European Journal of Operational Research 317 (2024) 487–499

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Discrete Optimization

A generic approach to conference scheduling with integer programming

Yaroslav Pylyavskyy [a,*], Peter Jacko [a,b], Ahmed Kheiri [a]

[a] Department of Management Science, Lancaster University, Lancaster LA1 4YX, UK
[b] Berry Consultants, 5 East Saint Helen Street, Abingdon OX14 5EG, UK

ARTICLE INFO

Keywords:
Scheduling
Integer programming
Combinatorial optimisation
Timetabling

ABSTRACT

Conferences are a key aspect of communicating knowledge, and their schedule plays a vital role in meeting the expectations of participants. Given that many conferences have different constraints and objectives, different mathematical models and heuristic methods have been designed to address rather specific requirements of the conferences being studied per se. We present a penalty system that allows organisers to set up scheduling preferences for tracks and submissions regarding sessions and rooms, and regarding the utilisation of rooms within sessions. In addition, we also consider hybrid and online conferences where submissions need to be scheduled in appropriate sessions based on timezone information. A generic scheduling tool is presented that schedules tracks into sessions and rooms, and submissions into sessions by minimising the penalties subject to certain hard constraints. Two integer programming models are presented: an exact model and an extended model. Both models were tested on five real instances and on two artificial instances which required the scheduling of several hundreds of time slots. The results showed that the exact model achieved optimal solutions for all instances except for one instance which resulted in 0.001% optimality gap, and the extended model handles more complex and additional constraints for some instances. Overall, this work demonstrates the suitability of the proposed generic approach to optimise schedules for in-person, hybrid, and online conferences.

1. Introduction

Conferences are events of great importance to scientific communities. They provide an opportunity for academics and researchers to present their research work and receive feedback from the community, and to learn from other presenters. In addition, participants benefit from networking opportunities, exchanging ideas, and potential future collaborations. Hence, a conference schedule brings a significant opportunity and challenge in offering the best possible experience to every participant. Conference organisers usually struggle to create, or even characterise, the best schedule due to the large number of preferences and constraints involved. Some of the constraints that make conference scheduling an arduous task are requests from presenters to present at a specific time, resolving presenters conflicts, handling capacity issues, to name a few. Additionally, COVID-19 pandemic has resulted in many conferences switching to online or hybrid mode, which introduces further complexity due to different timezones involved.

Due to the diverse conference terminology that has been used in the *conference scheduling problem* (CSP) literature, we clarify the conference terminology as used in this paper, which we believe may be applicable to many conferences. While various terms such as paper, presentation, talk, discussion, and panel are used in the literature, we use the term "submission" to refer to a formal event that requires scheduling at a conference. The term "track" is used to refer to a group of submissions with similar subject, whereas terms such as stream, subject area, and topic are used in the literature. We use the term "time slot" to refer to a fixed predefined amount of time available for presentation, and the term "session" is used to refer to a certain time period of the conference that consists of a number of time slots.

In this paper, we consider the CSP, which includes a set of tracks along with their corresponding submissions, a set of available sessions along with their corresponding time slots, and a set of available rooms. The objective is to achieve a schedule which is feasible to a number of (hard) constraints and minimises violations of a number of preferences (soft constraints) by assigning all tracks into sessions and rooms, and assigning all submissions into sessions. Based on the types of violations, a CSP can be approached from a Presenter-Based Perspective (PBP) or from an Attender-Based Perspective (ABP) (Thompson, 2002). A PBP approach aims to minimise violations associated with presenters preferences, such as a request to present on a specific day or at a specific time (Sampson, 2004). An ABP approach minimises violations regarding attendants preferences. Some examples are that all of the attendants wish to attend their favourite session, they do

* Corresponding author.
E-mail addresses: y.pylyavskyy1@lancaster.ac.uk (Y. Pylyavskyy), p.jacko@lancaster.ac.uk (P. Jacko), a.kheiri@lancaster.ac.uk (A. Kheiri).

https://doi.org/10.1016/j.ejor.2024.04.001
Received 1 June 2023; Accepted 1 April 2024
Available online 6 April 2024

not want to miss a session due to space shortage, and they do not want to choose between two sessions of their interest due to sessions being scheduled concurrently (Zulkipli et al., 2013). Some studies have adopted a mixed approach by considering both presenters and attendants preferences (Nicholls, 2007; Vangerven et al., 2018).

The CSP was introduced by Eglese and Rand (1987) and was proved to be $\mathcal{NP}$-hard by Quesnelle and Steffy (2015) and Vangerven et al. (2018). Even though the problem was introduced several decades ago, it has not received much attention from researchers compared to related problems, such as Class and Exam Scheduling (Sampson, 2004). To the best of our knowledge, there are only 16 published studies tackling the CSP (see Section 2). However, many conferences have different scheduling requirements, objectives, and constraints. As a result, a method that works well for a conference could be unsuitable for another conference. Thus, we believe that a generic framework for conference scheduling is needed. To this end, in this study we present a generic approach by considering both PBP and ABP to generate schedules for conferences in a fully automated manner.

Our work mainly differs from the existing literature in the sense that we generate both low-level and high-level conference schedules. That is, we consider preferences and constraints associated with both tracks and submissions to create a complete schedule of a conference. In addition, our approach is suitable for hybrid and online conferences as we take timezones into account to avoid scheduling submissions into unsuitable times. To the best of our knowledge, this is the first paper to consider timezone differences in conference scheduling. Moreover, we present a generic approach for conference scheduling by applying the weighted sum method as described in Ehrgott (2005) to create an objective function to optimise. The method is applied to our penalty system designed to accommodate scheduling preferences which are weighted according to their relative importance. An easy-to-use and configurable spreadsheet template is used to meet the demands of different conferences. The template offers a single data format that is easy to adjust in order to fit different conference data. We acknowledge that our template is not suitable for all conferences, as some conferences may have very specific structures, but our aim is to accommodate as many as possible. Furthermore, in contrast to most previous studies which assume that all submissions require the same amount of time for scheduling (one time slot), we allow for submissions to have a different amount of required time slots, such as keynote talks.

In Section 2, we present related work on conference scheduling problems, followed by Section 3 which describes the conference scheduling problem as considered in this work. Then, in Section 4, we present our exact model as a binary integer program with linear objective. Computational results are presented in Section 5, followed by Section 6 which presents the extended model with additional constraints along with computational results. Next, in Section 7, we summarise our work and suggest potential future lines of research. We present our proposed spreadsheet template used for storing input parameters in Appendix A.

2. Related work

A detailed survey on CSP can be found in Vangerven et al. (2018). Apart from those mentioned in Vangerven et al. (2018), we discuss the following studies which are related to our work.

A case study regarding the scheduling of 2001 San Antonio meetings of the Public Choice Society was presented in Potthoff and Munger (2003). The problem required the scheduling of submissions into sessions such that submissions of each track are evenly spread among sessions, and ensuring participants are not scheduled in more than one submission of the same session. The authors implemented an integer programming (IP) model to create the schedule of the conference which included 14 tracks and 96 submissions with 10 sessions available.

Potthoff and Brams (2007) extended the previous work by implementing their proposed IP method on both 2005 and 2006 annual meetings of Public Choice Society in New Orleans. The 2005 annual meeting required the assignment of 76 submissions from 13 tracks into 9 sessions, whereas the 2006 annual meeting included 45 submissions from 6 tracks with 9 sessions available. The model had the same objective and all the constraints as in the work of Potthoff and Munger (2003), and included an additional constraint in the IP formulation which considered the unavailability of some presenters to attend certain sessions. Both generated solutions had all constraints satisfied and successfully accomplished the objective.

Nicholls (2007) developed a simple heuristic algorithm to aid the scheduling process of the 2003 Western Decision Science Institute Annual Meeting. This conference involved 330 registered attendees, 295 submissions, 73 regular sessions including four time slots on average, 11 special sessions (a whole session is required for a submission), and 7 rooms of different sizes. The proposed heuristic was used to assist the Program Chair during the scheduling process rather than autonomously produce the conference schedule. The proposed heuristic algorithm did not have an objective function *per se*, but its main purpose was to resolve conflicts by utilising a set of rules and consider preferences from presenters and attendees.

Zulkipli et al. (2013) addressed the Capacity Planing problem variant of conference scheduling. The conference involved 3 tracks, 60 submissions, 5 sessions with 4 time slots each, and 3 rooms were available for parallel scheduling. They collected preferences from participants to create weights for each submission and used these to form the objective function of their goal programming model. The problem required the assignment of submissions into rooms and time slots in such manner that each session achieves a balanced number of submissions with respect to the weights.

Edis and Edis (2013) described a case study in which they considered an artificial conference including 10 tracks and 170 submissions with a 3 days time span. Each day had 4 sessions with 5 time slots each, and 3 rooms were available for parallel scheduling. In their case study, the goal of the primary objective was to minimise the concurrent occurrence of same or similar tracks within the same session in each day. In addition, a secondary objective of the problem was to distribute the number of submissions into sessions in a balanced manner. The authors formulated an integer programming model, along with an extended version to address both objectives.

Another case study was conducted by Quesnelle and Steffy (2015) in which they used real data from the 2013 PenguiCon Conference. The conference was attended by around 1000 participants and involved 253 submissions, 195 presenters, and 14 rooms. In their study, the authors provided problem definitions and showed that the scheduling problem under study along with some variants are all $\mathcal{NP}$-hard. They specifically defined and focused on the *Extended Conference Timetable Decision Problem (ECTTD)* and the *Preference Conference Optimisation Problem (PCO)*. The objective of the ECTTD problem is the assignment of presenters into submissions and time slots, as well as the allocation of submissions into rooms based on their availability and compatibility. The PCO problem includes the objective of minimising participants preferences conflicts by assigning presenters into submissions and time slots, as well as assigning submissions into appropriate rooms. Both ECTTD and PCO were solved with integer programming models.

