CoSPLib - A Benchmark Library for Conference Scheduling Problems

Ahmed Kheiri
ahmed.kheiri@manchester.ac.uk
The University of Manchester
Manchester, M15 6PB, UK

Yaroslav Pylyavskyy
y.pylyavskyy1@lancaster.ac.uk
Lancaster University
Lancaster, LA1 4YX, UK

Peter Jacko
p.jacko@lancaster.ac.uk
Lancaster University, UK &
Berry Consultants, UK

Abstract

Conference scheduling problems require the generation of schedules for conferences by considering multiple constraints to offer the best possible experience to participants. Previous studies have been published in the literature, however, the comparison and evaluation of the developed methods is challenging, as these methods have been designed to address rather specific requirements of the conferences being studied per se. In this work, we attempt to bridge this gap by making benchmark data publicly available to researchers. We also present a selection hyper-heuristic algorithm to solve the benchmark instances and provide computational results. We aim to raise awareness of the under-studied conference scheduling problem, and to encourage researchers to contribute to the benchmark dataset with new instances, constraints, and solving methods.

CCS Concepts

• **Computing methodologies → Planning and scheduling**; **Discrete space search**.

Keywords

Benchmark, Scheduling, Metaheuristic, Hyper-heuristic

ACM Reference Format:

Ahmed Kheiri, Yaroslav Pylyavskyy, and Peter Jacko. 2025. CoSPLib - A Benchmark Library for Conference Scheduling Problems. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3712255.3726570

1 Introduction

Conferences play a crucial role in academia, providing researchers with the opportunity to present their ideas and receive valuable feedback. They also serve as a platform for networking with colleagues in similar fields, fostering the formation of new collaborations. Researchers benefit from the opportunity to learn about new methods and ideas in their fields. When organising a conference it is therefore important to have a schedule that allows the conference to run smoothly. The conference scheduling problem (CoSP) is a combinatorial optimisation problem that was initially presented by Eglese and Rand [4]. Even though the CoSP relates to well-studied problems, such as class timetabling problems, it has not received much attention from researchers. The CoSP involves creating a high-level and a low-level schedule. The former involves organising tracks into sessions and allocating rooms, whereas the latter involves assigning submissions to specific time slots within sessions. The problem was proved to be $\mathcal{NP}$-hard in [15].

The literature has studies tackling the CoSP. However, it is challenging to compare and evaluate the developed methods. The primary issue is that many conferences have different scheduling requirements, objectives, and constraints, some of which are case-specific. As a result, there are various problem descriptions, objective functions, and developed methods in the literature which depend on the need of the particular conference. Therefore, benchmark data are needed to provide a fair comparison between the developed methods. Moreover, benchmark instances will create a competitive environment for the development of advanced algorithms to solve CoSP, and raise awareness of the problem which is a real-world problem that has not been studied as much as related problems. Last but not least, researchers could contribute to the development of CoSPLib by submitting new instances, constraints, and solving methods. In this paper, we are describing what we believe to be the first conference scheduling benchmark. We also present a hyper-heuristic algorithm to create conference schedules.

2 Background

Thompson [14] distinguished between two perspectives for CoSP, depending on constraints and objectives of the given problem: Presenter-Based Perspective (PBP) and Attender-Based Perspective (ABP). PBP focuses on optimising presenters' preferences, such as scheduling presenters on their preferred day or time, while ABP prioritises attendees' preferences, such as attending preferred sessions without conflicts or space-shortening problems. Thus, the quality of the schedule is subjective and sensitive to the perspective considered. Similarly to other studies, in this paper, we propose a variant that combines both perspectives in a single model.

