i) = \frac{g_i}{\text{sum}(\mathcal{G})} \forall 1 \leq i \leq n$. Let the set of slots be $\mathcal{S} = \{s_1, \dots, s_n, s_{n+1}, \dots, s_{2n}\}$, and the availability scores of the participants be like: $A_{p_1}(s_i)$ is 1 for $1 \leq i \leq n$ and 0 for $n+1 \leq i \leq 2n$; $A_{p_2}(s_i)$ is 0 for $1 \leq i \leq n$ and 1 for $n+1 \leq i \leq 2n$. Intuitively, scheduling a talk in the first half of the slots will bring no gain to p_2 while scheduling in second half of the slots will bring no gain to p_1 . In this setting, it can be easily found that $ICG(p_1) = ICG(p_2) = 1$. Given a polynomial time solution to participant fairness problem (definition 2), we can set $\epsilon = 0$, and check if such a conference schedule exists. Essentially, scheduling talks with $\epsilon = 0$ is identical to allocating the set of n talks into (i) half of the slots when p_1 is available, and (ii) the other half when p_2 is available while ensuring the cumulative gains of p_1 and p_2 are same (as $ICGs$ are already same). The answer on the existence/non-existence of such a schedule will also answer the existence/non-existence of number partitions with equal sums in polynomial time (as it is a polynomial-time reduction). Unless $P=NP$, such a polynomial time solution for

the participant fairness problem does not exist. Thus, the problem is NP-complete.

Proof of claim 1: We disprove the negation of the claim using a counter example as given in table 3a. Both participants have interest score of 1 for the talk. Now looking at the availability scores, while p_1 and p_2 have full ease of availability in s_1 and s_3 respectively, they both can make themselves available s_2 with 0.49 probability. If we consider a efficiency objective (participation maximization) here, we would end up scheduling the talk in either in s_1 or in s_3 ; if $\Gamma(t) = s_1$ or $\Gamma(t) = s_3$, then $TEP(\Gamma) = 1$; if $\Gamma(t) = s_2$, then $TEP(\Gamma) = 0.98$ which is less. However, maximizing participation will either end up with $[NCG(p_1) = 1, NCG(p_2) = 0]$ if $\Gamma(t) = s_1$ or $[NCG(p_1) = 0, NCG(p_2) = 1]$ if $\Gamma(t) = s_3$; As both of these results from efficiency optimization provide disparate satisfaction to the participants, they both are unfair. On the other hand, if we schedule the talk in s_2 ($\Gamma(t) = s_2$), then it becomes fair to both the participants as they will get similar satisfaction $[NCG(p_1) = 0.49, NCG(p_2) = 0.49]$ —they both get a chance to make themselves available in s_2 to attend the talk. Even though scheduling the talk in s_2 , ensures participant fairness, it has come at a loss in efficiency; i.e., $TEP(\Gamma)$ reduced from 1 to 0.98.

Proof of claim 2: We disprove the negation of the claim using a counter example as given in table 3b. Now, to maximize efficiency, we can just match talks in decreasing order of overall interest scores to slots in decreasing order of availability scores; i.e., $\Gamma^{EM}(t_1) = s_1$ and $\Gamma^{EM}(t_2) = s_3$, which will yield $TEP(\Gamma^{EM}) = 1.4$. The speaker satisfactions for the talks with this schedule will be: $NEC(t_1|\Gamma^{EM}) = 1$ (as $EC(t_1|\Gamma^{EM}) = 1$ and $IEC(t_1) = 1$), and $NEC(t_2|\Gamma^{EM}) = 0.8$ (as $EC(t_2|\Gamma^{EM}) = 0.4$ and $IEC(t_2) = 0.5$). Such disparity in speaker satisfactions can be attributed to speaker unfairness. In order to reduce speaker-side disparity, we can use a different schedule: $\Gamma(t_1) = s_3$ and $\Gamma(t_2) = s_2$; this yields speaker satisfactions $NEC(t_1|\Gamma) = 0.8$ and $NEC(t_2|\Gamma) = 0.75$ (as $EC(t_1|\Gamma) = 0.8$ and $EC(t_2|\Gamma) = 0.375$). This above schedule has in fact the lowest possible disparity in speaker satisfactions, i.e., the highest possible speaker fairness. Even though this schedule $\{(t_1, s_3), (t_2, s_2)\}$ is fairer to the speakers than the earlier $\{(t_1, s_1), (t_2, s_3)\}$, the gain in speaker fairness has come at a loss in efficiency; $TEP(\Gamma)$ reduced from 1.4 to 1.175.