The CSP of one of the largest conferences within the field of Operations Research, namely the EURO2016 Conference, was addressed in Stidsen et al. (2018). This particular conference included 25 areas of subject, 124 tracks, 1852 submissions, 11 sessions (for each typically 4 time slots were available), 5 buildings with 54 rooms in total, 1600 presenters, and attracted around 2000 participants. The problem required the allocation of subject areas into buildings, and the scheduling of tracks into sessions and rooms so as to comply with the hierarchical structure of the conference. The authors addressed the problem by implementing a multi-objective mixed integer programming (MIP) model which included 5 objectives ranked based on their significance and were sequentially solved following a lexicographic optimisation approach. These objectives were ranked in the following order: (1) Minimisation of the number of areas assigned to different buildings, (2) Maximisation of the number of related areas assigned to the same building, (3) Minimisation of the number of different rooms allocated for each track, (4) Minimisation of the number of time gaps within tracks and, (5) Maximisation of the residual room capacity. Furthermore, room capacity constraints were considered, and parallel scheduling of the same track was not allowed. However, Stidsen et al. (2018) commented that author conflicts were circumvented due to the policy of the conference that allows only one submission per author, and the room utilisation was partially considered due to insufficient data. The authors clarified that the proposed model generates only the high-level schedule, leaving intentionally the low-level schedule to the track organisers. The success of the proposed model is reflected by the fact that it was also used to schedule the IFORS2017, EURO2018, and IFORS2020 conferences, and a slightly improved version of the model to schedule the subsequent EURO and IFORS conferences.

Although our models share some common scheduling requirements with the model of Stidsen et al. (2018), they differ significantly. Similarly to Stidsen et al. (2018), we have considered the fact that the same track must not run in parallel, we minimise the number of rooms utilised per track, we consider the scheduling of tracks consecutively, and we take room capacities into account. However, in our study, we also consider the following requirements. We allow for conference organisers to express preferences regarding the scheduling of tracks into sessions. Our models take into account that certain rooms might not be available during some sessions. We consider presenters conflicts where a presenter might have multiple submissions which is circumvented in Stidsen et al. (2018) study due to the policy of conference. In addition, our approach is suitable for hybrid and online conferences as we consider the timezones of the presenters. Our models accommodate the preferences for sessions and preferences for rooms (accessibility and facility reasons) from presenters. In addition to presenters conflicts, we also consider attendees and track chairs conflicts. Furthermore, we consider the scheduling of submissions that have a different amount of required time slots. In contrast to Stidsen et al. (2018) who generate a high-level schedule, we generate both high-level and low-level schedules considering preferences and requirements on both tracks and submissions levels. In our work we do not consider areas and buildings as described in Stidsen et al. (2018), neither the assignment of similar areas into same buildings, but we allow for conference organisers to specify similar tracks which should not be scheduled in parallel. Our models are significantly different mainly because EURO conference is unusually large and follows an unusual hierarchical structure, which characteristics we believe are not representative of typical conferences.

Vangerven et al. (2018) addressed the CSP of four conferences, namely the MathSport 2013, MAPSP 2015 & 2017, and ORBEL 2017. Their main objective was to maximise the satisfaction of attendees in terms of attending their preferred submissions. Preferences of attendees were collected via e-mail by the authors. A secondary objective was the minimisation of session hopping, which occurs when participants miss parts of their preferred submissions because they are scheduled in different rooms. A third objective of this work was to satisfy the preferences of the presenters. To achieve the three objectives, the researchers proposed a hierarchical three-phased approach. Firstly, the authors maximise the satisfaction of attendees with an integer programming model. Then, they minimise session hopping with either dynamic programming or heuristic approach, and in the last phase, they satisfy the preferences of the presenters with an integer programming model. Their proposed method was implemented on four medium size conferences for which the number of rooms that allowed parallel scheduling ranged between 2 to 4, submissions ranged between 76 to 90, and profiles of attendees ranged between 58 to 101. They also showed that the CSP with n rooms for parallel scheduling is $\mathcal{NP}$-hard when $n \geq 3$.

Another study on CSP was conducted by Manda et al. (2019) who used the dataset from Ecology 2013 for testing purposes to deliver a schedule for the Evolution 2014 conference. While the former conference spanned 5 days including 324 submissions, 8 sessions with 10 time slots each, and 5 rooms, the latter conference spanned 4 days including 1014 submissions, 16 sessions with 5 time slots each, and 14 rooms. In their study, all submissions need to be scheduled into time slots by maximising the coherence within sessions and minimising the similarity between sessions that are scheduled in parallel. Three different approaches, a random, a greedy, and an integer linear programming model were implemented for the initialisation process. These initial solutions were further optimised with a hill climbing algorithm and a simulated annealing algorithm, and with two different optimisation methods. While the first optimisation method optimised the objectives concurrently, the other method was sequential. Through experimentation, the authors found that the different approaches for the initialisation process did not affect the final solution, and that concurrent optimisation was superior to sequential. Therefore, they followed the random initialisation process and the concurrent optimisation method to generate the final schedule. The delivered schedule though was significantly altered by the program committee.

The above case studies indicate that conferences that were considered for an exactly optimised scheduling typically involved up to a few hundred of submissions, while a few larger conferences with over a thousand of submissions were addressed using approximate methods. However, there is a large variety of both exact and approximate methods as well as their constraints and objectives. In the next sections we present new models which have been developed to provide a unifying and generic framework for conference scheduling and would fit most of the conferences described in the above case studies.

3. Description of the conference scheduling problem

In this section, we describe the essential elements of the problem studied in this paper to keep it self-contained; a more detailed description of the problem is provided in Kheiri et al. (2024). We also discuss the functionality of penalties and weights of our approach and present the real-world conferences that motivated our research. We consider that a conference is defined by a set of track-submission pairs $\mathcal{TSU}$, a set of sessions $\mathcal{S}$, and a set of rooms $\mathcal{R}$, whose descriptions and relationships are described next.

We split the whole time of the conference duration into a set of sessions $\mathcal{S}$ representing certain time periods between breaks, which include lunch and coffee breaks, that allow attendees to move between rooms. That is, we assume that every attendee will stay in the same room during a session. We assume that the sessions in $\mathcal{S} := \{1, 2, \dots, S\}$ are chronologically ordered, so S is the last session of the conference. Each session $s \in \mathcal{S}$ consists of a number of time slots defined by a set $\mathcal{TS}_s$. Each time slot ts is defined as a fixed predefined amount of time available for scheduling a submission (e.g., 15 or 20 min). For instance, if a time slot has a 15 min duration, then a session that consists of 4 time slots has a 60 min duration. We assume that all time slots have the same fixed predefined amount of time, but we do not assume that all sessions have the same number of time slots. We use r to denote the room from the set of available rooms $\mathcal{R}$. We assume that submissions are presented in parallel during sessions, and the maximum number of sessions that can be scheduled in parallel is given by the total number of available rooms.

The conference requires scheduling of a number of submissions which we include in set $\mathcal{TSU}$, where each submission $(t, su) \in \mathcal{TSU}$ is uniquely identified as a pair of its track and the submission itself. We assume that each submission (t, su) is categorised into exactly one track from the set of tracks $\mathcal{T}$ based on their subject similarity, i.e., $t \in \mathcal{T}$, and we allow for different tracks to contain different number of submissions $\mathcal{SU}_t$, i.e., $su \in \mathcal{SU}_t$. Each submission usually requires one time slot for scheduling at the conference. However, some submissions, such as a keynote speech, might require additional time slots, which must be scheduled in the same session. For each submission (t, su), we define $n^{(t,su)}$ which indicates its number of required time slots. Similarly, note that a track may utilise more than one session, but we do not know how many sessions beforehand. Although "submission" usually refers to a research paper, it may also refer to any type of formal event that requires scheduling, such as a keynote speech, job market, tutorial, workshop, or any other event. Additional tracks are created accordingly for such formal events.

Furthermore, let $\mathcal{H}$ be the set of all humans involved in the conference. A "presenter" is defined as a person who presents a submission during the conference; we allow for more than one presenter of each submission (which is required e.g., for a panel discussion). We define a set $\mathcal{TSU}^p$ which contains the submissions for which the presenter $p \in \mathcal{P} \subseteq \mathcal{H}$ is the same.

Moreover, we have the following hard requirements:

- **Tracks must be scheduled in only one room:** It is not allowed for a track to utilise more than one room as this would cause inconvenience for attendees that would have to move between rooms. This also implies that the same track must not be scheduled within the same session in different rooms. Assuming that attendees usually attend the whole track of their interest, scheduling the same track in parallel would result in attendees missing some of their preferred submissions.
- **Schedule must be free of presenters conflicts:** In many conferences, authors are allowed to present more than one submission. Therefore, we must ensure that two or more submissions which belong to the same presenter are either scheduled within the same room of a session or scheduled within different sessions. Note that a submission does not necessarily have only one presenter, it may include multiple presenters. We could relax this constraint by considering conflicts only on a time slot level instead of session level. However, this would require presenters to move between rooms which is inconvenient, and they would most likely want to attend both of those sessions.

