Information Systems 139 (2026) 102700

Contents lists available at ScienceDirect

Information Systems

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

Multi-sided fairness in sequential task assignment

Anna Dalla Vecchia [a],*, Sara Migliorini [a], Elisa Quintarelli [a], Kostas Stefanidis [b]

[a] Computer Science Department, University of Verona, Italy
[b] Data Science Research Centre, Tampere University, Finland

ARTICLE INFO

Keywords:
Sequential tasks
Multi-stakeholders fairness
Local fairness
Global fairness

ABSTRACT

Sequential task assignment is a crucial process in many contexts, where resource allocation over time is a key step to consider and often involves groups of people with diverse objectives, preferences, and constraints. Fairness in these scenarios is paramount, as it implies efficiency and satisfaction while also impacting performance. Although the definition of fairness depends on the context and domain, it generally ensures an equal distribution of tasks among participants, subject to certain constraints and guidelines. Moreover, it mitigates biases and disparities, promoting inclusivity and diversity within teams. In this paper, we highlight the different aspects of fairness in sequential task assignments and emphasize that the perspectives of various stakeholders must be considered. As motivating examples, we concentrate on two scenarios: (a) the timetable creation problem in the university domain, showing that the notion of fairness must be considered from both the students' and professors' points of view, and (b) the tourism traveling planning, where the perspectives of tour guides and tourists are taken into account during a planning process. We propose a generic formalization of the problem that an optimization algorithm can easily manage. The aim is to find and compare the fairness of different stakeholders and evaluate whether a fair solution for one of them can be fair for another with different constraints and preferences. We introduce the notion of local and global fairness to highlight that an optimal solution for one stakeholder does not necessarily mean it is optimal also for others, and some compromises need to be identified. Finally, we explore how global fairness can be achieved by integrating multiple solutions, each aligned with a local fairness perspective.

1. Introduction

Assigning tasks effectively is crucial in many contexts, as it can significantly impact the creation or destruction of a harmonious social environment. Consequently, considering fairness as a guiding ethical principle is a strategic approach to cultivating a high-performing and positive environment. Moreover, fair task assignment has a significant impact on various aspects of data beyond just the assignment process itself. For example, distributing tasks over time to groups of people within a company enables the establishment of long-term relationships, satisfaction, and the achievement of predefined goals, provided that everyone has an equal opportunity to showcase their skills and talents. In addition, achieving fairness for interacting groups of people reduces the perpetuation of stereotypes and guarantees equal opportunities for various stakeholders [1].

Fairness is not an easily and universally defined concept, and its meaning can vary across cultures, societies, and contexts. In this paper, we discuss how universal principles may underlie fairness, but the exact understanding and implementation can depend on different stakeholders' points of view and must be considered a multifaceted process [2].

Among all the possible scenarios, this paper concentrates on the problem of assigning sequential tasks to different stakeholders. This choice is justified by both the relevance of the problem in real-life situations and the complexity of managing multiple stakeholders competing for the same set of resources in a lifespan that spans the space of a single activity. In the following, the term *stakeholder* will be used to denote any individual or group of people that can be affected by or can affect the task assignment and have different preferences and constraints. Note that even when a single stakeholder represents a group of people, it is intended to be treated as an atomic entity with respect to the tasks being considered. In this scenario, achieving fairness involves finding the right balance between meeting the needs and expectations of various stakeholders. We call a solution *locally fair* if it is fair for a single stakeholder. Once a set of local fair solutions has been obtained, one for each stakeholder, we will need an algorithm to merge them while balancing the previously achieved local fairness, with the

* Corresponding author.
E-mail addresses: anna.dallavecchia@univr.it (A. Dalla Vecchia), sara.migliorini@univr.it (S. Migliorini), elisa.quintarelli@univr.it (E. Quintarelli), konstantinos.stefanidis@tuni.fi (K. Stefanidis).

https://doi.org/10.1016/j.is.2026.102700
Received 12 March 2025; Received in revised form 24 December 2025; Accepted 15 February 2026
Available online 16 February 2026

ultimate aim of reaching what we call *global fairness*. Notice that there is no guarantee that a perfect, fair model for all stakeholders can be achieved, namely that an optimal global solution satisfying all the preferences of all stakeholders can be reached, and some compromises need to be identified to achieve a global fairness solution satisfying all stakeholders in a balanced manner.

This work examines two case studies: one in the university domain and the other in the tourism sector. We analyze the local fairness of the two main stakeholders involved in each scenario, namely professors and students for the former and tour guides and tourists for the latter. The experimental process in two real-world domains demonstrates the practical utility and robustness of the approach and underlines that achieving a local fair solution for each stakeholder is not enough to build a fair global solution, but is a useful step to achieve it.

Given all these premises, the main contributions of this work can be summarized as follows:

- We formalize the concept of local and global fair solutions for sequential task assignments when two temporal dimensions are involved, i.e., a sequence of activities forming a task and a sequence of tasks.
- We integrate into the fairness formalization the possibility for each stakeholder to express personalized soft constraints with different levels of severity without the need to express their motivations.
- We develop a procedure for reaching local fairness in task assignment as a multi-objective optimization problem. In particular, we exploit the dominance-based Multi-Objective Simulated Annealing (MOSA) technique and propose an extension that utilizes the concept of fairness to enhance convergence speed in terms of the number of iterations. Such an extension introduces the concept of fairness into the definition of the next perturbation and can be successfully integrated into other MOSA variants, such as AMOSA [3], to accelerate convergence.
- We discuss how local fairness can clash with global fairness in a multi-stakeholder environment and evaluate the approach for finding a unique global fair solution.
- We test our proposal on two real-world domains: the university timetable construction and the tourism travel planning. For each use case, we evaluate the accuracy of the result and the performance of the approach, as well as the proposed optimization extension.

The remainder of the paper is organized as follows. Section 2 explores related work, while Section 3 outlines the formalization of the problem. Section 4 presents two case studies, and Section 5 illustrates our algorithm for achieving local fairness. In Section 6, we report the evaluation of the proposed approach in the two considered scenarios. Section 7 evaluates the degree of global fairness achieved by the found local fair solutions and discusses how to use the approach to ensure also global fairness. Finally, Section 8 summarizes the main contributions and outlines future work.

2. Related work

In this section, we review the main concepts underlying our framework, including the foundations of resource allocation and scheduling, fairness in resource allocation, and fairness in sequential and multi-stakeholder task assignment, as well as the principal proposals related to the scenarios considered in our running examples. We conclude by discussing how our proposal relates to the state of the art.

2.1. Resource allocation and scheduling foundations

Sequential task assignment builds on a rich literature on resource allocation, where tasks must be scheduled across constrained temporal and resource dimensions [4]. A foundational treatment is provided by classical scheduling theory, which formalizes problems such as the Resource-Constrained Project Scheduling Problem and Job-Shop Scheduling, both of which integrate resource requirements and temporal dependencies in unified optimization frameworks [5]. When decisions about the use of resources must be made across a sequence of discrete time periods, as is typical in many real-world applications, the problem naturally specializes into discrete-time scheduling, where tasks are assigned to time slots while satisfying exclusivity, availability, and precedence constraints [6].

Several application domains naturally fit this class of problems. University course timetabling, for example, is a well-established instance of this paradigm, in which lectures must be assigned to rooms, instructors, and student groups across a weekly allocation composed of discrete time periods [7]. Similarly, tourism trip planning can be modeled as a discrete-time resource-allocation problem in which activities, services, and transportation options must be scheduled over time while respecting opening hours, capacity limits, and visitor preferences [8]. These applications demonstrate how diverse domains can be unified through the common theoretical framework of discrete-time resource scheduling.

2.2. Fairness in resource allocation and classification settings

While traditional scheduling research rarely considered fairness explicitly, recent efforts across machine learning and decision support have increasingly integrated fairness notions into resource allocation processes. In these solutions, fairness is often conceptualized as a local resource-allocation problem, where the objective is to distribute limited resources efficiently while satisfying predefined fairness constraints. For example, statistical parity [9] requires that the protected groups (e.g., gender groups) experience equal selection rates, whereas equalized odds [10] requires that those groups experience equal error rates. Numerous techniques can be applied to mitigate bias in data, algorithms, or the output [11].