Proof of claim 3: We disprove the negation of the claim using a counter example as given in table 3c. In this example, the schedule $\Gamma = \{(t_1, s_2), (t_2, s_3)\}$ achieves speaker fairness— $NEC(t_1|\Gamma) = NEC(t_2|\Gamma) = 0.5$ (as $EC(t_1|\Gamma) = 1$, $EC(t_2|\Gamma) = 0.7$, while $IEC(t_1) = 2$, $IEC(t_2) = 1.4$). However, Γ is unfair for the participants— $NCG(p_1|\Gamma) = \frac{1}{1.7} < \frac{0.7}{1.7} = NCG(p_2|\Gamma)$ (as $CG(p_1|\Gamma) = 1$, $CG(p_2|\Gamma) = 0.7$, while $ICG(p_1) = ICG(p_2) = 1.7$). On the other hand, schedule $\Gamma' = \{(t_1, s_1), (t_2, s_4)\}$ is fair for the participants— $NCG(p_1|\Gamma') = NCG(p_2|\Gamma') = \frac{1.14}{1.7}$, while being unfair for the speakers as $NEC(t_1|\Gamma') = 1 > 0.2 = NEC(t_2|\Gamma')$.

Proof of lemma 2: There are three cases where we need to prove that IAM maximizes efficiency; (a) if all participants have identical ease of availability over all available slots $A_p(s) = A(s)$, $\forall s \in \mathcal{S}, p \in \mathcal{P}$ (this case is similar to physical conference settings where all participants gather at the same place, thus, have identical ease of availability); (b) if all participants have identical interests over all talks $V_p(t) = V(t)$, $\forall t \in \mathcal{T}, p \in \mathcal{P}$; (c) if both (a) and (b) are true. We, first, reduce the EM objectives in the following cases, and observe

Table 3: Examples

(a) Example Problem 1

Participants	$V_p(t)$		$A_p(s)$		
	t	s_1	s_2	s_3	
p_1	1	1	0.49	0	
p_2	1	0	0.49	1	

(b) Example Problem 2

Participants	$V_p(t)$		$A_p(s)$		
	t_1	t_2	s_1	s_2	s_3
p	1	0.5	1	0.75	0.8

(c) Example Problem 3

Participants	$V_p(t)$		$A_p(s)$			
	t_1	t_2	s_1	s_2	s_3	s_4
p_1	1	0.7	1	1	0	0.2
p_2	1	0.7	1	0	1	0.2

that they take a particular form where IAM gives solution.

Case-(a): Given $A_p(s) = A(s)$, $\forall s \in \mathcal{S}, p \in \mathcal{P}$. From eq. (5):

$$\begin{aligned}
 \text{EM} &\equiv \arg\max_{\Gamma} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} V_p(t) \times A_p(\Gamma(t)) \\
 &\equiv \arg\max_{\Gamma} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} V_p(t) \times A(\Gamma(t)) \equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} V_p(t) \times A(\Gamma(t)) \\
 &\equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} A(\Gamma(t)) \left[\sum_{p \in \mathcal{P}} V_p(t) \right] \equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} A(\Gamma(t)) \times \mathcal{V}(t)
 \end{aligned}$$

(where overall interest score = $\mathcal{V}(t) = \sum_{p \in \mathcal{P}} V_p(t)$)

Case-(b): Given $V_p(t) = V(t)$, $\forall t \in \mathcal{T}, p \in \mathcal{P}$. From eq. (5):

$$\begin{aligned}
 \text{EM} &\equiv \arg\max_{\Gamma} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} V_p(t) \times A_p(\Gamma(t)) \\
 &\equiv \arg\max_{\Gamma} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} V(t) \times A_p(\Gamma(t)) \equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} V(t) \times A_p(\Gamma(t)) \\
 &\equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} V(t) \left[\sum_{p \in \mathcal{P}} A_p(\Gamma(t)) \right] \equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} V(t) \times \mathcal{A}(\Gamma(t))
 \end{aligned}$$

(where overall availability score = $\mathcal{A}(s) = \sum_{p \in \mathcal{P}} A_p(s)$)

Case-(c): Given $A_p(s) = A(s)$, $V_p(t) = V(t)$, $\forall s \in \mathcal{S}, \forall t \in \mathcal{T}, p \in \mathcal{P}$. From eq. (5):

$$\begin{aligned}
 \text{EM} &\equiv \arg\max_{\Gamma} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} V_p(t) \times A_p(\Gamma(t)) \\
 &\equiv \arg\max_{\Gamma} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} V(t) \times A(\Gamma(t)) \equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} \sum_{p \in \mathcal{P}} V(t) \times A(\Gamma(t)) \\
 &\equiv \arg\max_{\Gamma} \sum_{t \in \mathcal{T}} V(t) \times A(\Gamma(t))
 \end{aligned}$$

In all the three cases above, EM reduces to a form where the terms are independent of individual participant interests and availability while depending only on the overall interest levels (\mathcal{V} in case (a), and participant-independent V in cases (b) & (c)) and overall availability (\mathcal{A} in case (b), and participant-independent A in cases (a) & (c)). Thus, to maximize the reduced objectives in all the cases, top values of overall availability A or \mathcal{A} need to be matched with top values of overall interests \mathcal{V} or V (this also follows from the Rearrangement inequality [15])—making it identical to IAM.