Apart from these hard requirements, we assume that conference organisers have a number of requests. These are soft requirements that are used to adjust scheduling preferences of conference organisers and satisfy additional requirements of a conference. To accommodate requests, we use a penalty system and a weight system. The soft requirements are:

- **Track-Session request:** We allow to penalise scheduling any track t into any session s by assigning a non-negative penalty α_s^t.
- **Track-Room request:** A non-negative penalty β_r^t is applied for scheduling a certain track t into a specified room r. This allows the allocation of tracks with high expected attendance into appropriate rooms and vice versa.
- **Session-Room request:** In case a room r is unavailable for scheduling during a particular session s, we apply a non-negative penalty $\gamma_{s,r}$.
- **Submission-Session request:** We allow to penalise scheduling any submission (t, su) into any session s by assigning a non-negative penalty $\epsilon_s^{(t,su)}$. This allows the accommodation of preferences from presenters regarding their preferred scheduled time.
- **Submission-Timezone request:** We allow to penalise scheduling any submission (t, su) into any session s by assigning a non-negative penalty $\delta_s^{(t,su)}$ as a result of unsuitability of any presenter's timezone. This is analogous to the Submission-Session request but we keep it separate to allow for a different weight. Considering the timezone may be important for online presenters as well as for in-person presenters experiencing a jet-lag.
- **Submission-Room request:** $\zeta_r^{(t,su)}$ specifies a non-negative penalty for scheduling a particular submission (t, su) into a specified room r. This is used for the consideration of special requests from presenters who might need to present at a particular room for accessibility or facilities issues.

Table 1
Characteristics of the instances: $|\mathcal{TSU}|$ is the number of submissions, $|\mathcal{T}|$ is the number of tracks, $|\mathcal{S}|$ is the number of sessions, $|\mathcal{R}|$ is the number of rooms, $|\mathcal{TS}|$ is the number of time slots across all the sessions, Required TS is the required number of time slots $n^{(t,su)}$ by all the submissions, and Available TS is the number of available time slots for scheduling across all the sessions and rooms, $|\mathcal{R}| \cdot |\mathcal{TS}|$, subtracting penalised time slots due to $\gamma_{s,r} > 0$.

Instance	$\lvert\mathcal{TSU}\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{TS}\rvert$	Required TS	Available TS
GECCO19	202	29	13	10	45	215	450
GECCO20	158	24	7	8	28	161	200
GECCO21	138	27	6	8	24	150	192
N2OR	35	8	4	4	9	36	36
OR60	329	45	8	23	24	417	540
OR60F	279	45	8	23	24	353	540
OR60F2	556	72	16	23	49	702	1,115
OR60F3	1,112	72	32	23	105	1,404	2,403

Each of the above penalties is weighted by a corresponding non-negative value from set $W = \{w_\alpha, w_\beta, w_\gamma, w_\delta, w_\epsilon, w_\zeta\}$ so as to allow the prioritisation of requests.

The goal is to assign all tracks into rooms and sessions, and assign all submissions into sessions in such a way that weighted penalties are minimised and all hard requirements are satisfied. Practically, the problem requires the generation of two schedules, a high-level schedule which indicates the room and sessions of each track, and a low-level schedule indicating the room and session of each submission.

Penalties and weights are used to adjust scheduling preferences. Setting different values for penalties allows the prioritisation of certain requests. For example, if satisfying a certain Track-Session request is more important than another Track-Session request, then we set a greater penalty value for the more important one. In addition, we can prioritise the type of requests that are more important to be satisfied with the use of weights. For instance, if satisfying Submission-Room requests is more important than Submission-Session requests, then we set a greater weight value for Submission-Room requests. Thus, by adjusting penalties and weights, we can explore different solutions along with their trade-offs.

The motivation for this work originated from scheduling the Genetic and Evolutionary Computation Conference (GECCO), the OR Society's 60th Annual Conference (OR60), and the New to OR Conference (N2OR) (Kheiri et al., 2024). We present the details of the instances in Table 1. The characteristics give a rough idea about the size of each instance, yet do not define a given problem fully as the importance of violating a given constraint is not provided. The ratio of Required TS to Available TS can give a rough assessment of hardness of each instance.

4. Methodology

In this section, we first discuss the benefits of using our proposed spreadsheet template for conference scheduling problems. Then we provide an overview of the notation, followed by the formal formulation of the exact model.

We use a spreadsheet file to store input data, which follows a specific template that offers flexibility. This flexible template has been created with the purpose of providing a generic approach suitable for conference scheduling problems. By using a single data format we want to minimise the need of modifying algorithms to fit specific data formats provided by conferences. Instead of adjusting algorithms each time, we could just transfer the given data to the template. In addition, our template makes it easy to modify weights and scheduling preferences. We present the spreadsheet file in detail in Appendix A.

4.1. Model notation

Sets and indices. We use the following sets and their corresponding indices in our formulation:

$t \in \mathcal{T}$: The set of tracks
$su \in \mathcal{SU}_t$: The subset of submissions belonging to a track t
$(t, su) \in \mathcal{TSU}$: The set of submissions where $\{(t, su) : t \in \mathcal{T}$ and $su \in \mathcal{SU}_t\}$
$h \in \mathcal{H}$: The set of humans involved in the conference
$p \in \mathcal{P}$: The set of presenters
$(t, su) \in \mathcal{TSU}^p$: The subset of submissions belonging to presenter p
$r \in \mathcal{R}$: The set of rooms
$s \in \mathcal{S}$: The set of sessions
$ts \in \mathcal{TS}_s$: The subset of time slots belonging to session s

Parameters. We use the following parameters in our formulation:

α_s^t : Penalty for scheduling track t into session s
w_α : Weight of penalty α_s^t
β_r^t : Penalty for scheduling track t into room r
w_β : Weight of penalty β_r^t
$\gamma_{s,r}$: Penalty for utilising room r within session s
w_γ : Weight of penalty $\gamma_{s,r}$
$\delta_s^{(t,su)}$: Penalty for scheduling submission (t, su) within session s for which the timezone is unsuitable
w_δ : Weight of penalty $\delta_s^{(t,su)}$
$\epsilon_s^{(t,su)}$: Penalty for scheduling submission (t, su) within session s
w_ϵ : Weight of penalty $\epsilon_s^{(t,su)}$
$\zeta_r^{(t,su)}$: Penalty for scheduling submission (t, su) into room r
w_ζ : Weight of penalty $\zeta_r^{(t,su)}$
$MaxS_t$: The upper bound on the number of required sessions of track t
$n^{(t,su)}$: The number of required time slots of submission (t, su)
$|\mathcal{TS}_s|$: The number of time slots within session s
$|\mathcal{TSU}^p|$: The number of submissions belonging to presenter p
$M_s^p = min\{|\mathcal{TSU}^p|, |\mathcal{TS}_s|\}$: The upper bound on the number of submissions that presenter p could possibly present during session s (in the same room)

Decision variables. We use the following decision variables in our formulation:

$Z_{s,r}^t \in \{0, 1\}$: 1 if track t is scheduled in session s and room r; 0 if not
$Y_r^t \in \{0, 1\}$: 1 if track t is assigned room r; 0 if not
$X_{s,r}^{(t,su)} \in \{0, 1\}$: 1 if submission (t, su) is scheduled in session s and room r; 0 if not

4.2. Constraints

In the exact model we have the following (hard) constraints:

$$\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}} X_{s,r}^{(t,su)} = 1 \qquad \forall\, (t, su) \in \mathcal{TSU} \tag{1}$$

$$M_s^p X_{s,r}^{(t,su)} + \sum_{r'\in\mathcal{R}\setminus\{r\}} \sum_{(t',su')\in\mathcal{TSU}^p} X_{s,r'}^{(t',su')} \le M_s^p \qquad \forall\, s \in \mathcal{S}, \forall\, r \in \mathcal{R}, \forall\, p \in \mathcal{P}, \quad \forall\, (t, su) \in \mathcal{TSU}^p \tag{2}$$

$$\sum_{r\in\mathcal{R}} Y_r^t = 1 \qquad \forall\, t \in \mathcal{T} \tag{3}$$

$$\sum_{s\in\mathcal{S}} Z_{s,r}^t - MaxS_t Y_r^t \le 0 \qquad \forall\, r \in \mathcal{R}, \forall\, t \in \mathcal{T} \tag{4}$$

$$\sum_{t\in\mathcal{T}} Z_{s,r}^t \le 1 \qquad \forall\, s \in \mathcal{S}, \forall\, r \in \mathcal{R} \tag{5}$$

$$\sum_{su\in\mathcal{SU}_t} n^{(t,su)} X_{s,r}^{(t,su)} - |\mathcal{TS}_s| Z_{s,r}^t \le 0 \qquad \forall\, s \in \mathcal{S}, \forall\, r \in \mathcal{R}, \forall\, t \in \mathcal{T} \tag{6}$$

$$\sum_{su\in\mathcal{SU}_t} X_{s,r}^{(t,su)} - Z_{s,r}^t \ge 0 \qquad \forall\, s \in \mathcal{S}, \forall\, r \in \mathcal{R}, \forall\, t \in \mathcal{T} \tag{7}$$

$$Z_{s,r}^t \in \{0, 1\} \qquad \forall\, t \in \mathcal{T}, \forall\, s \in \mathcal{S}, \forall\, r \in \mathcal{R} \tag{8}$$

$$Y_r^t \in \{0, 1\} \qquad \forall\, t \in \mathcal{T}, \forall\, r \in \mathcal{R} \tag{9}$$

$$X_{s,r}^{(t,su)} \in \{0, 1\} \qquad \forall\, (t, su) \in \mathcal{TSU}, \quad \forall\, s \in \mathcal{S}, \forall\, r \in \mathcal{R} \tag{10}$$

The first set of constraints, Eq. (1), ensures that each submission must be scheduled into exactly one session and one room. The next set of constraints, Eq. (2), resolves presenters conflicts, where $(t, su) \in \mathcal{TSU}^p \subset \mathcal{TSU}$ is a set of submissions including a presenter conflict such that $|\mathcal{TSU}^p| > 1$. For every presenter with multiple submissions, this set of constraints handles conflicts depending on the track of such submissions. If conflicting submissions belong to the same track, then such submissions are not allowed to be scheduled within different rooms of the same session. On the other hand, if conflicting submissions belong to different tracks, then such submissions are scheduled within different sessions.

