Efficient Scheduling of GECCO Conferences using Hyper-heuristic Algorithms

Ahmed Kheiri
a.kheiri@lancaster.ac.uk
Department of Management Science
Lancaster University
Lancaster, LA1 4YX, UK

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

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

ABSTRACT

We propose the development of a conference scheduler tailored specifically for the Genetic and Evolutionary Computation Conference (GECCO). Our proposed flexible approach allows GECCO organisers to optimise conference schedules according to their specific needs and available resources. Using hyper-heuristic methods, our scheduler generates optimised solutions for in-person and hybrid GECCO conferences. We validate our method using data from GECCO2019 and demonstrate its effectiveness by successfully creating schedules for GECCO conferences from 2020 onwards.

CCS CONCEPTS

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

KEYWORDS

Conference scheduling problem, Optimisation, Hyper-heuristic

ACM Reference Format:
Ahmed Kheiri, Yaroslav Pylyavskyy, and Peter Jacko. 2024. Efficient Scheduling of GECCO Conferences using Hyper-heuristic Algorithms. In *Proceedings of The Genetic and Evolutionary Computation Conference 2024 (GECCO '24).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Conferences serve as pivotal events within scientific communities, offering researchers a platform to disseminate their findings, engage in academic discussions, and establish valuable connections for potential collaborations. Traditionally, conferences were predominantly conducted in-person, fostering direct interaction among attendees. However, with the advent of the internet and particularly in response to the COVID-19 pandemic, conference formats have evolved significantly. Now, conferences can be conducted entirely online or in a hybrid format, blending physical attendance with remote participation options.

For conference participants, there are three primary costs. Firstly, there are financial expenses, including registration fees (for all participants), travel costs and accommodation (for those attending in-person). Secondly, there is the investment of time, such as travel time (for in-person attendees) and time spent preparing presentations (for all participants). Lastly, there is the environmental impact, as conferences can have a notable effect on the environment [5]. Conference organisers often face challenges in creating an optimised schedule due to numerous constraints. Some of these constraints include presenter requests for specific presentation times, resolving conflicts between presenters, and addressing capacity issues. Additionally, the COVID-19 pandemic has led many conferences to transition to online or hybrid formats, which brings additional complexity due to different time zones involved. Therefore, an optimised conference schedule plays a crucial role in providing the best possible experience for every participant, allowing them to maximise the value of their investment.

The conference scheduling problem was introduced by Eglese and Rand [3] in 1987 and has been proved to be $\mathcal{NP}$-hard by Vangerven et al. [15]. Despite being introduced a long time ago, it is a problem that has not received much of attention from researchers compared to related problems, such as class and exam scheduling [11, 12]. Some related software tools exist which aid the scheduling process of conferences. However, such tools do not construct or optimise the conference schedule per se, but only provide additional information and recommendations to the organisers that are involved in the conference scheduling. One of these software is Confer [1], which is a paper-recommendation tool that collects attendees' preferences and constraints.

To the best of our knowledge, there are only 17 published studies tackling the conference scheduling problem. Each of these studies focuses on a specific conference and only few of them have been reported as extendable to other conferences. Such models require major adjustments and redesigns before their application to other conferences. This paper addresses this knowledge gap by introducing a model that can create optimised conference schedules, thereby enhancing the transparency of the scheduling process.

The scheduler is designed to tackle the challenges of organising the Genetic and Evolutionary Computation Conference (GECCO) in both in-person and hybrid formats. Additionally, it can be customised for application in other conferences. The proposed model allows the organisers to have a complete schedule that gives them the recommended timing for each talk. This reduces the time needed to organise the conference, as well as giving a better overall experience to the participants at the conference. Our model also allows

GECCO '24, July 14–18, 2024, Melbourne, Australia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

for the inclusion of room accessibility, unlike any other model we have seen, which should result in more accessible conferences.

2 PROBLEM DESCRIPTION

In this section, we describe the problem, which can be generalised to address various conference formats beyond GECCOs. This is because most conferences share a similar structure: presentations are organised into sessions consisting of typically 3-5 talks, each lasting around 10-25 minutes. These sessions are grouped into tracks, which may further be categorised into subject areas in some conferences. Typically, conference days are divided into predetermined time slots, during which sessions can be scheduled, with a maximum of around 5 sessions per day. The remaining time is allocated for activities such as keynote talks, meal breaks, and poster sessions.