In [9], the research focused on scheduling submissions into sessions in a balanced manner using an integer programming model. Potthoff and Brams [8] refined this model by incorporating presenter availability, which successfully solved a CoSP that included more than 70 submissions. Edis and Edis [3] addressed a similar CoSP, aiming to minimise similar tracks scheduled concurrently and balance submissions across sessions, using an integer programming model for a conference with about two hundred submissions. Nicholls [6] introduced a heuristic algorithm to facilitate organisers in scheduling and tested it on a conference with roughly three hundred submissions and attendees. Stidsen et al. [13] addressed the CoSP of the EURO2016 Conference, which is one of the largest conferences within the field of operational research. This CoSP involved the assignment of areas into buildings and the allocation of tracks

GECCO '25 Companion, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3726570

into sessions and rooms. The authors followed a lexicographic approach using a multi-objective mixed-integer programming model that consisted of five ranked objectives to solve the problem. The objectives, in order of priority, aimed to minimise the number of areas scheduled in different buildings, maximise the number of areas scheduled in the same building, minimise the number of rooms utilised by each track, minimise time gaps across scheduled tracks and maximise the residual room capacity. Constraints included room capacity and prevention of the same track running in parallel, however, presenters' conflicts were excluded and room capacity was limited by data scarcity. Their solution approach focuses on the high-level schedule and leaves the management of the low-level schedule to the track organisers. Vangerven et al. [15] provided a solution to the scheduling of four conferences: MathSport 2013, MAPSP 2015 & 2017, and ORBEL 2017. The primary goal of the CoSP was to maximise attendees' satisfaction by aligning with their preferred submissions. Furthermore, the research aimed to minimise session hopping, resulting in attendees missing parts of their preferred sessions when switching rooms. The third objective included the maximisation of presenters' preferences. The authors successfully optimised the schedules of the four conferences with a hierarchical three-phase approach using integer programming, dynamic programming, and heuristics. Manda et al. [5] utilised the dataset from Ecology 2013 for testing purposes to optimise the schedule of the Evolution 2014 conference. The aim of the CoSP was to schedule submissions into time slots to maximise consistency within sessions and minimise similarity across parallel sessions. The authors experimented with random, greedy and integer linear programming approaches to create initial solutions, which they optimised using hill climbing and simulated annealing algorithms. Bulhões et al. [1] studied a different perspective from typical CoSP in which they mainly focused on grouped optimally submissions into tracks based on their similarity. However, they also scheduled submissions into sessions considering constraints such as presenter conflicts and parallel tracks with the purpose of maximising the total benefit and defined the problem as clustering-based CoSP. Three formulations were presented and tested on artificial instances, each of the approaches suitable for different sizes of instances. They also obtained optimal solutions for two real instances which were adopted by the organisers of the XV Latin American Robotics Symposium and the Brazilian Logic Conference. Patro et al. [7] defined the virtual CoSP that required maximisation of efficiency and fairness objectives subject to attendees' preferences and their availability based on time zone information. The authors developed an exact and a heuristic method that were tested on both real and artificial datasets. The virtual CoSP was limited to single-track conferences where parallel sessions are ignored. Riquelme et al. [12] generated artificial instances with similar characteristics to GECCO 2019 and solved them via integer programming and simulated annealing. The problem was defined by the authors as a track-based CoSP requiring the scheduling of tracks into sessions and rooms, and the scheduling of submissions into time slots of sessions with the purpose of minimising the number of missing seats. Rezaeinia et al. [11] supported the scheduling of LOGMS, INFORMS TSL Workshop and ICSP by developing and testing three different approaches. Although the CoSP tackled was the same as described in [15], the authors experimented by relaxing the problem and allowing submissions to be freely scheduled regardless of their assigned track. However, this resulted in many submissions being scheduled in tracks in which they did not fit well based on feedback from the organising committee. Finally, a generic approach for conference scheduling was presented by Pylyavskyy et al. [10] who developed two integer programming models and tested their performance on the instances presented in this study.