Constraints within Eq. (3) ensure that exactly 1 room is allocated to each track. Then we use "bigM" constraints, Eq. (4), to allocate tracks to their assigned room, where $MaxS_t$ is the upper bound on the total number of sessions that a given track might require to fit all the submissions. Although we could simply set $MaxS_t$ equal to the total number of sessions, we introduced this scenario to decrease the value of $MaxS_t$ so as to strengthen our formulation. The scenario assumes that a track will utilise at most $MaxS_t$ sessions, where $MaxS_t$ is given by sorting sessions in ascending order based on their number of time slots and sessions are added until the number of time slots is greater or equal to the number of time slots that the given track requires. For example, suppose "Forecasting" track has 6 submissions and "Simulation" track has 4 submissions, each requiring one time slot, and four sessions are available with the following number of time slots: $4, 3, 2, 2$. We sort the sessions $(2, 2, 3, 4)$ and set $MaxS_{Forecasting} = 3$ because $2+2+3 \ge 6$, and $MaxS_{Simulation} = 2$ because $2 + 2 \ge 4$, rather than setting $MaxS = 4$ for both tracks.

Constraints Eq. (5) ensure that at most one track is scheduled into every given session and room. Based on the assignment of Eq. (4), "bigM" constraints Eq. (6) ensure that at most $|\mathcal{TS}_s|$ submissions are allowed to be scheduled into the corresponding session and room, where $|\mathcal{TS}_s|$ is the number of available time slots corresponding to session $s \in \mathcal{S}$, and $n^{(t,su)}$ is the total number of time slots that a given submission requires. For instance, suppose track "Forecasting" is assigned into session "9–11am" and room "A", which is defined as $Z_{9-11am,A}^{Forecasting} = 1$. Also, suppose session "9–11am" has 3 time slots available $|\mathcal{TS}_{9-11am}| = 3$ and all $n^{(t,su)} = 1$, then at most 3 submissions corresponding to track "Forecasting" are allowed to be scheduled into session "9–11am" and room "A". The next set of constraints, Eq. (7), ensures that a given track is not assigned into a session-room pair for which no submissions are scheduled. In other words, we prevent $Z_{s,r}^t = 1$ if none of the submissions is scheduled within the given session and room. Lastly, Eqs. (8), (9), and (10) indicate that our decision variables $Z_{s,r}^t$, Y_r^t, and $X_{s,r}^{(t,su)}$ are binary.

4.3. Objective

Recall from Section 4.1 that $Z_{s,r}^t$ is a binary decision variable which is used for assigning tracks into sessions, where track $t \in \mathcal{T}$, session $s \in \mathcal{S}$, and room $r \in \mathcal{R}$, e.g., when $Z_{9-11am,A}^{Forecasting} = 1$ then track "Forecasting" is allocated into session "9–11am" and room "A". The coefficient of $Z_{s,r}^t$ is a weighted sum of penalties α_s^t, β_r^t, and $\gamma_{s,r}$. Penalty α_s^t is incurred for scheduling a specific track into a specified session (Tracks-Sessions penalty) weighted by w_α. Penalty β^t_r is incurred for scheduling a specific track into a specified room (Tracks-Rooms penalty) weighted by w_β, and penalty $\gamma_{s,r}$ is incurred for utilising a specific room within a specified session (Sessions-Rooms penalty) weighted by w_γ.

Recall also that $X^{(t,su)}_{s,r}$ is a binary decision variable which is used to schedule submissions into sessions, where submission $(t,su) \in \mathcal{TSU}$ corresponds to track $t \in \mathcal{T}$, session $s \in \mathcal{S}$, and room $r \in \mathcal{R}$, e.g., when $X^{(Forecasting,FC1)}_{9-11am,A} = 1$ this means that submission "FC1" corresponding to track "Forecasting" is scheduled in session "9–11am" and room "A". The coefficient of variable $X^{(t,su)}_{s,r}$ is a weighted sum of $\delta^{(t,su)}_s$, $\epsilon^{(t,su)}_s$, and $\zeta^{(t,su)}_r$. Penalty $\delta^{(t,su)}_s$ is a penalty for assigning a specific submission within a session for which the timezone is unsuitable (Submissions-Timezones penalty) weighted by w_δ. e.g., a submission is scheduled within a session that is unsuitable for the timezone of the presenter (03:00 am). Penalty $\epsilon^{(t,su)}_s$ is a penalty for scheduling a specific submission within a specified session (Submissions-Sessions penalty) weighted by w_ϵ, and $\zeta^{(t,su)}_r$ is a penalty for assigning a specific submission into a specified room (Submissions-Rooms penalty) weighted by w_ζ.

Based on the above, we formulate the following objective for the exact model:

$$\min \sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}}(w_\alpha\alpha^t_s + w_\beta\beta^t_r + w_\gamma\gamma_{s,r})Z^t_{s,r} + \sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{(t,su)\in\mathcal{TSU}}(w_\delta\delta^{(t,su)}_s + w_\epsilon\epsilon^{(t,su)}_s + w_\zeta\zeta^{(t,su)}_r)X^{(t,su)}_{s,r} \quad (11)$$

The objective function, Eq. (11), assigns tracks into rooms and sessions, and submissions into sessions by minimising the penalties associated with both tracks and submissions. To reduce the size and complexity of the model, we assign submissions into time slots of sessions in a post-processing algorithm after the IP model is solved. Even though we generate a low-level schedule with a post-processing algorithm, it is still possible for organisers to rearrange the order of submissions within the same session without any impact on the quality of the solution. Note that Eq. (11) implies that $\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{(t,su)\in\mathcal{TSU}} X^{(t,su)}_{s,r} = |\mathcal{TSU}|$, where $|\mathcal{TSU}|$ is a constant (the number of submissions). However, the sum of $Z^t_{s,r}$ is not a constant which may result in some $Z^t_{s,r}$ variables being equal to 1 without any submissions scheduled during a given session and room. We resolve this by including constraints Eq. (7). Alternatively, one could use $(1 + w_\alpha\alpha^t_s + w_\beta\beta^t_r + w_\gamma\gamma_{s,r})$ as a coefficient of $Z^t_{s,r}$ to ensure that variables, for which the sum of penalties is zero, are minimised.

5. Computational results

In this section, we present the results of the exact model which was tested on a number of real and artificial instances. For the real instances, we obtained past data and scheduling preferences information which were used to set penalties values and weights. The artificial instances were created due to a particular instance being infeasible and to test the models on larger instances. All instances and models are freely available at Kheiri et al. (2024).

The results were generated on an i7-11370H CPU Intel Processor with 8 cores at 3.30 GHz with 16.00 GB RAM. We used Python 3.8.3 to build our models which were solved using Gurobi 9.5.0. We use the following Gurobi parameters for the exact model; $MIPGap = 0$ and $timeLimit = 3600$. The former parameter allows the solver to terminate only when the gap between the lower and upper objective bound is zero, while the latter parameter implies a time limit of one hour. Note that even though the time required to build the models is negligible for some instances, it is significant for other instances. We report the time required to build the models for each instance in Appendix B.

In Table 2, we present the penalty values that were used for each type of constraint. For example, in GECCO19 instance, we set $\beta^t_r = 10$ for cases that are less important to be satisfied, while we set $\beta^t_r = 10\,000$ for significant cases. Penalty values reflect the importance of satisfying the particular constraint.

For most instances we keep all weights identical and equal to one ($w_\alpha = w_\beta = w_\gamma = w_\delta = w_\epsilon = w_\zeta = 1$). The only exception in which we set different weights is for the GECCO conferences where we use the following weights; $w_\alpha = 100$, $w_\gamma = 10$, and $w_\epsilon = 100$. In addition, we set $w_\delta = 100$ for GECCO21 which was held online.

Our results are summarised in Table 3 and we present in detail the violations of soft constraints in Table 4. For each instance, we set a time limit of one hour and we report the number of variables and constraints, the objective value, the gap which shows the relative gap between the two objective bounds, and the time required for the solver to terminate in seconds. An infeasible status for the objective means that the solution is infeasible. Additionally, note that even though we include the Submission-Room request ($\zeta^{(t,su)}_r$) in the model, there is no available data for this request.

Our model obtained the optimal solution for the N2OR conference instantly without any violations. The model achieved a solution with a relative gap of 0.001% within 17 s for GECCO19, but the optimal solution was not found within the time limit of 1 h. The solution had 1 violation for Tracks-Sessions requests ($w_\alpha \times \alpha^t_s = 100 \times 10,000 =$

Table 2
Penalty Values: α^t_s indicates values for Tracks-Sessions penalties, β^t_r indicates values for Tracks-Rooms penalties, $\gamma_{s,r}$ indicates values for Sessions-Rooms penalties, $\delta^{(t,su)}_s$ indicates values for Submissions-Timezones penalties, $\epsilon^{(t,su)}_s$ indicates values for Submissions-Sessions penalties, and $\zeta^{(t,su)}_r$ indicates values for Submissions-Rooms penalties. When a type of penalties is not used, it is denoted as "–".