When considering fairness in team formation contexts, [12] formulates the problem as a fair allocation task aiming to assign students to projects in a way that balances workloads and tasks across the resulting teams. Another example is [13], which examines fair team formation in an online labor marketplace. More recently, [14] explores a problem where teams have multidisciplinary requirements, and the incremental selection of team members is based on the match of their skills and the requirements. For assembling multiple teams and fairly allocating the best members among them, a heuristic incremental method is suggested to create team recommendations for multidisciplinary projects. Similarly, [15] formulates team assembly as a multi-objective optimization task where fairness-aware team assembly involves considering several competing objectives. The multi-objective optimization task is also addressed in [1], in which the authors introduce a sequence of recommendations for groups of users using an algorithm based on the Multi-Objective Simulated Annealing (MOSA) [16] technique. Here, the preferences of the individual users are combined to provide the best possible experience. The literature also includes works that propose methods for recommending sequential activities to groups of users, but these approaches focus on optimizing a unique utility function [17–20].

Fairness in task assignment is also explored in [21], where the problem is framed as a coalition-based task allocation. Workers collaborate to complete spatial tasks (e.g., home improvement and furniture installation) with the goal of maximizing the overall rewards for all participants. However, the objective of coalition-based task assignments is to allocate stable worker coalitions to tasks for maximizing rewards, which differs from the problem faced in this paper. Here, each user has a predefined set of mandatory tasks (e.g., students attending class and professors teaching them), and we aim to determine the optimal sequence of class allocations in a timeline that satisfies these predefined groups. For this reason, even the solutions proposed by [22,23] for a fair task allocation do not apply to our problem due to differing requirements; indeed, they do not accommodate sequential task allocation. Furthermore, this article delves into a deeper discussion of the meaning of fairness constraints and preferences that have not yet been explored in the existing literature.

2.3. Fairness in sequential and multi-stakeholder task assignment

Moving from single-stakeholder fairness to settings with multiple interacting stakeholders introduces a second layer of complexity. Sequential assignments create temporal dependencies, while differing stakeholder preferences require reconciliation across heterogeneous constraints. In recommendation systems, multi-stakeholder fairness has been explored in fields such as tourism and group decision-making [24–26], where balancing the satisfaction of different stakeholders (e.g., tourists and providers) requires tailored multi-objective formulations. Related work on multi-objective fairness in recommendation and decision-support systems [27–29] shows that improving fairness for one stakeholder group can come at the expense of another, motivating approaches that explicitly expose and manage such trade-offs.

2.4. Academic timetabling and tourism planning

Regarding the fair academic timetabling problem, at the best of our knowledge, existing literature primarily focuses on optimizing the course allocation without addressing the fairness perspective of the two groups of stakeholders involved [30–33]. In more detail, the work closest to ours is [30], where the authors propose a solution using simulated annealing techniques. Their approach determines the optimal timetable by minimizing the total penalty, but it does not explicitly incorporate the concept of fairness. Similarly, the authors in [31] frame the problem as a multi-objective problem, acknowledging only the students' needs. In contrast, [32] applies a genetic algorithm-based methodology; however, a key limitation is that the authors do not consider any form of personalization of the constraints specified by the stakeholders.

As the authors highlight in [34], fairness needs to incorporate multiple viewpoints. Defining a uniform concept of fairness for all users and groups is restrictive. For this reason, in [35], the authors propose a new framework to provide recommendations considering different fairness concepts. However, not all items need to be assigned in their case. On the other hand, in [36], the investigation of fair recommendations in two-sided online platforms is addressed as a constrained version of the fairly allocating invisible goods problem. However, the main limitation of the approach is that fairness is a static notion. The two-sided fairness problem is also addressed in [37] by observing the inequality in the distribution of drivers' income on hailing platforms.

2.5. Differences to prior works

Our work builds on the observation that improving fairness for one stakeholder group may come at the expense of another, but differs from existing approaches by formalizing fairness within a unified optimization framework that captures stakeholder-specific soft constraints. Unlike prior multi-stakeholder optimization models, our formulation explicitly accounts for the temporal structure of the problem, in which tasks unfold sequentially over multiple days and decisions made at one step influence or constrain subsequent assignments.

Furthermore, in contrast to previous works, fairness in our study is defined by personalizing a set of soft constraints. This allows for greater flexibility, as the constraints are expressed without the need for additional sensitive information, using negative numbers in undesired time slots. We remark that our approach is general enough to be applied to any scenario where two distinct stakeholder groups, suppliers and customers, each with different needs and competing for common resources, coexist. These perspectives must be considered since they may influence other factors, such as user satisfaction and the potential to improve the target goal (e.g., graduation or completion of a tourist trip). Concerning the proposed algorithm, the sequential assignment of resources is distributed across two orthogonal temporal domains, and we introduce a fair-aware modification of the MOSA algorithm, enabling the solution to be reached with fewer iterations.

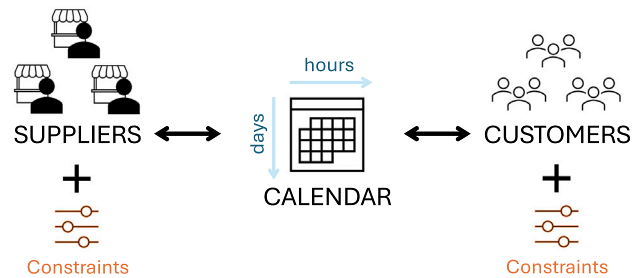


Fig. 1. Schematic representation of the considered task assignment problem.

3. Local and global fairness in sequential task assignment

In this section, we formalize the concept of fairness in sequential task assignments. Fig. 1 represents our scenario, where some individual users are suppliers of a specific service, and groups of users benefit from the service. Both sides have different constraints and preferences regarding scheduling the service provided/used on a horizontal temporal dimension (e.g., within a day) and a vertical temporal dimension (e.g., during consecutive days in a week or a month). Note that the granularity of horizontal and vertical temporal dimensions can be decided based on the considered scenario, and a group of customers can also be composed of a single user without modifying our formalization. The main aim of our contribution is to recommend a solution for scheduling the service that is as fair as possible for all stakeholders.

Following the classification introduced in [33,38], we start by distinguishing between two main kinds of constraints involved in fairness management.

- **Hard constraints** $\mathcal{H}$: a set of mandatory constraints that must be satisfied by any valid solution, e.g., a supplier cannot be present simultaneously in two places.
- **Soft constraints** $\mathcal{X}$: a set of constraints, including preferences, that are not strictly mandatory and are allowed to be violated in a valid solution, but stakeholders can assign higher or lower importance to them, e.g., a supplier may consider quite important, but not mandatory, to fulfill his/her tasks within a specific time interval.

The distinction between these two kinds of constraints is relevant in our formalization. Hard constraints allow the identification of a *valid solution*. In the proposed methodology, we will consider only valid solutions, and starting from them, we will evaluate their degree of fairness. Indeed, when a solution is invalid, i.e., it does not satisfy hard constraints, we do not compute its fairness value. Hard constraints can be considered real-world limitations determined by the problem at hand, whose satisfaction is necessary to identify a really implementable solution. Conversely, soft constraints are used to measure the degree of fairness and compare the goodness of two valid solutions. They represent the preferences expressed by each stakeholder and can be set aside in case they prevent the satisfaction of a hard constraint or

Table 1
Task assignment $\mathcal{A}$, where the task T_1 requires two slots of two hours (T_{1-2}), T_2 one slot of three hours (T_{2-3}), and T_3 one slots of two hours (T_{3-2}) and one of an hour (T_{3-1}).

<table>
<tr><th></th><th>8–9</th><th>9–10</th><th>10–11</th><th>11–12</th><th>12–13</th><th>13–14</th></tr>
<tr><td>Mon</td><td></td><td></td><td></td><td colspan="2">T_{1-2}</td><td></td></tr>
<tr><td>Tue</td><td></td><td></td><td colspan="2">T_{3-2}</td><td></td><td></td></tr>
<tr><td>Wed</td><td colspan="2">T_{1-2}</td><td colspan="3">T_{2-3}</td><td></td></tr>
<tr><td>Thu</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Fri</td><td></td><td></td><td>T_{3-1}</td><td></td><td></td><td></td></tr>
</table>

they are in contrast with the satisfaction of the preferences of other stakeholders.