GECCO 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 feasible schedule and minimise violations by assigning all tracks into sessions and rooms, and assigning all submissions into time slots. Based on the types of violations, a conference scheduling problem is approached by a Presenter-Based Perspective (PBP), or by an Attender-Based Perspective (ABP) [14]. A PBP approach aims to minimise violations associated with presenters, such as a request to present on a specific day or at a specific time. An ABP approach aims to minimise violations of attendees' preferences. These preferences include ensuring that attendees can participate in their favourite sessions, avoiding missing sessions due to space constraints, and preventing conflicts where attendees have to choose between two concurrent sessions of interest. Some studies (e.g., [9, 15]) have adopted a mixed approach by considering both presenters' and attendants' preferences. The study in [13] takes a mainly presenter-based approach, but also tries to allocate rooms based on the expected number of people coming, so it is also partly an attender based perspective approach.

Our work focuses on minimising both types of violations and we consider additional constraints which we summarise below:

(C1) All presentations must be scheduled; (C2) Ensure there are no scheduling conflicts; (C3) Number of talks in each session should not exceed the limit; (C4) Accommodate individuals who cannot attend specific sessions. On the occasion of a hybrid or online conference, we consider the time zone of presenters and schedule their submissions at suitable times. It is also applicable to in-person conferences to consider those who might experience jet lag; (C5) Accommodate individuals who cannot present in certain rooms for accessibility or facility issues; (C6) Certain tracks cannot be scheduled on specific sessions; (C7) Certain tracks cannot be scheduled in certain rooms. This stops a small track being allocated to a large room; (C8) Avoid scheduling same tracks in parallel; (C9) Avoid scheduling similar tracks at the same time; (C10) Accommodate requests from organisers to keep a room empty during a specified session; (C11) Minimise the number of rooms needed for each track, as utilising more rooms than necessary would not be convenient for participants; (C12) On the occasion of an organiser being responsible for more than one track or being a presenter on another track, we schedule such tracks within different sessions; (C13) In case of attendees who have declared attending preferences, we schedule such submissions in different sessions to avoid conflicts; (C14) Certain talks need to occur in a specific order, particularly for workshops; (C15) Some talks require multiple time slots, such as invited talks in workshops; and (C16) Schedule tracks in a consecutive manner to achieve a cohesive schedule.

Each of the above requests and attributes is considered as a soft constraint $SC = \{sc_1, sc_2, \ldots, sc_{16}\}$ and we allow organisers to assign a weight $w = \{w_{sc_1}, w_{sc_2}, \ldots, w_{sc_{16}}\}$ to each of them according to their subjective significance. A feasible solution is achieved by satisfying the hard constraint of *validity*, which ensures that at most one track is scheduled for each session and room, and at most one corresponding submission is scheduled per time slot. This ensures rooms are not double booked.

We now present our objective function which is a summation of the weighted soft constraints;

$$Min \sum_{i=1}^{16} w_{sc_i} \times V_{sc_i} \tag{1}$$

where w_{sc_i} indicates the corresponding weight of constraint SC_i, and V_{sc_i} is the corresponding violated amount of constraint SC_i.

3 METHODOLOGY

Previous studies have achieved quality solutions for a number of conferences using exact techniques [10], while others have employed heuristic methods like simulated annealing [11]. However, these studies only addressed a subset of the constraints we have outlined and often needed significant modifications to be applicable to other conferences. We propose a hyper-heuristic for our flexible approach to the conference scheduling problem. In contrast to other customised methods, hyper-heuristics offer problem domain independence, applicability of learning mechanisms, enhanced exploration of the solution space, and the ability to identify efficient problem-solving methods [7]. The concept of hyper-heuristics originated in the early 1960s [4]. Fisher and Thompson [4] stated that combining scheduling rules in production scheduling could lead to greater improvements than using them individually. This suggests that using a combination of low-level heuristics is likely to be more effective for solving our problem than relying on a single heuristic.