Instance	α^t_s	β^t_r	$\gamma_{s,r}$	$\delta^{(t,su)}_s$	$\epsilon^{(t,su)}_s$	$\zeta^{(t,su)}_r$
N2OR	–	–	–	–	[1]	–
GECCO19	[10000]	[10, 10000]	–	–	[1]	–
GECCO20	[10000]	[10, 10000]	[10000]	–	[1, 10]	–
GECCO21	[10000]	[10, 10000]	–	[1, 10]	[10000]	–
OR60	[10, 100]	[1, 100]	[1000]	–	[1]	–
OR60F	[10, 100]	[1, 100]	[1000]	–	[1]	–
OR60F2	[10, 100]	[1, 100]	[1000]	–	[1]	–
OR60F3	[10, 100]	[1, 100]	[1000]	–	[1]	–

Table 3
Exact Model Results: Objective indicates the aggregation of penalties caused by violations of soft constraints, Gap indicates the relative gap between the two objective bounds, and Time indicates the required time for the solver to terminate in seconds. N/A indicates the value is not available.

Instance	Variables	Constraints	Objective	Gap (%)	Time (s)
N2OR	720	347	0	0.000	0.1
GECCO19	30,320	26,911	1,000,010	0.001	3,600.0
GECCO20	10,384	5,750	6,110	0.000	8.3
GECCO21	8,136	4,797	11,130	0.000	19.7
OR60	69,851	36,185	Infeasible	N/A	3.6
OR60F	60,651	30,983	424	0.000	13.9
OR60F2	232,760	77,724	10	0.000	88.9
OR60F3	873,080	153,720	0	0.000	137.4

Table 4
Exact Model Violations: α's indicates incurred weighted Tracks-Sessions penalties (i.e., $\sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} w_\alpha\alpha^t_s Z^t_{s,r}$), β's indicates incurred weighted Tracks-Rooms penalties, γ's indicates incurred Sessions-Rooms penalties, δ's indicates incurred Submissions-Timezones penalties, ϵ's indicates incurred Submissions-Sessions penalties, and ζ's indicates incurred Submissions-Rooms penalties. When a type of penalties is not used, it is denoted as "–". N/A indicates the value is not available.

Instance	α's	β's	γ's	δ's	ϵ's	ζ's	Total
N2OR	–	–	–	–	0	–	0
GECCO19	1,000,000	10	–	–	0	–	1,000,010
GECCO20	0	10	0	–	6,100	–	6,110
GECCO21	0	30	–	11,100	0	–	11,130
OR60	N/A	N/A	N/A	–	N/A	–	N/A
OR60F	400	0	0	–	24	–	424
OR60F2	0	0	0	–	10	–	10
OR60F3	0	0	0	–	0	–	0

1,000,000) and 1 violation for Tracks-Rooms requests ($w_\beta \times \beta_r^t = 1 \times 10 = 10$). GECCO20 had 1 violation for Tracks-Rooms requests ($w_\beta \times \beta_r^t = 1 \times 10 = 10$) and 16 violations for Submissions-Sessions requests ($w_\epsilon \times (11 \times \epsilon_s^{(t,su)} + 5 \times \epsilon_s^{(t,su)}) = 100 \times (11 \times 1 + 5 \times 10) = 6,100$). The solution of GECCO21 had 3 violations for Tracks-Rooms requests ($w_\beta \times 3 \times \beta_r^t = 1 \times 3 \times 10 = 30$) and 30 violations for Submissions-Timezones requests ($w_\delta \times (21 \times \delta_s^{(t,su)} + 9 \times \delta_s^{(t,su)}) = 100 \times (21 \times 1 + 9 \times 10) = 11{,}100$). OR60 had some extensive tracks that resulted in infeasibility due to constraints Eqs. (6) and (7). In other words, the number of available sessions was not enough to avoid scheduling the same track in parallel, and consequently tracks could not be limited to utilise only one room. Therefore, we reduced the instance by removing submissions of such tracks to create a feasible version of OR60, which we refer to as OR60F. The model found the optimal solution which had 4 violations for Tracks-Sessions requests ($w_\alpha \times 4 \times \alpha_s^t = 1 \times 4 \times 100 = 400$) and 24 violations for Submissions-Sessions requests ($w_\epsilon \times 24 \times \epsilon_s^{(t,su)} = 1 \times 24 \times 1 = 24$). OR60F2 and OR60F3 are both larger versions of the OR60 instance. The solution of the former instance had only 10 violations for Submissions-Sessions requests ($w_\epsilon \times 10 \times \epsilon_s^{(t,su)} = 1 \times 10 \times 1 = 10$), while the solution of the latter instance had no violations.

5.1. Infeasible instances

Sometimes, it is not possible to completely schedule each track into exactly one room, which results in an infeasible model (e.g., OR60). A solution to this issue is to relax constraints Eq. (3) by changing the right hand side to $\leq MaxR_t$, where $MaxR_t$ is the maximum number of rooms that could be assigned to a track. We followed this procedure for the OR60 instance which we describe next. Firstly, we identified the four tracks that required more than one room to be scheduled. Specifically, three tracks had to utilise 2 rooms and one track had to utilise 3 rooms. However, the model was still infeasible because of presenters conflicts and we had to further relax constraints Eq. (3). After changing the right hand side value several times, we found a feasible model in which two tracks utilise 2 rooms, one track utilises 3 rooms, and one track utilises 4 rooms. Our model found the optimal solution within 51.2 s which had an objective value of 106. The solution had 1 violation for Tracks-Sessions requests ($w_\alpha \times \alpha_s^t = 1 \times 100 = 100$), 3 violations for Tracks-Rooms requests ($w_\beta \times 3 \times \beta_r^t = 1 \times 3 \times 1 = 3$), and 3 violations for Submissions-Sessions requests ($w_\epsilon \times 3 \times \epsilon_s^{(t,su)} = 1 \times 3 \times 1 = 3$).

6. Extended formulation

In this section, we present an extension of our formulation in which we consider additional constraints including some with non-linear terms. We first provide additional definitions, followed by a discussion of the additional soft and hard requirements. Next, we present the formal formulation of the extended model and we conclude the section by presenting computational results.

We refer to an "attendee" as a person who is a spectator of a submission, and define a set $\mathcal{TSU}^a$ which includes the submissions that the attendee $a \in \mathcal{A} \subseteq \mathcal{H}$ wishes to attend. A "track chair" is defined as the person who attends all the submissions of a track, and the set $\mathcal{T}^c$ consists of tracks chaired by the same person $c \in \mathcal{C} \subseteq \mathcal{H}$. Note that a human is allowed to have multiple roles and may attend a conference as a presenter, an attendee, and a track chair.

We consider the following additional hard requirements:

- **Avoid scheduling similar tracks in parallel:** Organisers can specify some tracks as being similar and request not to schedule such tracks in parallel. For each track t we denote a set $\mathcal{T}^t \subset \mathcal{T}$ of similar tracks including t.
- **Schedule must be free of attendees conflicts:** For attendees who have declared attending preferences, we either schedule such submissions within the same room of a session or within different sessions to resolve conflicts.
- **Schedule must be free of track chairs conflicts:** In case of a track chair being responsible for more than one track, we schedule such tracks within different sessions.

Additionally, we consider the following soft requirement:

- **Consecutive track sessions:** Tracks are preferred to be scheduled in consecutive sessions to achieve a cohesive schedule.

The additional soft requirement, is weighted by a non-negative value π_K. This weight, however, is a subsidy, not penalty, and therefore it has a negative sign as the objective is to be minimised. We model the consecutive tracks requirement as $Z_{s,r}^t \times Z_{s+1,r}^t$ which decreases the objective value by π_K when a track t is scheduled consecutively within sessions s and $s+1$. For instance, suppose we need to schedule track "Forecasting" into room "A" and two sessions from the set $\mathcal{S} = \{9-11am, 11-1pm, 1-3pm\}$ are required. If "9–11am" and "1–3 pm" sessions are used, then $Z_{9-11am,A}^{Forecasting} \times Z_{11-1pm,A}^{Forecasting} = 1 \times 0 = 0$ and $Z_{11-1pm,A}^{Forecasting} \times Z_{1-3pm,A}^{Forecasting} = 0 \times 1 = 0$. On the other hand, if either "9–11am" and "11–1 pm" or "11–1 pm" and "1–3 pm" are used, then we prefer any of these combinations which would result in decreasing the objective value by π_K given that none other penalties are incurred. Note that the more consecutive variables become equal to 1, the fewer the violations of the consecutive tracks soft constraint is achieved. By including the additional soft requirement in Eq. (11), we achieve the following objective function with non-linear terms:

$$\begin{aligned} \min \quad & \sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}}(w_\alpha\alpha_s^t + w_\beta\beta_r^t + w_\gamma\gamma_{s,r})Z_{s,r}^t \\ & + \sum_{s\in\mathcal{S}}\sum_{r\in\mathcal{R}}\sum_{(t,su)\in\mathcal{TSU}}(w_\delta\delta_s^{(t,su)} + w_\epsilon\epsilon_s^{(t,su)} + w_\zeta\zeta_r^{(t,su)})X_{s,r}^{(t,su)} \\ & -\pi_K \times \sum_{s\in\mathcal{S}\setminus\{S\}}\sum_{r\in\mathcal{R}}\sum_{t\in\mathcal{T}} Z_{s,r}^t \times Z_{s+1,r}^t \end{aligned} \tag{12}$$

Then, we convert the non-linear terms of the objective function into linear by introducing new binary variables. Let $K_{s,r}^t$ be a product variable of $Z_{s,r}^t$ and $Z_{s+1,r}^t$ which is used to schedule tracks in a consecutive manner.