3 CoSPLib

3.1 Problem Description

The CoSP that we consider in this study includes a set of *tracks* (stream, subject area, topic) along with their corresponding *submissions* (e.g., papers, presentations, talks), a set of available *sessions* along with their corresponding *time slots*, and a set of available *rooms*. We assume that submissions are already assigned to their tracks, and we define "submission" as any type of formal event (e.g., research paper, keynote speech, workshop, etc.) that needs to be scheduled. We consider "sessions" to be periods of time. For each session, a number of available time slots is assigned where submissions are scheduled. That is, a time slot is defined as a fixed amount of time for presentation. Even though time slots have fixed available time, sessions may consist of different number of time slots. While a submission typically requires one time slot, some submissions (e.g., tutorial) may require additional time slots to be completely scheduled, satisfying the HC_2 constraint as defined in Section 3.1.1. The problem requires the scheduling of tracks into sessions and rooms, and the scheduling of submissions into time slots within sessions and rooms while respecting a set of constraints.

3.1.1 Hard Constraints Constraints within this class must be satisfied in order to ensure a feasible schedule is achieved.

- HC_1 *Feasibility*: All submissions must be scheduled.
- HC_2 *Extended Submissions*: Submissions requiring more than one time slot must be scheduled within the same session.
- HC_3 *Presenters Conflicts*: On the occasion of having two or more submissions which belong to the same presenter, such submissions should be either scheduled within the same room of a session or scheduled within different sessions. Conflicts can be handled either on a session or a time slot level.
- HC_4 *Track Chairs Conflicts*: In case of a track chair being responsible for more than one track, such tracks should be scheduled within different sessions.

3.1.2 Soft Constraints This class includes soft constraints which do not have to be necessarily satisfied to achieve a feasible schedule, but the quality of the schedule is determined based on how many and which of these constraints are satisfied.

- SC_1 *Track-Session Preference Matrix*: The aversion of conference organisers to schedule a specified track into a specified session.
- SC_2 *Track-Room Preference Matrix*: The aversion of conference organisers to schedule a specified track into a specified room.
- SC_3 *Session-Room Preference Matrix*: The aversion of organisers to utilise a specified room during a specified session. Sometimes, certain rooms might be unavailable during certain sessions.
- SC_4 *Submission-Session Preference Matrix*: The aversion of presenters to present during a specified session.
- SC_5 *Parallel Tracks*: Avoid scheduling the same track in parallel.

Table 1: Parameters sheet

Sessions		Weights	
Local time zone:	GMT+0	Tracks_Sessions|Penalty:	0
Suitable scheduling times		Tracks_Rooms|Penalty:	0
From:	09:30	Sessions_Rooms|Penalty:	0
To:	21:30	Similar Tracks:	1
Less suitable scheduling times		Number of Rooms per Track:	10
From:	07:00	Parallel Tracks:	1
To:	23:00	Consecutive Tracks:	1
Penalty:	1	Submissions_Timezones:	0
Unsuitable scheduling times		Submissions Order:	1000
Penalty:	10	Submissions_Sessions|Penalty:	10
		Submissions_Rooms|Penalty:	0
		Presenters Conflicts:	100000
		Attendees Conflicts:	0
		Chairs Conflicts:	0
		Presenters Conflicts Timeslot Level:	0
		Attendees Conflicts Timeslot Level:	0

- SC_6 *Limit the Number of Rooms per Track*: Scheduling the same track into different rooms should be avoided. Having a track scheduled in multiple rooms is inconvenient for the participants as they would have to switch rooms frequently.
- SC_7 *Consecutive Tracks*: Schedule tracks consecutively.
- SC_8 *Submission's Order*: Submissions should be scheduled with respect to their specified order.
- SC_9 *Track-Track Preference Matrix*: The aversion of conference organisers to schedule a pair of tracks in parallel. This is because some conferences may have a number of tracks which are similar with the potential of attracting the interest of the same audience.
- SC_{10} *Presenter's Time Zone*: On the occasion of an online or hybrid conference, the time zone of each presenter should be considered to schedule their submission within a suitable session.
- SC_{11} *Submission-Room Preference Matrix*: The aversion of presenters to present within a specified room. Some examples are that a room may provide specific facilities which others do not provide, and some rooms may be easier to access in comparison to others.
- SC_{12} *Attendees Conflicts*: On the occasion of having two or more submissions for which an attendee has declared attending preference, such submissions should be either scheduled within the same room of a session or scheduled within different sessions. These conflicts can be handled on a session or a time slot level.