There are two main types of hyper-heuristics [2]: methodologies for *selecting* existing heuristics and methodologies for *generating* new heuristics. In this paper, we focus on selection hyper-heuristics which typically operate as follows: We begin with an initial solution and iteratively improve it through consecutive stages until a termination criterion is met. The initial solution may be randomly generated and is often far from optimal. The two stages are *heuristic selection* and *move acceptance*. In heuristic selection, a heuristic is chosen from a set of predefined low-level heuristics which generates a new solution. Move acceptance then determines whether to accept the new solution. In selection hyper-heuristics, low-level heuristics can include mutational and hill climbing strategies. Mutational heuristics generate solutions mostly at random, allowing the search process to explore different regions. On the other hand, hill climbing heuristics always produce non-worsening solutions, helping to improve upon the current solution. By using a combination of hill climbing and mutational heuristics, we can achieve a balance between exploration and exploitation.

We utilise a selection hyper-heuristic that employs a random selection method with improve or equal move acceptance which only accepts moves that do not worsen the current solution. We focus on employing simple low-level heuristics, including: (1) swapping a random talk with another randomly chosen talk from the same track, (2) exchanging a track with another randomly chosen track, and (3) inserting a random talk into a different position within the same track. Each low-level heuristic has an equal probability of selection, with a probability of 1/3. To facilitate the exploration of solutions, we incorporate a random shuffling of the current solution at regular intervals during the algorithm's execution. Following a preliminary experiment, we determined that shuffling every 30 minutes balances exploration and exploitation. Consequently, our approach involves running the hyper-heuristic algorithm, supporting exploitation, followed by shuffling the solution, supporting exploration. This process repeats iteratively until a termination criterion is met, with the hyper-heuristic running for another 30 minutes after each shuffle, ensuring a continuous exploration and exploitation of solutions.

Due to the nature of our problem, which involves creating a timetable for every time slot rather than for every session, our approach is considerably more computationally intensive compared to studies such as [13]. In [13], they focus on creating a high-level schedule. This means that their method focuses on the time and room each track is allocated to. They leave the low-level scheduling to each track organiser.

4 COMPUTATIONAL RESULTS

The proposed algorithm has been utilised to create the schedules of the GECCO conference from 2020 to 2023, and it will also be utilised for creating the schedule of GECCO2024. The model was tested on data from GECCO2019, showing promising results compared to the schedule generated by the organisers of GECCO2019. Additionally, the same algorithm has been tested on solving scheduling problems for other conferences, such as OR60 and N2OR. In this section, we present the results from GECCO2019 and the recent GECCO2023.

4.1 Problem settings for GECCO2023

GECCO2023 was hosted at the Altis Grand Hotel, an iconic venue in Lisbon known for hosting numerous significant events over the decades. The conference took place from July 15 to July 19, 2023, in a hybrid mode. The first two days (Saturday and Sunday) were dedicated to Workshops, Tutorials, Competitions, and Women+@GECCO. The subsequent three days (Monday to Wednesday) were allocated for Opening and Closing ceremonies, Main Tracks, Keynotes, Poster Sessions, ECiP (Evolutionary Computation in Practice), HOP (Hot Off the Press), HUMIES (Human-Competitive Awards) and Job Market. The Poster Sessions encompassed various submissions, including poster submissions, Late-Breaking Abstracts (LBAs), submissions from the student workshop and competitions.

There were a total of 8 rooms distributed across four different floors at the hotel. On the ground floor, there was 1 room available with a capacity of 90. Moving up to the first floor, there were 3 rooms, each varying in capacity between 50 and 150 attendees. Floor 12 hosted another single room with a capacity of 70. Continuing upwards to the 13th floor, there were 3 more rooms, with capacities ranging between 60 and 150 attendees. Additionally, the Plenary Room was situated on floor -1 and was exclusively available from Monday to Wednesday for events such as Invited Keynotes.

Some activities, such as Women+@GECCO, opening and closing sessions, Invited Keynotes, HUMIES, Job Market, breaks, social dinner, and Poster Sessions, were prearranged and did not need to be scheduled using the solver.