6.1. Constraints

In the extended model we have the following (hard) constraints:

(1) – (10)

$$\sum_{r\in\mathcal{R}}\sum_{t'\in\mathcal{T}^t} Z_{s,r}^{t'} \leq 1 \qquad \forall\, s\in\mathcal{S}, \forall\, t\in\mathcal{T} \tag{13}$$

$$M_s^a X_{s,r}^{(t,su)} + \sum_{r'\in\mathcal{R}\setminus\{r\}}\sum_{(t',su')\in\mathcal{TSU}^a} X_{s,r'}^{(t',su')} \leq M_s^a \qquad \forall\, s\in\mathcal{S}, \forall\, r\in\mathcal{R}, \forall\, a\in\mathcal{A}, \forall\,(t,su)\in\mathcal{TSU}^a \tag{14}$$

$$\sum_{r\in\mathcal{R}}\sum_{t^c\in\mathcal{T}^c\subset\mathcal{T}} Z_{s,r}^{t^c} \leq 1 \qquad \forall\, s\in\mathcal{S}, \forall\, c\in\mathcal{C} \tag{15}$$

$$K_{s,r}^t \leq Z_{s,r}^t \qquad \forall\, t\in\mathcal{T}, \forall\, r\in\mathcal{R}, \forall\, s\in\mathcal{S}\setminus\{S\} \tag{16}$$

$$K_{s,r}^t \leq Z_{s+1,r}^t \qquad \forall\, t\in\mathcal{T}, \forall\, r\in\mathcal{R}, \forall\, s\in\mathcal{S}\setminus\{S\} \tag{17}$$

$$K_{s,r}^t \geq Z_{s,r}^t + Z_{s+1,r}^t - 1 \qquad \forall\, t\in\mathcal{T}, \forall\, r\in\mathcal{R}, \forall\, s\in\mathcal{S}\setminus\{S\} \tag{18}$$

$$K_{s,r}^t \in \{0,1\} \qquad \forall\, t\in\mathcal{T}, \forall\, r\in\mathcal{R}, \forall\, s\in\mathcal{S}\setminus\{S\} \tag{19}$$

Constraints Eq. (13) prevent scheduling specified tracks in parallel, where $\mathcal{T}^t \subset \mathcal{T}$ is a set of similar tracks such that $|\mathcal{T}^t| > 1$. The next set of constraints, Eq. (14), resolves attendees conflicts, where $(t,su) \in \mathcal{TSU}^a \subset \mathcal{TSU}$ is a set of submissions including an attendee conflict such that $|\mathcal{TSU}^a| > 1$. In addition, M_s^a is an upper bound on the number of declared submissions that attendee a could possibly attend during session s given by $min\{|TSU^a|, |\mathcal{TS}_s|\}$, where $|TSU^a|$ is the number of declared submissions by attendee a. For every attendee with multiple declared submissions, this set of constraints handles conflicts depending on the track of such submissions. If conflicting submissions belong to the same track, then such submissions are not allowed to be scheduled within different rooms of the same session. On the other hand, if conflicting submissions belong to different tracks, then such submissions are scheduled within different sessions. The next set of constraints, Eq. (15), resolves track chairs conflicts, where $t^c \in \mathcal{T}^c \subset \mathcal{T}$ is a set of tracks including a track chair conflict such that $|\mathcal{T}^c| > 1$. For every track chair responsible for more than one track, this set of constraints ensures that such tracks are not scheduled within the same session. Constraints from Eq. (16) up to Eq. (18) introduce auxiliary variables $K^t_{s,r}$ which we will use to convert the non-linear terms in the objective into linear terms, while constraints Eq. (19) indicate that these variables are binary.

6.2. Objective

After replacing the non-linear terms in Eq. (12), we obtain the following objective function:

$$\min \sum_{s\in S}\sum_{r\in R}\sum_{t\in\mathcal{T}}(w_\alpha\alpha^t_s + w_\beta\beta^t_r + w_\gamma\gamma_{s,r})Z^t_{s,r} + \sum_{s\in S}\sum_{r\in R}\sum_{(t,su)\in\mathcal{T}\mathcal{SU}}(w_\delta\delta^{(t,su)}_s + w_\epsilon\epsilon^{(t,su)}_s + w_\zeta\zeta^{(t,su)}_r)X^{(t,su)}_{s,r} - \pi_K \times \sum_{s\in S}\sum_{r\in R}\sum_{t\in\mathcal{T}}K^t_{s,r} \quad (20)$$

The new objective function, Eq. (20), generates a schedule by minimising the penalties related to both tracks and submissions, and the violations regarding consecutive track sessions.

6.3. Computational results

We use the same weights for the extended model as we did for the exact model. Additionally, we use the following weight for consecutive track sessions; $\pi_K = 10$. We also used the same Gurobi parameters and included an additional parameter; $IntegralityFocus = 1$, which forces variables to take exact integer values. This additional parameter was used because we noticed that sometimes "bigM" constraints were violated due to variables that meant to be zero, instead took non-trivial values. As a result of this side-effect, infeasible solutions were accepted by the solver. The exact model was free of this side-effect and therefore we did not use that parameter.

For the extended model, the objective values have been computed through evaluation functions in order to obtain the objective value of the exact model, because the objective value of the extended model itself does not provide reliable information regarding the quality of the solution. In Table 5 we present the penalties along with constraints for each instance, and in Table 6 we present the results. An infeasible status for the objective means that the model is infeasible, while unknown indicates that the solver did not find a feasible solution within the time limit of one hour (3,600 s). The extended model is significantly larger compared to the exact model in the number of variables and especially in the number of constraints. Note that even though we include attendees and track chair conflicts in the model, we do not have available real data for these constraints.

In Table 7, we present in detail the violations of soft constraints for the extended model. The optimal solution was found for the N2OR instance, which had only 1 violation for a Submissions-Sessions request ($w_\epsilon \times \epsilon^{(t,su)}_s = 1 \times 1 = 1$). For GECCO19, the model found the optimal solution, which had 2 violations for Tracks-Sessions ($w_\alpha \times 2 \times \alpha^t_s = 100 \times 2 \times 10,000 = 2,000,000$) and 7 violations regarding consecutive tracks ($\pi_K \times 7 \times K = 10 \times 7 \times 1 = 70$). The model achieved the optimal solution for GECCO20 with the following violations; 1 violation for Tracks-Rooms ($w_\beta \times \beta^t_r = 1 \times 10 = 10$), 4 violations for consecutive tracks ($\pi_K \times 4 \times K = 10 \times 4 \times 1 = 40$), and 14 violations for Submissions-Sessions ($w_\epsilon \times (7 \times \epsilon^{(t,su)}_s + 7 \times \epsilon^{(t,su)}_s) = 100 \times (7 \times 1 + 7 \times 10) = 7,700$). The solution of GECCO21 is also optimal with 3 violations for Tracks-Rooms requests ($w_\beta \times 3 \times \beta^t_r = 1 \times 3 \times 10 = 30$) and 30 violations for Submissions-Timezones requests ($w_\delta \times (21 \times \delta^{(t,su)}_s + 9 \times \delta^{(t,su)}_s) = 100 \times (21 \times 1 + 9 \times 10) = 11,100$). The time limit was reached for OR60F where the model achieved a solution with a 2.5% optimality gap. The solution had 4 violations for Tracks-Sessions requests ($w_\alpha \times 4 \times \alpha^t_s = 1 \times 4 \times 100 = 400$), and 33 violations for Submissions-Sessions requests ($w_\epsilon \times 33 \times \epsilon^{(t,su)}_s = 1 \times 33 \times 1 = 33$). For the remaining instances, OR60F2 and OR60F3, the model reached the time limit without finding any solution.

Table 5
Penalties & Constraints: Penalties indicate the number of penalties, conflicts indicate the number of all conflict types, and similar tracks indicates the number of similar tracks.

Instance	Penalties	Conflicts	Similar tracks
N2OR	10	1	1
GECCO19	405	149	12
GECCO20	290	54	15
GECCO21	112	42	11
OR60	1,478	98	15
OR60F	1,382	70	0
OR60F2	2,059	70	0
OR60F3	3,457	70	0

Table 6
Extended Model Results: Objective indicates the aggregation of penalties caused by violations of soft constraints, Gap indicates the relative gap between the two objective bounds, and time indicates the required time for the solver to terminate in seconds. N/A indicates the value is not available.

Instance	Variables	Constraints	Objective	Gap (%)	Time (s)
N2OR	816	671	1	0.00	0.8
GECCO19	33,800	38,950	2,000,070	0.00	57.5
GECCO20	11,536	10,095	7,750	0.00	51.8
GECCO21	9,216	8,541	11,130	0.00	20.5
OR60	77,096	58,016	Infeasible	N/A	4.8
OR60F	67,896	52,718	433	2.50	3,600.0
OR60F2	257,600	152,244	Unknown	N/A	3,600.0
OR60F3	924,416	307,728	Unknown	N/A	3,600.0

Table 7
Extended Model Violations: K's indicates incurred weighted penalties of consecutive tracks (other notation as in Table 4).