3.1.3 Data Format Description Each instance within the benchmark dataset is stored as an Excel file and follows a specific format. The file consists of the following nine sheets: parameters, submissions, tracks, sessions, rooms, tracks_sessions penalty, tracks_rooms penalty, similar tracks, and sessions_rooms penalty.

The parameters sheet includes time zone and weights settings (see Table 1). Under the "Sessions" column it is possible to set the local time zone of the location at which the conference takes place. It is also possible to set the time range that is considered suitable for scheduling submissions of online presenters. If the time of the presenter's location is outside the specified range for their scheduled submission, then a penalty is incurred. The "Weights" column allows the setting of weight values for all the soft constraints.

The submissions sheet contains information and constraints for each submission as shown in Table 2. The first column indicates the reference of each submission and the second column shows their assigned track. In the third column the number of time slots that each submission requires is indicated, followed by the fourth column where the scheduling order of each submission is specified (if irrelevant, zero value is used). For example, Sub 1 should be the first scheduled submission in Track 1, and Sub 2 should be scheduled second. The "Presenters" and "Attendees" columns indicate the presenters and attendees for each submission respectively from which presenters and attendees conflicts are defined. The next number of columns is given by the total number of available sessions

Table 2: Submissions sheet

Reference	Track	Required Time Slots	Order	Presenters	Attendees	Session 1	Session 2	Session 3	Room 1	Room 2
Sub 1	Track 1	1	1	P1	A1					
Sub 2	Track 1	1	2	P2	A2	1	10			
Sub 3	Track 2	1	0	P3	A3, A5					
Sub 4	Track 2	2	0	P4, P5	A4					10

and form the submission-session preference matrix. For example, if Sub 2 is scheduled in Session 1 or Session 2, then a penalty of 1 or 10 will be incurred respectively. Similarly, the remaining number of columns is given by the total number of available rooms and form the submission-room preference matrix (e.g., a penalty of 10 will be incurred if Sub 4 is scheduled in Room 2).

The tracks sheet contains track names in the first column, followed by the track chairs in the second column as shown in Table 3. The sessions sheet includes the name of each session in the first column (see Table 4). In the second column, the available number of time slots for each session are indicated. For instance, up to four submissions can be scheduled in Session 1, but in Session 2 at most two submissions can be scheduled assuming a duration of 15 minutes per time slot. The remaining columns indicate the date, the start time, and the end time of each session. The rooms sheet simply contains the name of each room in the first column.

Table 3: Tracks sheet

Tracks	Chairs
Track 1	C1, C2
Track 2	C3
Track 3	C4
Track 4	C5

Table 4: Sessions sheet

Session	# Time Slots	Date	Start Time	End Time
Session 1	4	28/07/2021	09:30	10:30
Session 2	2	28/07/2021	14:00	14:30
Session 3	2	29/07/2021	09:30	10:00
Session 4	3	29/07/2021	10:30	11:15

The tracks_sessions penalty sheet corresponds to the track-session preference matrix. In Table 5, an example is provided where a penalty of 1 incurs if Track 1 is scheduled in Session 1, while a penalty of 10 incurs if Track 4 is scheduled in Session 4. The tracks_rooms penalty sheet is used to form the track-room preference matrix as shown in Table 6. In this example a penalty of 10 is incurred if Track 2 is scheduled in either Room 1 or Room 3, while a penalty of 1 is incurred if Track 2 is scheduled in Room 4.

Table 5: Tracks_Sessions Penalty Sheet

	Session 1	Session 2	Session 3	Session 4
Track 1	1			
Track 2				
Track 3				
Track 4				10

Table 6: Tracks_Rooms Penalty Sheet

	Room 1	Room 2	Room 3	Room 4
Track 1				
Track 2	10		10	1
Track 3				
Track 4				

The similar tracks sheet corresponds to the track-track preference matrix. An example is provided in Table 7 where a penalty of 1 is incurred if Track 1 is scheduled in parallel with Track 2, while a penalty of 10 is incurred if Track 1 is scheduled in parallel with Track 3. Lastly, the sessions_rooms penalty sheet forms the session-room preference matrix as shown in Table 8. In this example, a penalty of 100 is incurred if any room except for Room 4 is utilised during Session 2.