During Saturday and Sunday, a total of 28 sessions were allocated for workshop tracks, with each workshop spanning between 0.5 to 3 sessions. Additionally, there were two additional sessions reserved for the student workshop. Detailed information for each workshop, including the mode of attendance (online vs in-person) for speakers and organisers, time zones for online presentations, constraints for workshops and individual talks, as well as the desired order and duration of talks, has been provided. For tutorial tracks, a total of 32 sessions were available. Additionally, there were two sessions designated for competition tracks.

During Monday to Wednesday, approximately 192 time slots were available for talks. This calculation was based on having 6 sessions spread across 3 days, with 8 rooms available, considering 4 talks per session. Out of a total of 209 submissions, 29 were designated as HOP submissions. Four sessions were specifically allocated for HOP, with 7-8 time slots available per session. There were a total of 26 Best Papers (BPs) to be scheduled. **Altogether, the solver needed to schedule a total of 487 slots.**

Table 1 provides a breakdown of the number of submissions for each track.

Table 1: Conference track summary

ACO-SI	11	ENUM	6	SBSE	4
ACO-SI BP	2	ENUM BP	3	SBSE-NE BP	2
CS	10	GA	9	Theory	10
CS BP + Impact	3	GA BP	2	Theory-GECH BP	2
ECOM	16	GECH	6	HOP	29
ECOM BP	2	GP	13	ECiP	5
EML	23	GP BP	3	Introductory Tutorials	15
EML BP	3	NE	8	Advanced Tutorials	8
EMO	17	RWA	21	Specialized Tutorials	7
EMO BP	3	RWA BP	2	WS - IAM	10
WS - BBOB	10	WS - LAHS	12	WS - iGECCO	5
WS - BENCH	6	WS - NEWK	7	WS - Student	15
WS - EC+DM	8	WS - QD-Benchmarks	5	WS - AABOH	6
WS - ECADA-KL	16	WS - QuantOpt	17	WS - GGP	7
WS - EGML-EC	4	WS - SAEOpt-GEWS	10	WS - SymReg	7
WS - ERBML	6	WS - SBOX-COST	7	WS - EvoSoft	8
WS - EvoRL	4	WS - SWINGA	7	WS - ECXAI	6
WS - SAEOpt	7				

4.2 GECCO2023 scheduling considerations

We describe the scheduling considerations for GECCO2023. For each consideration below, we show, within brackets, the corresponding constraint (outlined in section 2) used to model the consideration and the weight assigned. The higher the weight, the more important the constraint is:

(1) There are 42 submissions with unavailability constraints (C4 100); (2) Each submission may potentially clash with 0-10 other submissions (C2 10000); (3) The order of talks must be adhered to (C14 10000); (4) BP sessions are not to be scheduled on Wednesday

to allow organisers time to prepare certificates for the winners (C6 1000000); (5) BP sessions should be allocated to larger rooms whenever possible (C7 10000); (6) Avoid scheduling three or more BP sessions simultaneously (C7 10000); (7) ECiP session is to be scheduled on Monday morning (C6 1000000); (8) ECiP and HOP sessions can take place in the Plenary Room (C7 10000); (9) The Plenary Room must not be used during the Job Market session (C10 10000); (10) Various speaker requests to be considered, including scheduling talks consecutively in a specific order (C14 10000), avoiding certain time slots due to teaching commitments, early departures, late arrivals, and other limitations (C4 100 & C5 100); (11) Requests from workshop organisers, such as not scheduling certain workshop or tutorial tracks simultaneously, must also be accommodated (C9 10000); (12) Each session in the Main Track sessions lasts 1 hour and 30 minutes, with each submission allocated 20 minutes (C3 10000000); (13) There is flexibility to extend a few sessions in the Main Track to be longer (i.e., 5 time slots), but most sessions should have 4 time slots (C3 10000000); (14) Sessions scheduled on the upper floors must begin 10 minutes later to allow participants ample time to reach the rooms, considering the limited capacity of the elevators. To address this, we incorporated a dummy submission titled "time needed for attendees to use the elevator or the stairs". This applies to all Main Track Sessions (Mon-Tue) with four talks or less (C1 10000000); (15) During room allocation, consideration needs to be given to the number of accepted submissions for each track and reports from previous GECCOs to estimate attendance, influencing the room allocation decisions (C7 10000); (16) Due to the high number of submissions and expected attendance in the EML track, priority was given to assigning EML (including its BP session) to the largest room (C7 10000); (17) The plan was initially to schedule presentations irrespective of onsite or online formats. However, we have grouped them together to facilitate a smoother transition between the two modes, aiming to minimise potential challenges or disruptions when switching between onsite and online presentations (C14 10000); (18) Avoid simultaneous session endings before lunch on Tuesday. This challenge stems from the large number of attendees participating in-person, where it is undesirable for everyone to take their lunch break simultaneously. Therefore, we have to implement staggered ending times for sessions to address this issue (C3 10000000); (19) Considering historical practices, it has been customary to schedule Introductory Tutorials on Saturday and Specialised Tutorials on Sunday (C16 10000); (20) Avoid scheduling the same workshop and tutorial tracks as the first session if they were scheduled first the previous year (C4 1000000).