Instance	α's	β's	γ's	δ's	ϵ's	ζ's	K's	Total
N2OR	–	–	–	–	1	–	0	1
GECCO19	2,000,000	0	–	–	0	–	70	2,000,070
GECCO20	0	10	0	–	7,700	–	40	7,750
GECCO21	0	30	–	11,100	0	–	0	11,130
OR60	N/A	N/A	N/A	–	N/A	–	N/A	N/A
OR60F	400	0	0	–	33	–	0	433
OR60F2	N/A	N/A	N/A	–	N/A	–	N/A	N/A
OR60F3	N/A	N/A	N/A	–	N/A	–	N/A	N/A

6.3.1. Infeasible instances

We modified our extended model as in Section 5.1 and tried to solve OR60, however, this time our model was still infeasible due to constraints Eq. (13) that prevent scheduling similar tracks in parallel. We first tried to identify which particular constraints to relax from Eq. (13), but it was not a straightforward task. Therefore, we had to remove some similar tracks restrictions which decreased the number of similar tracks from 15 to 8. Specifically, we removed six tracks that were labelled as similar with one track, and another pair of tracks. This led to a feasible model which reached the time limit of one hour and returned a solution with an objective value of 143. The solution had 1 violation for Tracks-Sessions requests ($w_\alpha \times \alpha^t_s = 1 \times 100 = 100$), 30 violations for Tracks-Rooms requests ($w_\beta \times 30 \times \beta^t_r = 1 \times 30 \times 1 = 30$), and 13 violations for Submissions-Sessions requests ($w_\epsilon \times 13 \times \epsilon^{(t,su)}_s = 1 \times 13 \times 1 = 13$).

7. Conclusion

This work has provided two integer programming models along with a generic approach to address conference scheduling problems. We have shown that our approach generates low-level schedules for conferences in a fully automated manner. Our weighted penalty system allows the exploration of multiple solutions by adjusting the weights of the soft constraints. An easy-to-use spreadsheet template is used to fit the needs of different conferences. Apart from in-person conferences, we have considered timezone constraints in this work which also makes it suitable for scheduling hybrid and online conferences. We have demonstrated the suitability of our mathematical models by testing them on real data from five different conferences and on additional artificial instances. The results have shown the success of the exact model in finding optimal solutions for almost all instances. The extended model also found optimal and near-optimal solutions for some instances, and revealed some limitations due to its increased size and the complexity of some constraints.

The additional constraints in the extended model add much more complexity, but we believe such constraints are essential for conference scheduling. Having many hard constraints brings limitations to our models in terms of feasibility. In addition, some conferences have to schedule same tracks in parallel due to limited number of sessions, such as OR60. Ideally, to achieve a more robust generic approach, we would want to convert most hard constraints into soft so as to explore additional trade-offs and solutions. However, such a mathematical model would be too slow in terms of computational time. Therefore, in order to overcome such limitations, we suggest the investigation of alternative methods for future research, such as heuristics (Pylyavskyy et al., 2020), to develop an approach to the largest and most complex conference scheduling problems. Lastly, in our future work along with developing heuristics, we will also consider the submissions ordering constraint, which will allow organisers to express preferences regarding the presentation sequence of submissions within their tracks.

CRediT authorship contribution statement

Yaroslav Pylyavskyy: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Peter Jacko:** Conceptualization, Formal analysis, Methodology, Supervision, Validation, Writing – review & editing. **Ahmed Kheiri:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Methodology, Software, Supervision, Validation, Writing – review & editing.

Acknowledgements

The work reported in this paper has been supported by the UK Research and Innovation through the Programme Grant EP/V520214/1.

Appendix A

In this section, we provide a sample spreadsheet file to demonstrate its usage. The spreadsheet file consists of the following sheets; Submissions, Tracks, Sessions, Rooms, Parameters, Tracks-Sessions Penalty, Tracks-Rooms Penalty, Similar Tracks, and Sessions-Rooms Penalty.

The Submissions sheet includes all the necessary information regarding submissions as well as submissions related penalties as shown in Fig. 1. Column A contains the reference name or ID of each submission. Column B indicates the track name of the corresponding submission. Column C is used to indicate the number of time slots that each submission requires. Column D is ignored in our model. Column E refers to the timezone in which the presenters of the corresponding submission are located. Column F is used to list the presenter or multiple presenters of the corresponding submission. Similarly, column G is used to list the attendees. This column may include non-presenters, such as co-authors, but it may also include presenters. The latter case will be considered as an attendee conflict during optimisation. The next number of columns is determined by the total number of available sessions, where each column corresponds to a session (from column H to column K in this example). Under these columns a penalty value may be set accordingly so as not to schedule the corresponding submission into the corresponding session. For instance, Submission_7 must be ideally scheduled in Session_1 or in Session_2 so we keep these values empty. Additionally, we do not want to schedule Submission_7 in Session_3 or Session_4, but if that cannot be fully satisfied then we prefer Session_3. To do so, we set a penalty value of 1 for Session_3 and a penalty value of 10 for Session_4. Then, the number of the remaining columns is determined by the total number of available rooms, where each column corresponds to a room (from column L to column O in this example). Within these columns a penalty value may be set accordingly so as not to schedule the corresponding submission into the corresponding room. For example, if we want Submission_9 scheduled in Room_2, we penalise all rooms except for Room_2.

The Tracks sheet contains information regarding track names and the list of track chair names as presented in Fig. 2.

In column A, the tracks are listed, and column B contains the names of track chairs for each track. Each track is not limited to only one track chair, it may have multiple track chairs. A chair conflict is created in case there are two tracks with the same person as track chair. If any of the track chairs is also a presenter in another track, then this will be considered as a presenter conflict. Lastly, an attendee conflict is created when the same person is a track chair and at the same time is an attendee at a submission that belongs to another track.

Next, the Sessions sheet contains all the necessary information regarding available sessions as displayed in Fig. 3. Column A includes the names of the sessions. Column B refers to the total number of available time slots per session. Column C indicates the date for each session, while column D and E are used to set the starting and ending time for each session respectively.

We skip the Rooms sheet as it simply contains the names of the available rooms. The Parameters sheet includes settings for hybrid or online conferences, and allows to set weight values for penalties as shown in Fig. 4. Columns A and B are associated with settings regarding hybrid or online conferences. The local timezone field refers to the timezone that applies at the location of the conference. Next, suitable scheduling times fields indicate the ideal scheduling time window for which submissions are not penalised. Less suitable scheduling times fields create a new time window for which submissions are slightly penalised, while unsuitable scheduling times are heavily penalised submissions. All times are converted into local times of online presenters. For instance, a submission would be penalised by 1 if the converted local time of the presenter is between 7:00 and 9:30 or between 21:30 and 23:00. If the converted local time is between 23:00 and 7:00 then a penalty of 10 will apply, otherwise if the converted local time is between 9:30 and 21:30 then no penalty applies. These settings are used to identify suitable sessions that are convenient for online presenters. Lastly, columns D and E are used to set the weight values. Setting different weight values allows the prioritisation of the listed types of penalties.

The Tracks-Sessions Penalty sheet is used to define penalty values to avoid scheduling a specified track into a specified session as presented in Fig. 5. Column A includes all tracks, and the number of next columns is given by the total number of sessions available, where each column corresponds to a session (from column B to column E in this example). For instance, Track_5 must be ideally scheduled in Session_3 and/or in Session_4 so we keep these values empty. Additionally, we do not want to schedule Track_5 in Session_1 or Session_2, but if that cannot be fully satisfied then we prefer Session_2. To do so, we just set a small penalty value for Session_2 and a high penalty value for Session_1.

Reference	Track	Required Timeslots	Order	Time Zone	Presenters	Attendees	Session_1	Session_2	Session_3	Session_4	Room_1	Room_2	Room_3	Room_4
Submission_1	Track_1	1	0	GMT+0	Name_Surname_1	Name_Surname_35								
Submission_2	Track_1	2	0	GMT+1	Name_Surname_2	Name_Surname_36								
Submission_3	Track_1	3	0	GMT+2	Name_Surname_3	Name_Surname_37								
Submission_4	Track_1	4	0	GMT+3	Name_Surname_4	Name_Surname_38								
Submission_5	Track_1	1	0	GMT+4	Name_Surname_5	Name_Surname_39								
Submission_6	Track_2	1	0	GMT+5	Name_Surname_6	Name_Surname_40								
Submission_7	Track_2	1	0	GMT+6	Name_Surname_7	Name_Surname_41			1	10				
Submission_8	Track_2	1	0	GMT+7	Name_Surname_8	Name_Surname_42								
Submission_9	Track_2	1	0	GMT+8	Name_Surname_9	Name_Surname_43					10		10	10
Submission_10	Track_2	1	0	GMT+9	Name_Surname_10	Name_Surname_44								
Submission_11	Track_3	1	0	GMT+10	Name_Surname_11	Name_Surname_45	1	10						
Submission_12	Track_3	1	0	GMT+11	Name_Surname_12	Name_Surname_46	1	10						
Submission_13	Track_3	1	0	GMT+12	Name_Surname_13	Name_Surname_47								
Submission_14	Track_4	1	0	GMT-12	Name_Surname_14	Name_Surname_48								
Submission_15	Track_4	1	0	GMT-11	Name_Surname_15	Name_Surname_49								
Submission_16	Track_4	1	0	GMT-10	Name_Surname_16	Name_Surname_50								
Submission_17	Track_4	1	0	GMT-9	Name_Surname_17	Name_Surname_51								
Submission_18	Track_5	2	1	GMT-8	Name_Surname_18	Name_Surname_52			1	10				
Submission_19	Track_5	1	0	GMT-7	Name_Surname_19	Name_Surname_53								
Submission_20	Track_5	1	0	GMT-6	Name_Surname_20	Name_Surname_54								
Submission_21	Track_5	1	0	GMT-5	Name_Surname_21	Name_Surname_55								
Submission_22	Track_5	1	0	GMT-4	Name_Surname_22	Name_Surname_56								
Submission_23	Track_6	1	0	GMT-3	Name_Surname_23	Name_Surname_57								
Submission_24	Track_6	1	0	GMT-2	Name_Surname_24	Name_Surname_58								

Fig. 1. Submissions sheet.

Tracks	Chairs
Track_1	Name_Surname_1
Track_2	Name_Surname_2
Track_3	Name_Surname_3
Track_4	Name_Surname_4
Track_5	Name_Surname_5
Track_6	Name_Surname_6
Track_7	Name_Surname_7
Track_8	Name_Surname_8

Fig. 2. Tracks sheet.