Future Work: (i) We limited ourselves in modeling single-track conference scenario; however, larger conferences often have parallel sessions to accommodate a higher number of talks. The proposed modular framework can be reused with tweaked participant and speaker satisfaction metrics as participants would choose the best talk out of all parallel sessions. The interest score terms (in eqs. (1) and (3)) will be replaced with maximum over the set of parallel talks in each slot. (ii) We modeled speaker satisfaction using the expected crowd at her talk, however, the speaker's convenience in the assigned time slot could also play a role and can be modeled accordingly; (iii) Although we considered the participants and

Algorithm 1 *RRFS* ($\mathcal{P}, \mathcal{T}, \mathcal{S}, V, A, \lambda_1, \lambda_2$)

Input : Set of participants \mathcal{P} , set of talks \mathcal{T} , set of slots \mathcal{S} (such that $|\mathcal{T}| \leq |\mathcal{S}|$), the interest scores $V(p, t) \forall p \in \mathcal{P}, t \in \mathcal{T}$, the availability scores $A(p, s) \forall p \in \mathcal{P}, s \in \mathcal{S}$, weight for participant fairness objective λ_1 , and weight for speaker objective λ_2 .

Output: A fair conference schedule Γ .

- 1: $NSch \leftarrow \mathcal{T}$ ▷ Set of all talks which are not yet scheduled
- 2: $Nall \leftarrow \mathcal{S}$ ▷ Set of all slots which are not yet allotted
- 3: Let $RJO(\mathcal{P}, \mathcal{T}, \mathcal{S}, V, A, \lambda_1, \lambda_2)$ represent the relaxed joint optimization problem (i.e., $X_{t,s} \in \{0, 1\}$ in eq. (11) is replaced with $X_{t,s} \geq 0$).
- 4: **while** $NSch \neq \emptyset$ **do**
- 5: $X \leftarrow RJO(\mathcal{P}, \mathcal{T}, \mathcal{S}, V, A, \lambda_1, \lambda_2)$ ▷ A fractional solution
- 6: **while** X is not a zero-matrix **do** ▷ Rounding loop
- 7: $t, s = \arg\max_{t', s' \in \mathcal{T}, \mathcal{S}} X_{t', s'}$ ▷ Maximum element
- 8: Set $\Gamma(t) = s$
- 9: $NSch \leftarrow NSch \setminus \{t\}$
- 10: $Nall \leftarrow Nall \setminus \{s\}$
- 11: Set $X_{i,s} = 0 \forall i \in \mathcal{T}$, and $X_{t,j} = 0 \forall j \in \mathcal{S}$
- 12: **end while**
- 13: $\mathcal{T} \leftarrow NSch, \mathcal{S} \leftarrow Nall$
- 14: **end while**
- 15: Return Γ .

FairConf	NCG_{gini}	NCG_{mean}	NEC_{gini}			NEC_{mean}		
			\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3
1 Full	0.19	0.43	0.08	0.02	0.06	0.82	0.91	0.87
\mathcal{T}_{123}	0.19	0.44	0.03	0.01	0.05	0.91	0.87	0.81
\mathcal{T}_{1231}	0.17	0.51	0.13	0.01	0.05	1.24	0.87	0.81
\mathcal{T}_{12312}	0.13	0.56	0.13	0.06	0.05	1.24	1.17	0.81
\mathcal{T}_{123123}	0.10	0.58	0.13	0.06	0.07	1.24	1.17	0.96
2 Full	0.11	0.54	0.06	0.08	0.10	0.95	1.16	1.18
\mathcal{T}_{123121}	0.10	0.59	0.13	0.06	0.05	1.37	1.17	0.81

Table 4: RECSYS priority scheduling results (rounded upto two decimal places). \mathcal{T}_{1231} refers to FairConf scheduling of top, medium, low, and finally a repetition for top priority talks as discussed in §5.2.6. “1 Full” refers to scheduling all talks at once without any grouping. “2 Full” refers to 2 rounds of full scheduling where the first one is same as “1 Full” and the second one is trying to schedule all talks in the remaining slots as repetitions.

speakers to be separate agents in our model, a single agent could be both a participant and a speaker; thus for such agents with dual roles, the participant satisfaction measure needs to be suitably modified. (iv) We have limited ourselves to individual fairness for the speakers and participants, without considering their sensitive attributes. However, our joint optimization framework can accommodate additional constraints to include group fairness objectives.