Table 7: Tracks_Tracks Penalty Sheet

	Track 1	Track 2	Track 3	Track 4
Track 1	-	1	10	
Track 2	-	-		
Track 3	-	-	-	
Track 4	-	-	-	-

Table 8: Sessions_Rooms Penalty Sheet

	Room 1	Room 2	Room 3	Room 4
Session 1				
Session 2	100	100	100	
Session 3				
Session 4				

3.2 The Library

The dataset is made publicly available and can be downloaded from https://github.com/ahmedkheiri/CoSPLib. It consists of 16 instances from 4 conferences, namely the Genetic and Evolutionary Computation Conference (GECCO), the OR Society's 60th Annual Conference (OR60), the New to OR Conference (N2OR), and the International Symposium on Forecasting (ISF). We present the characteristics of each instance in Table 9. We created new instances based on OR60: OR60F by removing submissions at random, and OR60F2 and OR60F3 by adding submissions, tracks, and sessions.

Table 9: Benchmark instances

Instance	Subm.	Tracks	Sess.	Rooms	Slots	Instance	Subm.	Tracks	Sess.	Rooms	Slots
GECCO19	202	29	13	10	45	GECCO23	207	26	6	9	60
GECCO20	158	24	7	8	28	GECCO23 Workshop	233	55	8	9	80
GECCO20 Poster	131	1	2	1	132	ISF22	311	49	11	10	36
GECCO20 Workshop	131	26	8	10	40	N2OR	35	8	4	4	9
GECCO21	138	27	6	8	24	OR60	329	45	8	23	24
GECCO21 Workshop	203	28	8	10	56	OR60F	279	45	8	23	24
GECCO22	179	39	7	8	56	OR60F2	556	72	16	23	49
GECCO22 Workshop	138	59	8	10	80	OR60F3	1112	72	32	23	105

4 Hyper-heuristic for CoSP

We have developed a selection hyper-heuristic to generate optimised conference schedules. (A survey of selection hyper-heuristics can be found in [2].) The solution is represented by two matrices which correspond to a high-level and a low-level conference schedule. In the first matrix (high-level), rows represent sessions, columns represent rooms, and elements indicate the scheduled track as shown in Table 10. In the second matrix (low-level), rows represent the time slots for each session, columns represent rooms, and elements show the scheduled submission as shown in Table 11.

Table 10: High-level Schedule Solution Example

Session \ Room	Room 1	Room 2	Room 3
Session 1	Track 1	Track 2	Track 3
Session 2	Track 1	Track 2	Track 5
Session 3	Track 1	Track 4	Track 6

Table 11: Low-level Schedule Solution Example

Timeslot \ Room	Room 1	Room 2	Room 3
Timeslot 1 (Session 1)	Sub 1	Sub 9	Sub 14
Timeslot 2 (Session 1)	Sub 2	Sub 10	Sub 15
Timeslot 3 (Session 1)	Sub 3	Sub 11	Sub 16
Timeslot 1 (Session 2)	Sub 4	Sub 12	Sub 20
Timeslot 2 (Session 2)	Sub 5	Sub 13	Sub 21
Timeslot 1 (Session 3)	Sub 6	Sub 17	Sub 22
Timeslot 2 (Session 3)	Sub 7	Sub 18	Sub 23
Timeslot 3 (Session 3)	Sub 8	Sub 19	Sub 24