Sessions	Max Number of Timeslots	Date	Start Time	End Time
Session_1	2	7/28/2021	9:30	10:30
Session_2	4	7/28/2021	10:45	11:45
Session_3	2	7/29/2021	9:30	10:30
Session_4	3	7/29/2021	10:45	11:45

Fig. 3. Sessions sheet.

We use Tracks-Rooms Penalty sheet to control the scheduling process of tracks into rooms as displayed in Fig. 6. Column A contains all tracks, and the number of next columns is given by the total number of rooms available, where each column corresponds to a room (from column B to column E in this example). For instance, if we want Track_1 and Track_2 scheduled in Room_4, then we set a penalty value for all rooms except for Room_4.

The Similar Tracks sheet allows to define which pair of tracks should not be scheduled in parallel as shown in Fig. 7. Column A includes all tracks, and the number of next columns is given by the total number of tracks, where each column corresponds to a track (from column B to

	A	B	C	D	E
1	Sessions			Weights	
2	Local time zone:	GMT+0		Tracks_Sessions|Penalty:	10
3	Suitable scheduling times			Tracks_Rooms|Penalty:	10
4	From:	9:30		Sessions_Rooms|Penalty:	20
5	To:	21:30		Similar Tracks:	10
6	Less suitable scheduling times			Number of Rooms per Track:	0
7	From:	7:00		Parallel Tracks:	0
8	To:	23:00		Consecutive Tracks:	10
9	Penalty:	1		Submissions_Timezones:	5
10	Unsuitable scheduling times			Submissions Order:	0
11	Penalty:	10		Submissions_Sessions|Penalty:	5
12				Submissions_Rooms|Penalty:	5
13				Presenters Conflicts:	0
14				Attendees Conflicts:	0
15				Chairs Conflicts:	0
16				Presenters Conflicts Timeslot Level:	0
17				Attendees Conflicts Timeslot Level:	0

Fig. 4. Parameters sheet.

	A	B	C	D	E
1		Session_1	Session_2	Session_3	Session_4
2	Track_1				
3	Track_2				
4	Track_3				
5	Track_4				
6	Track_5	10	1		
7	Track_6				
8	Track_7				
9	Track_8				

Fig. 5. Tracks-Sessions Penalty sheet.

	A	B	C	D	E
1		Room_1	Room_2	Room_3	Room_4
2	Track_1	10	10	10	
3	Track_2	10	10	10	
4	Track_3				
5	Track_4				
6	Track_5				
7	Track_6				
8	Track_7				
9	Track_8				

Fig. 6. Tracks-Rooms Penalty sheet.

	A	B	C	D	E	F	G	H	I
1		Track_1	Track_2	Track_3	Track_4	Track_5	Track_6	Track_7	Track_8
2	Track_1								
3	Track_2								
4	Track_3						1		1
5	Track_4								
6	Track_5								
7	Track_6								
8	Track_7								
9	Track_8								

Fig. 7. Similar Tracks sheet.

	A	B	C	D	E
1		Room_1	Room_2	Room_3	Room_4
2	Session_1				
3	Session_2				
4	Session_3				
5	Session_4			10	

Fig. 8. Sessions-Rooms Penalty sheet.

	A	B	C	D	E
1		**Steelhouse LT**	**Stafford 1**	**Stafford 2**	**Steelhouse 1**
2	**Wed1**	Modelling and Simulation	Supply Chain & Transportation Management	Metaheuristics	Education
3	**Wed2**	Modelling and Simulation	Optimisation	Metaheuristics	Education
4	**Thu1**	Analytics	Optimisation	Metaheuristics	Big Data and AI
5	**Thu2**	Analytics	Optimisation	Consultancy	Big Data and AI
6					
7	**Wed1**	NEW19A3748	NEW19A3746	NEW19A3750	NEW19A3759
8	**Wed1**	NEW19A20	NEW19A3730	NEW19A3750	NEW19A3758
9	**Wed2**	NEW19A3734	NEW19A3731	NEW19A3756	NEW19A3723
10	**Wed2**	NEW19A3743	NEW19A3739	NEW19A3742	NEW19A3722
11	**Thu1**	NEW19A3729	NEW19A3721	NEW19A22	NEW19A19
12	**Thu1**	NEW19A3736	NEW19A3740	NEW19A21	NEW19A3744
13	**Thu2**	NEW19A3738	NEW19A3745	NEW19A11	NEW19A3737
14	**Thu2**	NEW19A3754	NEW19A3747	NEW19A3733	NEW19A3728
15	**Thu2**	NEW19A3755	NEW19A10	NEW19A3749	NEW19A3735

Fig. 9. Solution example.

column I in this example). Suppose Track_3 is similar to Track_6 and Track_8 and we do not want to schedule Track_3 and Track_6 or Track_3 and Track_8 in parallel. We define this by simply setting a penalty value for that pairs of tracks. Notice that the value of the penalty does not support preferences here among pairs of tracks because we include this in the model as a hard constraint.

Lastly, we use Sessions-Rooms Penalty sheet to define unavailability of rooms for certain sessions as presented in Fig. 8. Column A contains all sessions, and the number of next columns is given by the total number of available rooms, where each column corresponds to a room (from column B to column E in this example). For instance, if Room_3 is unavailable during Session_4, then we add a penalty value for that session-room pair.

After the completion of the optimisation, we generate a new spreadsheet file that contains the optimised schedule and the violations report. In Fig. 9, we present a solution example by solving the N2OR instance with the extended model where the upper timetable refers to the tracks solution and the lower timetable refers to the submissions solution. The violations report is presented in Fig. 10, where we report the objective value along with details of the violations.

	A	B	C
1	Obj	Evaluate Submissions|Sessions	
2	Final Objective = 1	NEW19A3728 - Thu2	1
3		Total	1

Fig. 10. Violations report example.

Table 8
Overall computing time: t_b indicates the time required to build the model, t_s indicates the required time for the solver to terminate, and t_t indicates the total time required. All times are in seconds.

Instance	Exact Model			Extended Model		
	t_b	t_s	t_t	t_b	t_s	t_t
N2OR	0.1	0.1	0.2	0.1	0.8	0.9
GECCO19	158.9	3,600.0	3,758.9	152.2	57.5	209.7
GECCO20	4.9	8.3	13.2	5.7	51.8	57.5
GECCO21	1.8	19.7	21.5	2.2	20.5	22.7
OR60	143.6	3.6	147.2	151.8	4.8	156.6
OR60F	79.0	13.9	92.9	99.2	3,600.0	3,699.2
OR60F2	272.6	88.9	361.5	344.5	3,600.0	3,944.5
OR60F3	1,072.7	137.4	1,210.1	1,211.8	3,600.0	4,811.8

Appendix B

In Table 8 we present the time required to build the mathematical models for each instance.

References

Edis, E., & Edis, R. S. (2013). An integer programming model for the conference timetabling problem. *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, *9*(2), 55–62.

Eglese, R. W., & Rand, G. K. (1987). Conference seminar timetabling. *Journal of the Operational Research Society*, *38*(7), 591–598.

Ehrgott, M. (2005). *Multicriteria optimization*: *vol. 491*, Springer Science & Business Media.

Kheiri, A., Pylyavskyy, Y., & Jacko, P. (2024). CSPLib - A benchmark library for conference scheduling problems. https://github.com/ahmedkheiri/CSPLib.

Manda, P., Hahn, A., Beekman, K., & Vision, T. J. (2019). Avoiding "conflicts of interest": a computational approach to scheduling parallel conference tracks and its human evaluation. *PeerJ Computer Science*, *5*, Article e234.

Nicholls, M. G. (2007). A small-to-medium-sized conference scheduling heuristic incorporating presenter and limited attendee preferences. *Journal of the Operational Research Society*, *58*(3), 301–308.

Potthoff, R. F., & Brams, S. J. (2007). Scheduling of panels by integer programming: results for the 2005 and 2006 new orleans meetings. *Public Choice*, *131*(3–4), 465–468.

Potthoff, R. F., & Munger, M. C. (2003). Use of integer programming to optimize the scheduling of panels at annual meetings of the Public Choice Society. *Public Choice*, *117*(1–2), 163–175.

Pylyavskyy, Y., Kheiri, A., & Ahmed, L. (2020). A reinforcement learning hyper-heuristic for the optimisation of flight connections. In *2020 IEEE congress on evolutionary computation* (pp. 1–8).

Quesnelle, J., & Steffy, D. (2015). Scheduling a conference to minimize attendee preference conflicts. In *Proceedings of the 7th multidisciplinary international conference on scheduling: theory and applications* (pp. 379–392).

Sampson, S. E. (2004). Practical implications of preference-based conference scheduling. *Production and Operations Management*, *13*(3), 205–215.

Stidsen, T., Pisinger, D., & Vigo, D. (2018). Scheduling EURO-k conferences. *European Journal of Operational Research*, [ISSN: 0377-2217] *270*(3), 1138–1147.

Thompson, G. M. (2002). Improving conferences through session scheduling. *Cornell Hotel and Restaurant Administration Quarterly*, *43*(3), 71–76.

Vangerven, B., Ficker, A. M., Goossens, D. R., Passchyn, W., Spieksma, F. C., & Woeginger, G. J. (2018). Conference scheduling — A personalized approach. *Omega*, [ISSN: 0305-0483] *81*, 38–47.

Zulkipli, F., Ibrahim, H., & Benjamin, A. M. (2013). Optimization capacity planning problem on conference scheduling. In *2013 IEEE business engineering and industrial applications colloquium* (pp. 911–915).