For the remaining constraints. C11 was assigned a weight of 10000 for the workshop days and 10 otherwise. Constraints C8, C12, and C15 were given the following weights: 1, 100, and 10000000 respectively.

4.3 Time zones

There were 31 submissions across Workshops and Tutorials, and another 25 submissions across Main Tracks and HOP which needed to be scheduled online considering their time zone differences. Table 2 and Table 3 present the distribution of submissions based on different time zones for online submissions. Our priority is to allocate these submissions to sessions marked in green as the top choice, then to sessions marked in amber as the next preference. The third choice, indicated in red, should be avoided whenever possible by the scheduler.

Table 2: Time zone availability (Saturday-Sunday)

Time zone	#	08:30-10:20	10:40-12:30	14:00-15:50	16:10-18:00
America/Los Angeles (GMT -07:00)	2	00:30-02:20	02:40-04:30	06:00-07:50	08:10-10:00
America/New York (GMT -04:00)	4	03:30-05:20	05:40-07:30	09:00-10:50	11:10-13:00
America/Sao Paulo (GMT -03:00)	1	04:30-06:20	06:40-08:30	10:00-11:50	12:10-14:00
Asia/Hong Kong (GMT +08:00)	1	15:30-17:20	17:40-19:30	21:00-22:50	23:10-01:00
Asia/Kolkata (GMT +05:30)	3	12:30-14:20	14:40-16:30	18:00-19:50	20:10-22:00
Asia/Shanghai (GMT +08:00)	2	15:30-17:20	17:40-19:30	21:00-22:50	23:10-01:00
Asia/Tokyo (GMT +09:00)	2	16:30-18:20	18:40-20:30	22:00-23:50	00:10-02:00
Australia/Perth (GMT +08:00)	1	15:30-17:20	17:40-19:30	21:00-22:50	23:10-01:00
Europe/Kiel (GMT +02:00)	1	09:30-11:20	11:40-13:30	15:00-16:50	17:10-19:00
Europe/London (GMT +01:00)	5	08:30-10:20	10:40-12:30	14:00-15:50	16:10-18:00
Europe/Moscow (GMT+03:00)	2	10:30-12:20	12:40-14:30	16:00-17:50	18:10-20:00
Europe/Stockholm (GMT +02:00)	2	09:30-11:20	11:40-13:30	15:00-16:50	17:10-19:00
GMT +01:00	1	08:30-10:20	10:40-12:30	14:00-15:50	16:10-18:00
GMT +02:00	1	09:30-11:20	11:40-13:30	15:00-16:50	17:10-19:00
GMT +10:00	1	17:30-19:20	19:40-21:30	23:00-00:50	01:10-03:00
GMT -04:00	1	03:30-05:20	05:40-07:30	09:00-10:50	11:10-13:00
GMT -08:00	1	23:30-01:20	01:40-03:30	05:00-06:50	07:10-09:00

Table 3: Time zone availability (Monday-Wednesday)