As described previously, each type of constraint is expressed in reference to its own specific temporal dimension, which is defined by the notion of temporal domain.

Definition 1 (*Temporal Domain*). A temporal domain $\mathbb{TD}$ is a tuple $(TD, \leq)$ where TD is a set of not-empty temporal instants and $\leq$ is an ordering relation on TD [39].

Concerning the representation in Fig. 1, each assignment is characterized by two orthogonal subdivisions of the temporal domain that are defined with respect to two different temporal granularities.

Definition 2 (*Temporal Granularity*). Given a temporal domain $\mathbb{TD}$, the temporal granularity is a mapping G from the integers $\mathbb{I}$ (set of indexes) to a subset of $\mathbb{TD}$ such that:

1. If $i < j$ and $G(i)$ and $G(j)$ are not empty, then the elements of $G(i)$ are less than all the elements of $G(j)$.
2. If $i < k < j$ and $G(i)$ and $G(j)$ are not empty, than also $G(k)$ is not empty.

In other words, the granularity defines a partition of the temporal domain in elements considered indivisible units, called *granules*. Given two temporal granularities G_1 and G_2, we can say that $G_1 \prec G_2$, if the granules of G_1 are finer than the granules of G_2, namely $\forall g_1 \in G_1\ \exists g_2 \in G_2$ such that $g_1 \subset g_2$.

In this paper, we will consider only discrete temporal domains, namely, temporal domains where each element, except the last, has an immediate successor, and each element, except the first, has an immediate predecessor. In our case, the considered discrete temporal domain $\mathbb{TD}$ has been subdivided using two orthogonal granularities: the set of work days in a week W and the set of useful hours inside a day H. Clearly, the relation $H \prec W$ holds between these two temporal granularities.

Given two temporal granularities, the intersection between the induced subdivisions will be called *time slot*.

Definition 3 (*Time Slot*). Given two temporal granularities G_1 and G_2, such that $G_1 \prec G_2$, a time slot is a pair $\langle g_1, g_2 \rangle$, such that $g_1 \in G_1$ and $g_2 \in G_2 \wedge g_1 \subset g_2$.

With reference to the specific configuration considered in this paper, let W be the set of work days in a week, i.e., $W = \{Monday, Tuesday, Wednesday, Thursday, Friday\}$, and H be the set of hours in a typical work day, e.g., $H = \{8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19\}$. We represent a time slot in a day as a pair $\langle h, w \rangle$, where $h = (h_{begin}, h_{end})$, with $h_{begin}, h_{end} \in H$, and $w \in W$.

In the following, we will use the $\mathcal{S}$ symbol to denote the available time slots concerning a particular choice of the two orthogonal granularities G_1 and G_2. Each time slot in $s \in \mathcal{S}$ can also be graphically represented as a cell inside a matrix $M^{\mathcal{S}}$. With reference to the representation in Table 1, the rows denote the work days in W, and the columns are the time intervals inside the day.

Given such subdivisions in time slots, the considered problem aims to assign all the tasks in a predefined set to the proper time slot, considering the two kinds of identified constraints.

Definition 4 (*Shared Tasks*). Given a set $\mathcal{T}$ of available tasks that need to be allocated inside a set of time slots of $\mathcal{S}$, each shared task $T \in \mathcal{T}$ is characterized by an identifier and a temporal duration θ defined with respect to the finer granularity: $T = \langle id, \theta \rangle$.

In the following, for not cluttering the notation, we will denote a generic task $T = \langle id, \theta \rangle$, as T_{id_θ}. For instance, in the representation of Table 1, the task T_{1_2} has identifier 1 and takes 2 h to be completed.

Definition 5 (*Sequential Task Assignment*). Given a set of shared tasks $\mathcal{T} = \{T_{1\_\theta_1}, T_{2\_\theta_2}, \dots, T_{m\_\theta_m}\}$ and a set of time slots $\mathcal{S}$, a sequential task assignment $\mathcal{A}$ is a function $\mathcal{A} : \mathcal{T} \rightarrow \wp(\mathcal{S})$ which associate each shared task in $\mathcal{T}$ to a set of timeslots in $\mathcal{S}$.

Given a shared task $T_{id\_\theta} \in \mathcal{T}$, we say that $T_{id\_\theta} \sqsubset \{s_1 \dots s_n\}$, if T_{id_θ} has been allocated to the set of time slots $\{s_1 \dots s_n\}$. For instance, $T_{id\_\theta} \sqsubset \{\langle(8, 10), Monday\rangle, \langle(14, 15), Thursday\rangle\}$ means that such shared task will be provided on Monday from 8 a.m. to 10 a.m. and on Thursday from 2 p.m. to 3 p.m.

Table 1 illustrates an example of the allocation of a set of shared tasks inside a set of time slots. As you can notice, two slots of two hours are allocated to both the shared tasks with identifier 1 (T_{1_2}) and identifier 3 (T_{3_2}); conversely, a single slot of three hours has been allocated to the shared task with identifier 2 (T_{2_3}).

Each possible sequential task assignment represents a *solution* to our problem. However, among all the possible solutions, we are only interested in the valid ones, namely the sequential task assignments that comply with the set of given hard constraints.

Definition 6 (*Valid Solution*). Given a set of hard constraints $\mathcal{H}$, a set of shared tasks $\mathcal{T}$, and a set of time slots $\mathcal{S}$, a sequential task assignment $\mathcal{A}$ is said to be a valid solution if and only if $\mathcal{A}$ satisfies all the constraints in $\mathcal{H}$.

Several valid solutions can be identified for the same set of shared tasks $\mathcal{T}$, set of time slots $\mathcal{S}$, and hard constraints $\mathcal{H}$. However, from the perspective of fairness, some solutions may be better than others. In particular, every stakeholder can define some time slot preferences, namely soft constraints, through a matrix formalized as follows.

Definition 7 (*Soft Constraints Matrix*). Given the set of available time slots $\mathcal{S}$, the preference of a stakeholder k for each time slot is expressed by a value $n \in [-1, 0]$, where n closer to -1 highlights an undesirable time slot. A set of soft constraints $\mathcal{X}_k$ for k can be represented as a matrix $M_k^{\mathcal{X}}$, with the same structure as $M^{\mathcal{S}}$, but containing in each cell the value n, instead of the assigned task.

For instance, Table 2 represents the preferences of a user k where to avoid cluttering the notation, all the zero values are omitted. In this case, the less preferable slots for k are before 10 a.m., time slots from 10 a.m. to 11 a.m. have a medium level of dislike, and all other time slots can be considered equally preferable by the user. This soft constraint representation makes it easier to preserve stakeholder privacy since they do not require any explanation.

Given a set of soft constraints $\mathcal{X}_k$ for stakeholder k, we generically define the local fairness of a solution in the following way.

Definition 8 (*Local Fairness*). Given a set of hard constraints $\mathcal{H}$ and a set of soft constraints $\mathcal{X}_k$, a set of shared tasks $\mathcal{T}$ and a set of time slots $\mathcal{S}$, the local fairness value for the stakeholder k of a valid solution $\mathcal{A}$, denoted as $f(\mathcal{A}, k)$, is defined as the degree to which $\mathcal{A}$ satisfies the constraints in $\mathcal{X}_k$ for stakeholder k.

Table 2
Example of soft constraints represented as a matrix $M_k^{\mathcal{X}}$.

	8–9	9–10	10–11	11–12	12–13	13–14
Mon	−1	−1	−0.5			
Tue	−1	−1	−0.5			
Wed	−1	−1	−0.5			
Thu	−1	−1	−0.5			
Fri	−1	−1	−0.5			

Table 3
Notation summary.