Our hyper-heuristic uses three low-level heuristics, specifically two swap heuristics and a reverse heuristic. One of the swap heuristics is applied on the high-level schedule and swaps a track with another track randomly. The other swap heuristic is applied on the low-level schedule and swaps two submissions of the same track at random. The reverse heuristic is applied on the low-level schedule where a randomly selected sequence of submissions of the same track are reversed. To enhance the exploration of solutions, we introduce random shuffling of the current solution at regular intervals during the algorithm's execution. Preliminary experiments showed that shuffling every 600 seconds balances exploration and exploitation. A feasible solution is achieved by satisfying the hard constraints, and the quality of the solution is given by evaluating the violations of the weighted soft constraints presented in Section 3. Each of the soft constraints, $SC = \{sc_1, \ldots, sc_n\}$, is assigned a weight, $w = \{w_{sc_1}, \ldots, w_{sc_n}\}$, according to their subjective significance, where n is the total number of constraints. The goal is to minimise the total penalty cost derived from violations of the soft constraints: $\sum_{n=1}^{n} w_{sc_n} \times V_{sc_n}$, where V_{sc_n} indicates the corresponding violated amount of constraint SC_n.

An i7-11370H CPU Intel Processor at 3.30GHz with 16.00GB RAM was used to generate the results and the algorithm was developed in Python 3.8. Each instance was solved in a single run within a 1 hour time limit. The weights used and the results are reported in Table 12. Note that, as we are using a heuristic method, the hard constraints are treated as soft constraints but with very high weights to ensure their satisfaction in the final solution. This approach resulted in feasible solutions in all instances, with the only exception being the GECCO20 instance, which resulted in 3 presenter conflicts.

5 Conclusion

In this work, we have provided a benchmark dataset, the CoSPLib, for the under-studied conference scheduling problem. We believe that CoSPLib will facilitate the comparison and evaluation processes of different developed methods tackling CoSPs. We hope that the research community will contribute to the further development of the CoSPLib by submitting new instances, constraints, and methods. This could potentially create competition in terms of developing new scheduling methods and raise awareness of the real-world problem of conference scheduling. We have also presented a hyper-heuristic algorithm and solved all instances from the benchmark dataset. We have reported the computational results of our method to allow their comparison with the results of future works.

Table 12: Weights (upper table) and violations (lower table) of soft constraints and objective value per instance. When a constraint is not used, it is denoted as "-"

Instance	w_{SC_1}	w_{SC_2}	w_{SC_3}	w_{SC_4}	w_{SC_5}	w_{SC_6}	w_{SC_7}	w_{SC_8}	w_{SC_9}	$w_{SC_{10}}$	$w_{SC_{11}}$	$w_{SC_{12}}$
GECCO19	10^2	1	-	10^2	1	10	1	-	1	-	-	-
GECCO20	10^2	1	10	10^2	1	10	1	10^6	1	-	-	-
GECCO20 Poster	-	-	-	1	-	-	-	-	-	-	-	-
GECCO20 Workshop	10^2	-	-	10^2	1	10^4	10^4	10^4	1	-	-	-
GECCO21	10^2	1	-	10^2	1	10	1	10^6	1	1	-	-
GECCO21 Workshop	10^2	1	-	10^2	1	10	10^4	10^4	1	1	-	-
GECCO22	10^2	1	10	10^2	1	10	1	10^6	1	1	-	-
GECCO22 Workshop	10^2	1	-	10^2	1	10^4	10^4	10^4	1	1	-	-
GECCO23	10^2	1	10	10^2	1	10	1	10^6	1	1	-	-
GECCO23 Workshop	10^2	-	-	10^2	1	10^4	10^4	10^4	1	1	-	-
ISF22	10^2	10^2	10^5	10^2	10^4	50	1	10^2	10^4	-	-	10
N2OR	-	-	-	10	1	10	1	10^3	1	-	-	-
OR60	10^2	1	10	10	1	10	1	10^3	1	-	-	-
OR60F	10^2	1	10	10	1	10	1	10^3	-	-	-	-
OR60F2	10^2	1	10	10	1	10	1	10^3	-	-	-	-
OR60F3	10^2	1	10	10	1	10	1	10^3	-	-	-	-