Time zone	#	11:00-12:30	14:30-16:00	16:30-18:00	11:00-12:30	14:30-16:00	09:00-10:30
America/Chicago GMT-05:00	1	05:00-06:30	08:30-10:00	10:30-12:00	05:00-06:30	08:30-10:00	03:00-04:30
America/Denver GMT-06:00	1	04:00-05:30	07:30-09:00	09:30-11:00	04:00-05:30	07:30-09:00	02:00-03:30
America/Los Angeles GMT-07:00	1	03:00-04:30	06:30-08:00	08:30-10:00	03:00-04:30	06:30-08:00	01:00-02:30
America/New York GMT-04:00	3	06:00-07:30	09:30-11:00	11:30-13:00	06:00-07:30	09:30-11:00	04:00-05:30
America/Sao Paulo GMT-03:00	1	07:00-08:30	10:30-12:00	12:30-14:00	07:00-08:30	10:30-12:00	05:00-06:30
America/Vermont GMT-05:00	1	05:00-06:30	08:30-10:00	10:30-12:00	05:00-06:30	08:30-10:00	03:00-04:30
Asia/Shanghai GMT+08:00	2	18:00-19:30	21:30-23:00	23:30-01:00	18:00-19:30	21:30-23:00	16:00-17:30
Europe/London GMT+01:00	2	11:00-12:30	14:30-16:00	16:30-18:00	11:00-12:30	14:30-16:00	09:00-10:30
Europe/Stockholm GMT+02:00	4	12:00-13:30	15:30-17:00	17:30-19:00	12:00-13:30	15:30-17:00	10:00-11:30
GMT+01:00	2	11:00-12:30	14:30-16:00	16:30-18:00	11:00-12:30	14:30-16:00	09:00-10:30
GMT+02:00	3	12:00-13:30	15:30-17:00	17:30-19:00	12:00-13:30	15:30-17:00	10:00-11:30
GMT+05:30	1	15:00-16:30	18:30-20:00	20:30-22:00	15:00-16:30	18:30-20:00	13:00-14:30
GMT+08:00	1	18:00-19:30	21:30-23:00	23:30-01:00	18:00-19:30	21:30-23:00	16:00-17:30
GMT-04:00	2	06:00-07:30	09:30-11:00	11:30-13:00	06:00-07:30	09:30-11:00	04:00-05:30

4.4 Results

The algorithm was executed using VBA within Excel. The computational process took place on a system powered by an AMD Ryzen 7 5700U processor with Radeon Graphics running at 1.80GHz. The system was equipped with 16.0GB of RAM. The operating system used was Windows 11 Home. The algorithm ran for a duration of 4 hours, resulting in 8 intervals as we shuffle the solution every 30 minutes. Table 4 presents the results for GECCO2019, GECCO2023 Workshop (schedule for the first two days of the conference), and the GECCO2023 Main (schedule for the last three days). The table displays the objective of the initial solution, generated randomly, the number of iterations/moves and the corresponding objective value achieved for each interval.

The results presented in the table demonstrate that the selection hyper-heuristic effectively refined the poor initially generated solutions, leading to substantial improvements. However, the observed improvements between intervals were relatively modest, indicating potential opportunities for exploration of alternative strategies to achieve more significant enhancements in solution quality. The algorithm performed approximately 2.5 million iterations for both GECCO2019 and GECCO2023 Main, while the number of iterations for GECCO2023 Workshop was around 1.3 million at each interval.

Table 4: Objective of the initial solution, objectives achieved at each interval, and the number of iterations (in brackets) while solving the problem instances

	GECCO2019	GECCO2023 Workshop	GECCO2023 Main
Initial Solution	34420681	1317411	200721
Interval 1	59 (2592130)	1100 (1360013)	632 (2403019)
Interval 2	58 (2597282)	1100 (1319465)	632 (2397676)
Interval 3	56 (2596836)	1100 (1341150)	632 (2571085)
Interval 4	56 (2597326)	1100 (1335878)	632 (2620305)
Interval 5	56 (2599057)	1100 (1340014)	632 (2604743)
Interval 6	56 (2552030)	1100 (1336409)	632 (2583656)
Interval 7	56 (2437472)	900 (1344850)	632 (2568762)
Interval 8	56 (2388709)	900 (1329266)	632 (2600064)

GECCO2019

We conduct a comparison between the solution attained by our solver and the schedule manually constructed by the organisers of GECCO2019. Our hyper-heuristic solver identified 14 violations, whereas the manual schedule contained 25 violations. Both solutions successfully scheduled all submissions without conflicts. However, it is important to note that neither solution was evaluated based on participants' preferences, as we lacked access to this information. It was observed that both the solver and the manual solution had one track running in parallel. In our solver, the number of rooms allocated per track was violated twice, whereas in the manual solution, it was violated six times. The constraints regarding the usage of rooms by certain tracks were violated twice in the solver, whereas in the manual solution, it was violated four times. The constraints concerning the scheduling of tracks in a consecutive manner were violated 11 times in the solver, while in the manual solution, it was violated 13 times.