Symbol	Description
$\mathcal{H}$	Set of hard constraints
$\mathcal{X}$	Set of soft constraints
K	Set of stakeholders
$\mathcal{X}_k$	Set of soft constraints of the stakeholder k
$\mathbb{TD}$	Temporal domain
G, G_i	Temporal granularity
$\mathcal{S}$	Set of time slots
$\mathcal{T}$	Set of shared tasks
T_{id_θ}	Shared task with id and duration θ
$\mathcal{A}$	Task assignment
$M^{\mathcal{S}}$	Matrix representation of the time slot in $\mathcal{S}$
$M_k^{\mathcal{X}}$	Matrix of soft constraint $\mathcal{X}$ of the stakeholder k
$f(\mathcal{A})$	Local fairness of the assignment $\mathcal{A}$
$g(\mathcal{A})$	Global fairness of the assignment $\mathcal{A}$

In reference to the formalization provided in Definition 7, local fairness is measured by the sum of the elements in the matrix $M_k^{\mathcal{X}}$. Clearly, a value near zero represents the optimal fairness situation for user k, while a value less than zero means that some soft constraints are not satisfied by the provided solution.

Definition 9 (*Global Fairness*). Given a set of hard constraints $\mathcal{H}$ and a collection of sets of soft constraints $\mathcal{X}_{k_1} \dots \mathcal{X}_{k_n}$, for stakeholders $K = \{k_1 \dots k_n\}$, a set of shared tasks $\mathcal{T}$ and a set of time slots $\mathcal{S}$, the global fairness of a valid solution $\mathcal{A}$, denoted as $g(\mathcal{A})$, is defined as the degree to which $\mathcal{A}$ satisfies the constraints in $\mathcal{X}_{k_i}$ for all stakeholders in K.

In Definition 9 the generic notion of "degree" of satisfaction is used, in practice, we mean that in order to achieve a global fair solution, we need not only to increase each individual local fairness values, but also to balance them, so that no one is satisfied at expense of another. A complete formulation of local and global fairness measures will be provided in Section 4 in relation to the two considered case studies.

At this stage, the notions of hard and soft constraints are defined and applied to a sequential task assignment. These preliminary definitions also introduce the concepts of local and global fairness in evaluating valid solutions. Table 3 provides a summarization of the main symbols introduced here and used throughout the paper.

Given this formalization, the following section introduces the considered case study on which the proposed algorithms for local and global fairness achievement will be applied.

4. Case study

As case studies, we consider two scenarios: the timetable creation problem in the university domain, where professors and students share classes, and the travel planning problem in the tourist domain, where guides and tourists share guided visits. In both domains, one source, namely classes and visits, is shared between two groups of users (i.e., stakeholders).

Our purpose is to automatically define a fair timetable, where the concept of fairness is examined from the perspectives of both stakeholders. We will solve the two sub-problems separately to achieve local fairness and then combine the best solutions to provide a global fair solution, if any.

4.1. University timetable construction

As the first application domain, we consider the Italian university scenario and the problem of automatically defining a fair timetable covering all the planned class sessions planned during a semester.

In this case, the opposite stakeholders are on one side, the set of students attending the same degree course in the same year, and on the other side, each individual professor. Notice that these two stakeholder categories follow two different grouping criteria: all the students of the same degree course and year are considered as a unique stakeholder, since they share the same constraints, need to attend the same classes, and in the considered scenario, they cannot express individual preferences. Indeed, in Italy, students cannot personalize their degree program, which is instead standard and depends on the degree course and year. Conversely, each professor is considered an individual with his/her preferences and constraints, coming also from the need to teach in different classes belonging to several degree courses.

In the university setting, the set of shared tasks $\mathcal{T}$ (see Definition 4) is represented by the set of classes such as, for example, *Programming*, *Physics*, and *Probability*. Each degree program d includes a subset of these classes, divided across different academic years y which range from 1 to 6.[1] The subset of courses available in a given year y of a degree program d is denoted as C_y^d.

In the timetable construction, each day can be viewed as a sequence of class sessions (i.e., a sequence of lectures), while a week consists of a sequence of university days. Consequently, the sequential task assignment problem we examine involves two temporal dimensions: one representing the sequence of lectures within a single day (horizontal) and the other capturing the sequence of days within a university week (vertical). More in detail, a sequential task assignment $\mathcal{A}$ (see Definition 5) refers to the weekly course allocation, which defines the timeline of a typical week for students in year y of the degree program d, ensuring that the courses in C_y^d are properly allocated.

The set of hard constraints $\mathcal{H}$ that each valid solution needs to satisfy includes these two constraints: H_1^p, namely, "*each professor p cannot take two class sessions in the same time slot for different courses*", and $H_1^{S_y^d}$, namely, "*students S attending year y of the degree program d cannot attend two class sessions simultaneously*". Given that, the sequential task assignment problem can be viewed from two perspectives regarding fairness: the professors' and the students' sides. Both sides have different preferences, and their degree of satisfaction helps determine the fairest solution.

4.1.1. Local fairness for professors

For the professor sub-problem, besides to the set of hard constraints $\mathcal{H}$ mentioned above, each professor can express his/her undesired time slots specified as *soft constraints*. In particular, a professor could identify the more desired time slots by assigning a numeric value in the interval $[-1, 0]$ to each slot: the closer it is to -1, the more unfavorable it is. Table 4 exemplifies the notation to formulate soft constraints. In the example, the considered professor declares that he/she does not desire to teach on Friday (score −1), he/she would not like to teach on Wednesday in the very early morning (score −0.5), and he/she would not like too much to teach on Monday morning (score −0.25).

Finally, we consider another *soft constraint* for professors, i.e., the possibility of having classes on two consecutive days. Whenever a professor specifies this preference, to manage it, we introduce the function $f_{consecutive}()$ that returns 0 if the preference is respected by a schedule, a negative value otherwise.

According to Definition 8, fairness can be evaluated for a specific professor p, where a value closer to zero indicates a fairer weekly course assignment.

[1] In Italy, Bachelor's programs last three years, and Master's programs last two years. Long single-cycle degree programs last five or six years.

Table 4
Soft constraints expressed by a professor.

	8–9	9–10	10–11	11–12	12–13	13–14
Mon	−0.25	−0.25	−0.25			
Tue						
Wed	−0.5	−0.5				
Thu						
Fri	−1	−1	−1	−1	−1	−1

Table 5
Weekly course assignment W_y^d, where the course C_1 requires two slots of two hours (C_{1-2}), C_2 one slot of three hours (C_{2-3}), and C_3 one slot of two hours (C_{3-2}) and one of an hour (C_{3-1}).

	8–9	9–10	10–11	11–12	12–13	13–14
Mon	C_{1-2}					
Tue			C_{3-2}			
Wed	C_{1-2}		C_{3-1}			
Thu			C_{2-3}			
Fri						

Table 6
Fairness Value for a professor considering the schedule in Table 5 for the courses C_1, C_2, and C_3.

	8–9	9–10	10–11	11–12	12–13	13–14
Mon	−0.25	−0.25	−0.25			
Tue						
Wed	−0.5	−0.5				
Thu						
Fri	−1	−1	−1	−1	−1	−1
Sum	−0.75	−0.75	0	0	0	0
Tot			−1.5			

Professor Fairness. *Given a valid weekly course assignment $\mathcal{A}$ (i.e., an assignment satisfying all the hard constraints in $\mathcal{H}$) and the set of soft constraints $\mathcal{X}_p$ expressed by a professor p, the fairness of the solution $\mathcal{A}$, denoted as $f(\mathcal{A}, p)$, is computed by summing the values of the soft constraints for those time slots in which a lesson has been finally allocated, with the score obtained by the function $f_{consecutive}()$.*

For instance, given the course assignment of Table 5 and the soft constraints of Table 4, the final fairness value is −1.5. In Table 6, it is possible to see how the fairness score is computed: only the colored scores (i.e., only the slots with a lecture in the assignment of Table 5) are considered in the sum.

4.1.2. Local fairness for students

Since it is difficult to define individual fairness for each student and because in the Italian university style degree curricula are standard, i.e. most courses are mandatory in a specific year and the schedule is not flexible, the fairness value, in this case, is computed for the set of students attending a specific year of the same degree program. In other words, in this case, the stakeholder is a group of students who have to attend the classes altogether.