Instance	V_{SC_1}	V_{SC_2}	V_{SC_3}	V_{SC_4}	V_{SC_5}	V_{SC_6}	V_{SC_7}	V_{SC_8}	V_{SC_9}	$V_{SC_{10}}$	$V_{SC_{11}}$	$V_{SC_{12}}$	Objective
GECCO19	0	20	-	0	3	70	11	-	10	-	-	-	114
GECCO20	0	0	0	4,600	1	20	4	0	40	-	-	-	4,695
GECCO20 Poster	-	-	-	0	-	-	-	-	-	-	-	-	0
GECCO20 Workshop	2,100	-	-	6,000	26	70,000	0	0	0	-	-	-	78,126
GECCO21	0	30	-	0	1	10	1	0	0	111	-	-	153
GECCO21 Workshop	300	250	-	11,100	22	100	0	0	41	240	-	-	12,053
GECCO22	0	0	0	1,900	0	0	3	0	20	446	-	-	2,369
GECCO22 Workshop	3,100	100	-	400	0	0	0	0	20	425	-	-	4,045
GECCO23	0	10,112	0	900	1	10	5	5,000,000	0	58	-	-	5,011,086
GECCO23 Workshop	100	-	-	0	0	0	0	20,000	0	51	-	-	20,151
ISF22	600	100	100,000	0	0	150	13	0	10,000	-	-	0	110,863
N2OR	-	-	-	0	0	0	1	0	0	-	-	-	1
OR60	0	916	0	210	29	260	16	34,000	46	-	-	-	35,477
OR60F	0	811	0	220	13	250	16	9,000	-	-	-	-	10,310
OR60F2	0	1,016	10,000	620	5	670	53	39,000	-	-	-	-	51,364
OR60F3	0	6,232	0	730	30	2,260	71	123,000	-	-	-	-	132,323

References

[1] T. Bulhões, R. Correia, and A. Subramanian. 2022. Conference scheduling: a clustering-based approach. *European Journal of Operational Research* 297, 1 (2022), 15–26.

[2] J. H. Drake, A. Kheiri, E. Özcan, and E. K. Burke. 2020. Recent advances in selection hyper-heuristics. *European Journal of Operational Research* 285, 2 (2020), 405–428.

[3] E. Edis and R. Edis. 2013. An integer programming model for the conference timetabling problem-konferans çizelgeleme problemi için bir tamsayili programlama modeli. *Celal Bayar Üniversitesi Fen Bilimleri Dergisi* 9, 2 (2013), 55–62.

[4] R. W. Eglese and G. K. Rand. 1987. Conference Seminar Timetabling. *Journal of the Operational Research Society* 38, 7 (1987), 591–598.

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

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

[7] G. K. Patro, P. Jana, A. Chakraborty, K. P. Gummadi, and N. Ganguly. 2022. Scheduling Virtual Conferences Fairly: Achieving Equitable Participant and Speaker Satisfaction. In *Proceedings of the ACM Web Conference 2022.* 2646–2656.

[8] R. F. Potthoff and S. J. Brams. 2007. Scheduling of panels by integer programming: Results for the 2005 and 2006 New Orleans meetings. *Public Choice* 131, 3–4 (2007), 465–468.

[9] R. F. Potthoff and M. C. Munger. 2003. Use of Integer Programming to Optimize the Scheduling of Panels at Annual Meetings of the Public Choice Society. *Public Choice* 117, 1–2 (2003), 163–175.

[10] Y. Pylyavskyy, P. Jacko, and A. Kheiri. 2024. A generic approach to conference scheduling with integer programming. *European Journal of Operational Research* 317, 2 (2024), 487–499.

[11] N. Rezaeinia, J. C. Góez, and M. Guajardo. 2024. Scheduling conferences using data on attendees' preferences. *JORS* (2024), 1–14.

[12] F. Riquelme, E. Montero, L. Pérez-Cáceres, and N. Rojas-Morales. 2022. A Track-Based Conference Scheduling Problem. *Mathematics* 10, 21 (2022).

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

[14] G. M. Thompson. 2002. Improving Conferences Through Session Scheduling. *Cornell Hotel and Restaurant Administration Quarterly* 43, 3 (2002), 71–76.

[15] B. Vangerven et al. 2018. Conference scheduling — A personalized approach. *Omega* 81 (2018), 38–47.