The solver consistently outperformed the manual solution, indicating its effectiveness in generating schedules with fewer constraint violations. The solver achieved better results with less effort. Since GECCO2019 was an in-person event, it did not exhibit the challenges associated with scheduling hybrid conferences. Nevertheless, the results demonstrated the effectiveness of the solver as a proof of concept, showcasing its potential to efficiently generate schedules for complex events. The success of the solver in optimising scheduling for GECCO2019 instilled confidence in its capabilities, leading to its adoption for subsequent conferences such as GECCO2020 through GECCO2023. Given its successful track record, the solver will be used for GECCO2024, confirming its role as a valuable tool for managing conference scheduling effectively.

GECCO2023

The following constraint violations are reported: (1) Three main tracks were not scheduled in consecutive order. This constraint however was given low priority for the main tracks, but it was considered a high priority for the schedule on Saturday and Sunday; (2) Seven talks were scheduled on days they specified as unavailable. Further examination revealed that these talks had indicated unavailability throughout the entire days; (3) The Theory track was allocated to two different rooms; (4) The GECH track was assigned to a room with limited seating capacity for its size. Additionally, the tracks ACO-SI and CS were allocated to a room with somewhat limited seating capacity for tracks of their size; (5) Two online talks were not scheduled during their respective most preferred time slots based on their time zones. One was scheduled from 06:00 to 07:30 (amber time), and another was scheduled from 21:30 to 23:00 (amber time); (6) An online workshop submission was scheduled from 22:00-23:50, considering the speaker's time zone the optimal slots would be 16:30-18:20 or 18:40-20:30. However, scheduling the talk in any of these ideal slots would necessitate scheduling another online talk in the same workshop for either 03:30-05:20 or 05:40-07:30, which is less than ideal; (7) An introductory tutorial is scheduled for the first time slot on Sunday rather than Saturday. However, the speaker contacted us and requested that their tutorial be scheduled for Sunday. Furthermore, a specialised tutorial is scheduled for the last time slot on Saturday. However, the speakers have requested for the tutorial to be scheduled for Saturday; and (8) The final constraint violation arises from conflicting availability preferences of authors within the same session. Specifically, one author for an onsite submission in a particular workshop expressed unavailability on Saturday, yet the session was scheduled for that day. Conversely, another author from a different submission in the same session is only available on Saturday. This creates a dilemma where satisfying the constraint for one submission would inevitably violate the constraint for the other. All other constraints were successfully met or satisfied.

5 CONCLUSION

In this paper, we have explored the application of a hyper-heuristic approach for creating conference schedules, including those for GECCO. This method aims to provide organisers with a comprehensive schedule that gives them the recommended timing for each activity. The solver creates a high-level schedule, which focuses on the time and room each track is allocated to, and a low-level schedule, to assign the individual talks to time slots, while taking into account considerations such as presenter preferences, attendee preferences, room capacity, room accessibility, session hopping (i.e., minimising travel between rooms), and more. The model promotes inclusive, accessible, and sustainable events. Examples include: scheduling specific talks in specific rooms which supports advanced accessibility; minimising the changeover times for each track for conferences that involve sessions in different locations; offering flexible options for participants with caring responsibilities or if they have any special requirements, such as religious needs, to attend all or part of the conference. By employing a weighted sum approach, we can achieve close-to-optimal solutions for our scheduling problem. The scheduler could help ensure that conferences run smoothly. Ultimately, we hope that this tool will enable academics to maximise their conference experience and alleviate the scheduling burden on organisers.