As it happened for the professor sub-problem, besides to the set of hard constraints $\mathcal{H}$, we impose some soft constraints representing the overall preferences of this group and whose degree of satisfiability is measured by a function. In particular, we have selected from [38] the more relevant soft constraints to provide good schedules for students. The selected satisfaction functions representing the considered set of soft constraints are:

- f_{lunch}: since a fair timetable guarantees a lunch break on each day of classes, the function has been introduced to measure the number of days with a lunch break.
- $f_{hourGap}$ and $f_{weekGap}$: they measure the number of gaps between two consecutive classes on the same day or in the week. In particular, they are defined to minimize the number of free hours between classes (i.e., we want to avoid the situation of a day where there is a lesson from 8.30 to 9.30 a.m. and another from 6.30 to 7.30 p.m.) and the number of free days in a week between two busy days, e.g., a class scheduled only on Monday and on Friday is not a desirable scenario for commuting students. This example highlights the flexibility of the approach in integrating contextual information into the problem, formalized as soft constraints.
- f_{day}: it measures the number of days with at least a lesson and is used to minimize the days of lectures in a week.
- f_{early} and f_{late}: measures the number of classes in the early morning or the late evening. They have been introduced to avoid very early and very late classes with the final aim to encourage the use of public transport.

All these functions return a value close to zero when a schedule complies with the soft constraint they are checking or a negative value otherwise. Again, the more the final value is next to zero, the fairer the allocation is.

For instance, Table 7 reports the timeline of the first year of the degree program in Computer Science, and Table 8 summarizes the values obtained by each function and the final fairness score. We can notice that f_{lunch} is not zero since there is no lunch break on Monday. On the other hand, $f_{hourGap}$ has a value close to zero as there is only one free hour for two days between classes in the timeline. The other non-zero value is obtained by f_{day}, which points out the number of days with a class scheduled. All the remaining functions are at zero, being that all other soft constraints are satisfied.

Based on Definition 8, fairness can be defined for the students attending the yth year of the degree d.

Student class fairness. *Given a valid weekly course assignment $\mathcal{A}$ (i.e., an assignment satisfying all the hard constraint in $\mathcal{H}$) and the value of each soft constraint function defined for a student class S_y^d, the assignment's fairness $f(\mathcal{A}, S_y^d)$, is computed by summing the values of the soft constraint functions.*

Given the values in Table 8, the student class fairness equals −1.01. Again, the weekly course assignment becomes fairer as the final value approaches zero.

4.1.3. Global fairness for the university timetable problem

Considering the concept of global fairness introduced in Definition 9, we can now contextualize it explicitly for the university use case.

Global university timetable fairness. *Given the set of professors P, the set of the degree programs D, the set of the years Y of each degree program, and a valid weekly course assignment $\mathcal{A}$ (i.e., an assignment satisfying all the hard constraints $\mathcal{H}$). The global university timetable fairness $g(\mathcal{A})$ is computed as a pair of values $g(\mathcal{A}) = \langle g_a(\mathcal{A}), g_s(\mathcal{A})\rangle$:*

$$g_a(\mathcal{A}) = \frac{1}{2} \cdot \left(\frac{1}{|P|} \sum_{p \in P} f(\mathcal{A}, p) + \frac{1}{|S|} \sum_{d \in D} \sum_{y \in Y} f(\mathcal{A}, S_y^d) \right)$$

$$g_s(\mathcal{A}) = \sqrt{\frac{1}{|P|} \sum_{p \in P} (f(\mathcal{A}, p) - g_a(\mathcal{A}))^2 + \frac{1}{|S|} \sum_{d \in D} \sum_{y \in Y} (f(\mathcal{A}, S_y^d) - g_a(\mathcal{A}))^2}$$

where $|S|$ is the number of different S_y^d involved in $\mathcal{A}$.

In other words, the global fairness for the university case study is derived from the local fairness scores of *Professor fairness* and *Student class fairness*. The combination of these two values includes a computation of an average fairness value and the standard deviation from that value. Indeed, the global fairness computation also needs to consider balancing the local fairness associated with each stakeholder.

4.2. Tourism traveling planning

The tourism domain is considered a second case study, and now the problem is the automatic definition of travel planning. In this scenario, opposing stakeholders include tourists involved in the same sightseeing,

Table 7
An example of a complete weekly course assignment W_y^d, where d = Computer Science and $y = 1$. C_{pro-2} and C_{pro-3} are two and three hours slots of Programming, C_{arc-2} two hours slots of Computer Architecture. Then, C_{mat-2} and C_{mat-3} are two and three-hour slots of Mathematical Analysis, and C_{log-2} and C_{log-3} are two and three-hour slots of Logic.

<table>
<tr><th></th><th>8–9</th><th>9–10</th><th>10–11</th><th>11–12</th><th>12–13</th><th>13–14</th><th>14–15</th><th>15–16</th><th>16–17</th><th>17–18</th><th>18–19</th></tr>
<tr><td>Mon</td><td></td><td></td><td></td><td colspan="2">C_{pro-2}</td><td colspan="2">C_{arc-2}</td><td colspan="2">C_{log-2}</td><td></td><td></td></tr>
<tr><td>Tue</td><td></td><td colspan="2">C_{arc-2}</td><td></td><td colspan="3">C_{pro-3}</td><td colspan="3">C_{mat-3}</td><td></td></tr>
<tr><td>Wed</td><td></td><td></td><td colspan="2">C_{arc-2}</td><td></td><td colspan="2">C_{mat-2}</td><td colspan="3">C_{log-3}</td><td></td></tr>
<tr><td>Thu</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Fri</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Table 8
Soft constraint score computed for the student sub-problem looking at the timeline of Table 7.

f_{lunch}	$f_{hourGap}$	$f_{weekGap}$	f_{day}	f_{early}	f_{late}	Sum
−0.33	−0.08	0	−0.6	0	0	−1.01

thus sharing the same constraints, and tour guides with their own constraints.

In this use case, the set of shared tasks defined in Definition 4 is represented by the set of visitable points of interest, e.g., $\mathcal{T}$ = {*Archaeological museum at the Roman Theatre*, *Natural History museum*, *The Cathedral*, …}. Starting from the set of visitable points of interest, it is possible to create different guided tours, where each one is considered as a subset of $\mathcal{T}$.

As in the previous case study, the problem we address considers two temporal dimensions: each week is represented as a sequence of days, each consisting of a sequence of point-of-interest visits. Again, the task assignment problem and the fairness analysis can be examined from the perspectives of the two stakeholders: the guides and the tourists.

Every tourist guide is responsible for multiple points of interest $v \in \mathcal{T}$, thus the first *hard constraint* defined starting from the guide perspective $\mathcal{H}_1^g$ is "*each guide* g *cannot take two visits in the same slot for different groups of tourists*". Secondly, another *hard constraint* $\mathcal{H}_2^g$ is established: "*each guide* g *cannot hold a visit to a point of interest* v *during its closing hours*". Finally, we can identify a hard constraint also from the perspective of the tourist $\mathcal{H}_1^t$ regarding the overlaps between items: "*each tour group* $t \in T$ *cannot attend two visits in the same time slot*". Overall, the set $\mathcal{H}$ of hard constraints is composed of three constraints that any valid solution needs to satisfy.

Similarly to the professor sub-problem, each guide can express soft constraints to indicate their preferences for the slots, using a number between −1 and 0, where −1 indicates the least preferred slots. According to Definition 8, we can establish the notion of fairness for each guide g.

Guide local fairness. *Given a valid weekly tourist visit assignment* $\mathcal{A}$ *(i.e., an assignment satisfying all the hard constraints* $\mathcal{H}$*) and the set of soft constraints* $\mathcal{X}_g$ *expressed by a guide* g*, the fairness of the solution* $\mathcal{A}$*, denotes as* $f(\mathcal{A}, g)$*, is computed by summing the values of the soft constraints for those time slots in which a visit has been finally allocated.*

The representation of constraints and the assignment of guide sub-problem are analogous to the previously mentioned example involving professors in Section 4.1.1.