5.1 Future work

There are several possible research directions to take this work in. One issue of the proposed model is that if a change needs to be made, it may require generating an entirely new schedule. However, if conference organisers identify such changes before distributing the timetable to participants, this issue can be mitigated. Our algorithm should be efficient enough to quickly generate a new schedule, allowing for flexibility in responding to changes. The main issue arises when the timetable has already been distributed to the participants. If we rerun the algorithm with new changes, we risk generating a completely different schedule from the current one. This could pose a significant problem if, for example, we change the time of a track to a different day, causing participants who have already booked their travel to be unable to attend the new session. A solution to this may be to employ a minimal perturbation approach [8]. The minimal perturbation problem incorporates new changes with the current solution to generate a new problem, aiming to produce a solution that is as close to the current solution as possible. This approach minimises disruptions to conference participants. In GECCO2023, the schedule has undergone several last-minute changes, but these modifications were made manually after carefully reviewing and addressing the violated constraints.

Another potential direction is to explore multi-objective approaches other than the weighted sum approach. One such approach is the lexicographic approach, where different objectives are categorised into an order of priority levels. Objectives at lower levels are considered infinitely more important than those at higher levels. Consequently, the algorithm seeks the optimal solution for objectives at lower levels before addressing those at higher levels. Some researchers argue that this method is highly practical [6], while others criticise it [16]. In the weighted sum approach, our set of objectives is combined into a single objective by multiplying each objective by a user-supplied weight. This method is widely used, but determining the appropriate weights for each objective can be challenging. Typically, weights are assigned in proportion to the relative importance of each objective in the problem. However, one of the main challenges with the weighted sum method arises in non-convex multi-objective problems, where certain Pareto-optimal solutions may be missed. Consequently, further research could explore alternative methods that may perform better in such cases.

ACKNOWLEDGMENTS

The authors express their gratitude for the significant contributions made by all the organisers of GECCO from 2019 to 2023. Special thanks are extended to Arnaud Liefooghe, the Hybrid Scheduling Co-Chair of GECCO2023. This work was supported by the UK Research and Innovation under Grant EP/V520214/1.

REFERENCES

[1] Anant Bhardwaj, Juho Kim, Steven Dow, David Karger, Sam Madden, Rob Miller, and Haoqi Zhang. 2014. Attendee-Sourcing: Exploring The Design Space of Community-Informed Conference Scheduling. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 2, 1 (Sep. 2014), 2–10. https://doi.org/10.1609/hcomp.v2i1.13163

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

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

[4] H. Fisher and G. L. Thompson. 1963. Probabilistic learning combinations of local job-shop scheduling rules. In *Industrial Scheduling*, J. F. Muth and G. L. Thompson (Eds.). Prentice-Hall, Inc, New Jersey, 225–251.

[5] P Greenstreet. 2020. *A new hyper-heuristic with weight sum approach to conference scheduling problem.* Master's thesis. Lancaster University.

[6] James P Ignizio. 1985. *Introduction to linear goal programming.* Sage Beverly Hills, CA.

[7] Ahmed Kheiri. 2020. Heuristic Sequence Selection for Inventory Routing Problem. *Transportation Science* 54, 2 (2020), 302–312.

[8] Antony E Phillips, Cameron G Walker, Matthias Ehrgott, and David M Ryan. 2017. Integer programming for minimal perturbation problems in university course timetabling. *Annals of Operations Research* 252 (2017), 283–304.

[9] Yaroslav Pylyavskyy, Peter Jacko, and Ahmed Kheiri. 2024. A generic approach to conference scheduling with integer programming. *European Journal of Operational Research* (2024). https://doi.org/10.1016/j.ejor.2024.04.001

[10] Jeffrey Quesnelle and Daniel Steffy. 2015. Scheduling a conference to minimize attendee preference conflicts. In *Proceedings of the 7th multidisciplinary international conference on scheduling: theory and applications.* MISTA, Prague, 379–392.

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

[12] Scott E. Sampson and Elliott N. Weiss. 1996. Designing conferences to improve resource utilization and participant satisfaction. *Journal of the Operational Research Society* 47, 2 (1996), 297–314.

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

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

[15] Bart Vangerven, Annette M.C. Ficker, Dries R. Goossens, Ward Passchyn, Frits C.R. Spieksma, and Gerhard J. Woeginger. 2018. Conference scheduling — A personalized approach. *Omega* 81 (2018), 38–47.

[16] Milan Zeleny. 1973. *Compromise programming.* Multiple Criteria Decision Making, University of South Carolina Press, Columbia. 262–301 pages.