Similarly to students, collecting tourists' constraints is not trivial; therefore, individual fairness cannot be defined. Likewise, in this scenario, the fairness value is calculated for a set of customers, i.e., the group of tourists attending the same tourist tour together. For them, we establish some soft constraints through the functions defined as follows:

- $f_{hourGap}$ and $f_{weekGap}$: they measure the number of gaps between two consecutive visits within a day or week. The aim is to minimize free hours between activities and empty days, as tourists typically prefer consecutive visits without breaks.
- f_{day}: it measures the number of days with at least a visit to a point of interest, and it could be used when the tourist group has a limited number of days to spend.
- $f_{computeDistance}$: it measures the distance that users walk between points of interest visits; thus, when designing tours, the distance between attractions should be minimized.

The functions above return a negative value if the timeline contradicts them and zero otherwise. Once again, fairer timelines have a fairness value close to zero.

Tourist local fairness *Given a valid weekly tourist visit assignment* $\mathcal{A}$ *(i.e., an assignment satisfying all the hard constraints in* $\mathcal{H}$*) and the value of each soft constraint function defined for a group of tourists* t*, the fairness of the assignment is* $f(\mathcal{A}, t)$*, which is computed by summing the values of the soft constraint functions.*

The way the functions are represented and the sub-problem's assignment is similar to the example previously described for students in Section 4.1.2.

The notion of global fairness (Definition 9) regarding the tourism domain can be defined similarly to the university use case.

Tourism global fairness. *Given the set of the guides* G*, the set of tour groups* TG*, and a valid weekly tourist visit assignment* $\mathcal{A}$ *(i.e., an assignment satisfying all the hard constraints* $\mathcal{H}$*), the tourism global fairness* $g(\mathcal{A})$ *is computed as a pair* $\langle g_a(\mathcal{A}), g_s(\mathcal{A})\rangle$

$$g_a(\mathcal{A}) = \frac{1}{2} \cdot \left(\frac{1}{|G|} \sum_{g \in G} f(\mathcal{A}, g) + \frac{1}{|TG|} \sum_{t \in TG} f(\mathcal{A}, t) \right)$$

$$g_s(\mathcal{A}) = \sqrt{\frac{1}{|G|} \sum_{g \in G} (f(\mathcal{A}, p) - g_a(\mathcal{A}))^2 + \frac{1}{|TS|} \sum_{t \in TS} (f(\mathcal{A}, t) - g_a(\mathcal{A}))^2}$$

Once again, global fairness is computed starting from the local fairness of all the stakeholders involved taking care of both their average and standard deviation values.

5. The FaST-MOSA algorithm

The local and global fairness problems introduced in the previous section can be viewed as a multi-objective optimization problem subject to certain constraints. In particular, for each stakeholder k, the hard constraints $h \in \mathcal{H}$ can be considered the optimization constraints. In contrast, soft constraints $x \in \mathcal{X}_k$ can be translated into objective functions with eventually different weights.

$$min\langle x_1, \dots x_n \rangle \tag{1}$$

$$\text{subject to } h_1 \dots h_m \tag{2}$$

Finding an optimal solution for such optimization problems is computationally expensive; therefore, we will employ heuristics to find a satisfactory solution within a reasonable amount of time. In particular, we will exploit the dominance-based Multi-Objective Simulated Annealing (MOSA) [16] technique. The choice towards the MOSA technique, instead of other approaches like a greedy algorithm is due to two main reasons: (i) it is able to reach a global optimum if annealed sufficiently slowly, while other solutions, like the greedy algorithm, can stuck in local optima [40]; (2) in the search space exploration, instead of using completely random artificial solutions, we can rely on the concept of fairness to accelerate the convergence process by guiding the perturbation process, originating our extension called FaST-MOSA.

5.1. Multi-objective optimization with MOSA

The MOSA technique resembles the physical process of heating a material and slowly lowering the temperature to decrease defects. In the same way, the heuristic is based on the use of a parameter, called temperature, which is high at the beginning, allowing a random exploration of the search space, then, it is progressively decreased to converge the solution to the optimum. It has been proven that the MOSA technique can reach a global optimum if the solution is annealed sufficiently slowly [41]. Conversely, one of the major drawbacks of many optimization heuristics, like the greedy algorithm, is the possibility of getting stuck in local optima.

The exploration of the search space is based on the comparison of a current solution s_{curr} with another potential solution s_{new} which is obtained from s_{curr} by applying to it a simple modification, named perturbation. The probability of choosing s_{new} in place of s_{curr} depends on both the goodness of the new solution compared to the previous one and the value of the temperature. Indeed, at higher temperatures (i.e., at the beginning), the probability of choosing the new solution is greater, independently of its goodness, allowing a more random search space exploration. The goodness evaluation is based on the notion of *dominance*, which also allows us to define a partial order on the solutions: given two solutions s_1 and s_2, we say that s_1 dominates s_2 (i.e., $s_1 \prec s_2$) if s_1 is better in at least one object function and equivalent in the remaining ones:

$$s_1 \prec s_2 \Leftrightarrow \begin{cases} \forall i \in \{1, \dots n\} \; x_i(s_1) \le x_i(s_2) \\ \exists j \in \{1 \dots n\} \; x_j(s_1) < x_j(s_2) \end{cases} \tag{3}$$

The set of mutually non-dominating solutions is called *Pareto set* $\mathcal{P}$, and a solution not dominated by another solution is called *Pareto optimum*. Given a Pareto set $\mathcal{P}$ it is possible to compute the *Pareto front* $\mathcal{F} \subset \mathbb{R}^n$, which is the set of points in the objective space: $\mathcal{F} = \{\bar{x}(s) | s \in \mathcal{P}\}$, where $\bar{x}$ is the evaluation of the solution s with respect to the various objective functions $x_1, \dots x_n$. A MOSA algorithm aims to move the current Pareto front towards the optimal one (i.e., the Pareto front of the Pareto optimum set) while encouraging the diversification of the candidate solutions.

Algorithm 1

Input: $\mathcal{S}$ time slots, $\mathcal{T}$ shared tasks, t_{init} initial temperature
Output: $\mathcal{P}$ Pareto set of task assignments

1: $\mathcal{P} \leftarrow \emptyset$
2: $s \leftarrow randomAssignment(\mathcal{S}, \mathcal{T})$
3: $\mathcal{P} \leftarrow \mathcal{P} \cup \{s\}$
4: $t \leftarrow t_{init}$
5: **while** $t < t_{min}$ **do**
6: $\quad s' \leftarrow perturbation(s)$
7: $\quad p \leftarrow acceptanceProbability(s, s', t, \mathcal{P})$
8: $\quad$ **if** $rand(0, 1) < p$ **then**
9: $\quad\quad \mathcal{P} \leftarrow \mathcal{P} \setminus \{s\} \cup \{s'\}$
10: $\quad\quad s \leftarrow s'$
11: $\quad$ **end if**
12: $\quad t \leftarrow updateTemperature(t)$
13: **end while**
14: **return** $\mathcal{S}$

In the considered problem, a solution s represents a task assignment $\mathcal{A}$, i.e., a weekly course or touristic visit allocation. Thus, the Pareto set $\mathcal{P}$ contains a set of equally good assignments regarding the given notion of fairness. The algorithm starts with a random task assignment (line 2), and at each iteration at line 6, a perturbation of the current solution s is done, obtaining a new solution s'. For our purpose, the perturbations are moving one shared task to a new slot or swapping the time slots of two tasks. Then, it evaluates the probability p of taking s' as the new current solution. The computation of p of the line 7 depends on the energy of the two solutions and the temperature t. The energy of a solution s, denoted by $E(s, \mathcal{F})$, measures the portion (i.e., number of solutions) of the current Pareto front $\mathcal{F}$ that currently dominates s: $E(s, \mathcal{F}) = |\{v \in \mathcal{F} \mid v \prec \bar{f}(s)\}|$. Note that the energy of a solution s belonging to the Pareto front is 0. Given that, the acceptance probability $P(s, s', t)$ is defined as:

$$P(s, s', t) = \min\left(1, \exp\left(-\frac{E(s, \mathcal{F}') - E(s', \mathcal{F})}{T}\right)\right) \tag{4}$$

where $\mathcal{F}' = \mathcal{F} \cup \{s'\}$. Notice that a candidate solution dominated by one or more members of the current estimated Pareto front can still be accepted with a probability $p > 0$. On the other hand, a solution belonging to the Pareto front is always accepted with probability $p = 1$.

If s' is chosen in place of s (lines 8–11), the Pareto set and the current solution are adequately updated.

5.2. Fair sequential task assignment-MOSA

The core of the MOSA's solution exploration is the perturbation applied at each iteration. To determine s', i.e., the new task assignment, it is essential to define the possible operation that can be performed: (a) moving a randomly chosen shared task $t \in \mathcal{T}$ to a random empty slot and (b) swapping two randomly chosen shared tasks $t_1, t_2 \in \mathcal{T}$.

In this article, we evaluate two alternatives: the first is an approach based on pure random perturbations, and the second one uses only perturbations that can resolve some unfair situations.

The first solution adheres to the original MOSA algorithm, where perturbations are randomly chosen from the two provided. It performs a random exploration of the search space to avoid getting stuck in local optima.

On the other hand, the second one proposes only changes that can improve at least one constraint. The priority is given to hard constraints, which must be satisfied to obtain a valid solution. Then, the perturbation can propose a change to improve the satisfiability of soft constraints. For clarity, we will refer to this new algorithm as Fair Sequential Task Assignment-MOSA (FaST-MOSA). The proposed variation reduces the explored search space in a clever way, discarding the evaluation of solutions that will not lead to a useful path. This is possible because in our specific scenario, all shared tasks in $\mathcal{T}$ need to be allocated in some way, i.e., we do not perform a choice among them, and the allocation order does not matter, i.e., we do not care if a task T_i takes place immediately before or after the task T_j. The focus is on allocating tasks to time slots, and we are only interested in identifying free and busy time slots. The proposed FaST-MOSA is an efficient alternative, i.e., it allows us to reach the solution with fewer iterations. Moreover, the proposed principles driving the perturbation construction could be applied to other MOSA variants, as discussed in the following section.

5.3. Extension to other MOSA variants

The term MOSA denotes not only a specific Multi-Objective Simulated Annealing technique, but also a family of algorithms that share a common framework while differing in their specific strategies and implementations. Since our approach operates at the level of the perturbation mechanism, FaST-MOSA can be naturally applied and integrated into other variants of the MOSA family without altering their underlying annealing procedure. In this paper, besides the classical MOSA, we also consider the Archived Multi-Objective Simulated Annealing (AMOSA) [3] as a representative variant. This variant shares the same core structure of the classical MOSA presented in Alg. 1, differing only in how it adjusts temperature schedules, acceptance rules, or adaptive parameters. In particular, it relies on an archive of non-dominated solutions and an acceptance rule based on the amount of domination, while preserving the same overall annealing pipeline as MOSA, to increase the performance of every single iteration.

The performed experiments reported in Sections 6 and 7 provide evidence that our solution could be effectively employed in other variants of the MOSA family, providing a flexible mechanism to embed fairness considerations into a family of optimization algorithms.

6. FaST evaluation

This section extensively evaluates the proposed approach using two real datasets: the timeline of the Computer Science Department of the University of Verona and the dataset of tourist Points of Interest (POIs) visited in Verona. The sets of hard and soft constraints for these two use cases are those presented in Sections 4.1 and 4.2, respectively.

Regarding the university dataset, we use a subset of the original one that concerns the courses offered in the bachelor's degree programs in Computer Science, Bioinformatics, and Biotechnology. The professors are limited to those with at least one module in one of the degree programs chosen for the evaluation. For each year of the degree program, we have a list of the shared tasks C_{id_θ} associated with a specific professor, representing the courses to be allocated in the available time slots.

The tourism dataset encompasses five tours covering 31 POIs. Each POI is assigned to a tour guide, and there are seven guides available. Every guide is responsible for multiple POIs and their associated tour visits. For each tour, we have a list of planned POI visits V_{id_θ}, each linked to a specific tour guide.

Given such datasets, we first explore the best hyperparameter configuration, then we compare the performances of the two strategies presented at the end of the previous section (i.e., FaST-MOSA and FaST-AMOSA) with the traditional MOSA and the AMOSA variant, and finally, we evaluate how the local fairness of a stakeholder can impact the local fairness of the other stakeholders, with the aim of finding the global solution.

All the experiments were conducted on a MacBook Pro equipped with an Apple M3 Pro processor and 18 GB of RAM. The code and dataset can be found in the GitHub repository.[2]

6.1. Tuning hyperparameters

To find the best configuration for the local fairness solution, we run our approach by varying (a) the initial temperature, which controls the number of iterations, and (b) the maximum number of perturbations at each iteration. Since the first step of the Alg. 1 is a random item allocation, the experiments are conducted with at least five seeds for each hyperparameter.

6.1.1. Initial temperature

The initial temperature t_{init} is tested from 10 to 100 for the university domain, and from 10 to 50 for the tourism dataset. The minimum temperature t_{min} is set to 5, and the maximum number of perturbations at each iteration is set to 10. The difference between the span of the trials is guided by the processing time: running the algorithm with guides and tours, which have many constraints, requires a longer execution time.

Fig. 2(a) reports the results obtained for the university domain. For an initial temperature greater than 20, the orange solution obtained with FaST-MOSA is consistently fairer than the other three approaches, both in terms of $g_a(\mathcal{A})$ and $g_s(\mathcal{A})$ as depicted on the left and on the right, respectively. Notably, FaST-MOSA achieves higher average global fairness values (higher g_a) while simultaneously exhibiting a lower dispersion across stakeholders (lower g_s), indicating a more balanced and equitable distribution of fairness. Even when employing the AMOSA approach, our FaST variant yields better results regardless of the initial temperature value, demonstrating its effectiveness in improving both the overall fairness level and stability.

Similar results are observed for the tourism travel planning problem, reported in Fig. 2(b), where our fairness-driven perturbation strategy achieves fairer solutions with lower initial temperature values, with both MOSA and AMOSA variants.

For both use cases, the processing time increases as the initial temperature rises, and the time increase is faster with FaST-MOSA. Note that FaST-MOSA can achieve a better fairness score with a lower initial temperature value and, thus, in less time. On the other hand, the AMOSA algorithm allows a higher number of iterations, maintaining the time computation limited compared to the traditional MOSA. Although the setup used to collect the results is not a high-performance computing system, the outcomes are produced in a reasonable timeframe, as shown in Fig. 3.

6.1.2. Number of perturbations at each iteration

We also test the maximum number of perturbations for each iteration, considering a range from 5 to 50 for both the use cases. The minimum temperature t_{min} and the initial temperature t_{init} are set to 5 and 25, respectively.

In Figs. 4(a) and 4(b), we observe that better results are achieved as the number of iterations increases. This deep exploration of the search space enables the improvement, albeit at the expense of increased processing time.

Also during these tests, FaST-based perturbations reaches the desirable results first with both the MOSA variants and datasets. This trend indicates that incorporating fairness guidance into the perturbation mechanism accelerates convergence towards balanced solutions, reducing the need for extensive exploration even in the early stages of the annealing process.

6.2. Perturbation evaluation

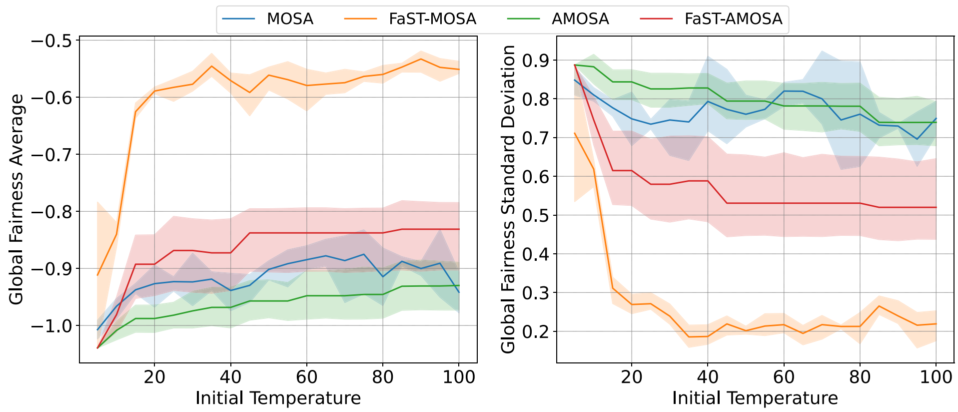
This section illustrates the results obtained with the two types of perturbations. This article presents a novel perturbation solution (i.e., the fairness-driven perturbation of FaST-MOSA and FaST-AMOSA) that is effective for this problem since the course and visit allocation orders are irrelevant. As we can notice from the plots of the previous section, the fairness-driven perturbation of the proposed MOSA variants reaches better results with a lower value of the analyzed hyperparameters compared to the random perturbations of the classical MOSA. Therefore, regarding time and the number of iterations of the algorithm, FaST-based strategy is better since it reaches the fairest result before the two considered MOSA variants.

In relation to the university use case illustrated in Figs. 2(a) and 4(a), the fairest solution, where the majority of preferences are met, is reached with an initial temperature of 20 and fewer than 20 perturbations using the FaST-MOSA algorithm.

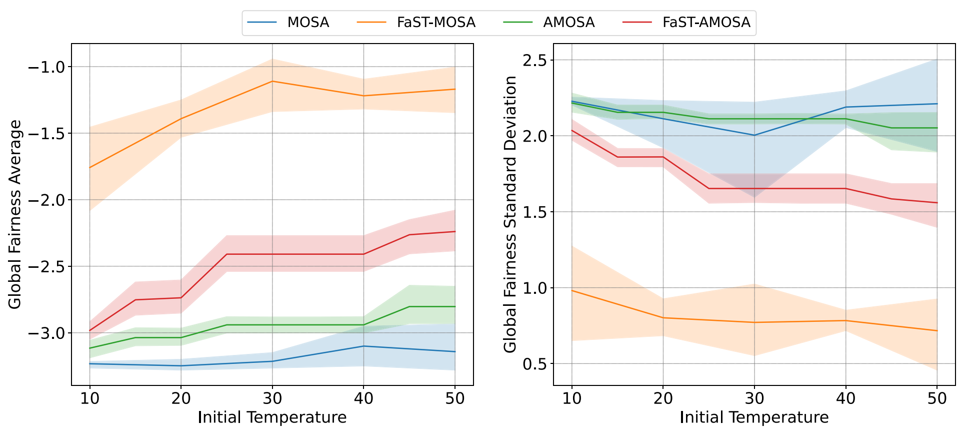
Regarding the tourism use case, as depicted in Figs. 2(b) and 4(b), the initial temperature remains more stable, showing little improvement as it increases. However, the goodness of FaST-MOSA is evident. In contrast, the number of perturbations has a greater influence, highlighting the effectiveness of fairness-driven perturbations: with only a maximum of 10 perturbations, the FaST-based variants achieve the same fairness scores that random perturbations reach only when using 50 perturbations.

In conclusion, the perturbations of FaST-MOSA enable better results with lower hyperparameter values, thereby minimizing the computational time across all use cases. Moreover, when execution time is a critical factor, applying fairness-driven perturbations within AMOSA allows the algorithm to achieve good-quality solutions with a limited computational budget.

[2] https://github.com/4nnina/fair_seq_task_assignment.



(a) University.



(b) Tourism.

Fig. 2. The global fairness score's variance and mean as the initial temperature changes.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

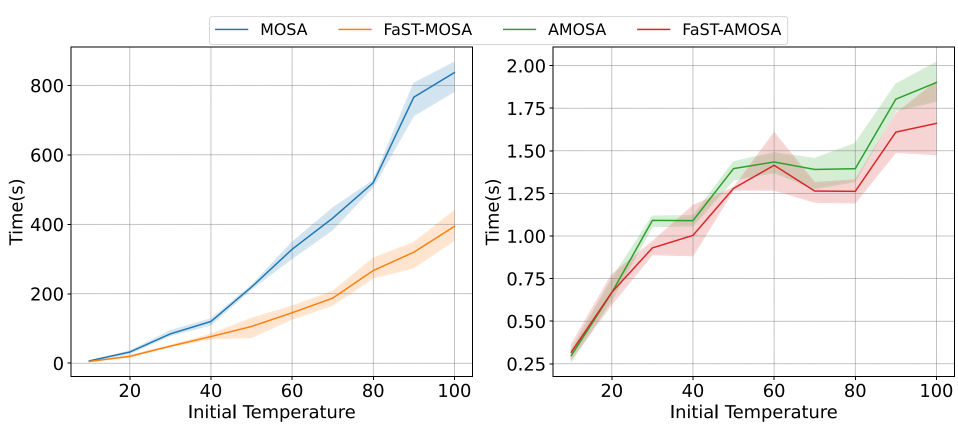
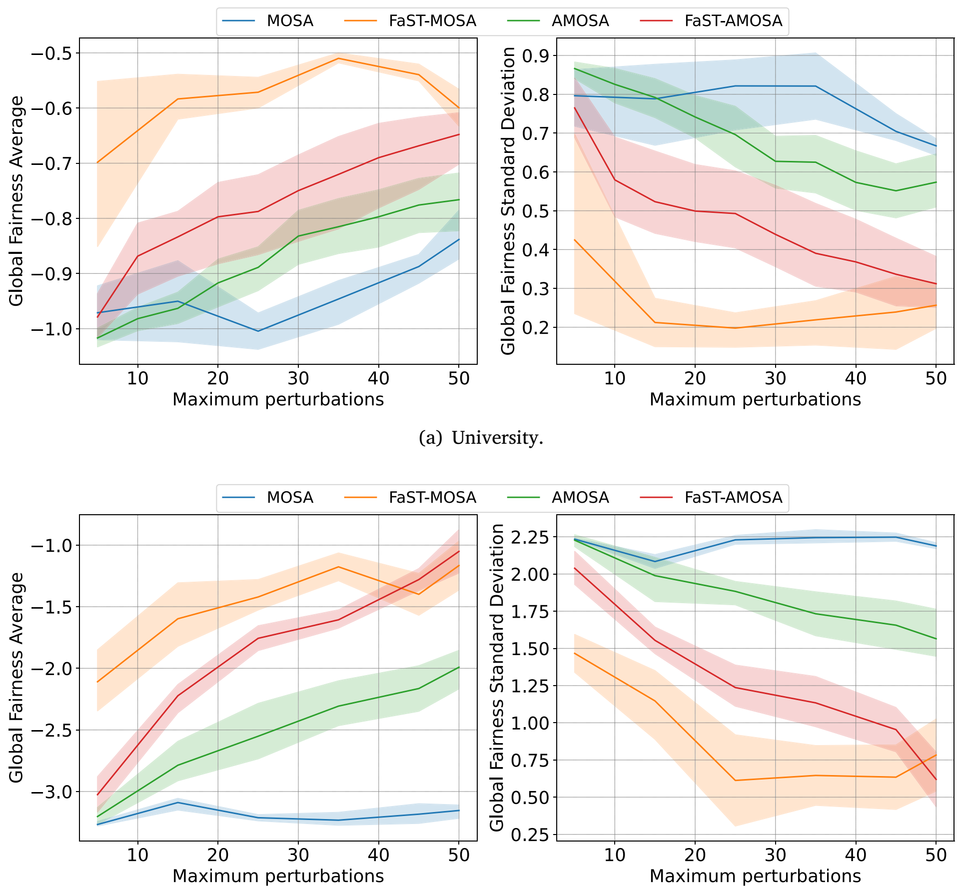


Fig. 3. The computational time's variance and mean as the initial temperature changes for the university dataset.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

7. Global fairness evaluation

In this section, we outline the process of achieving a global fair solution for all stakeholders. The identification of a global solution can be achieved in two ways: (1) starting from a set of identified local fair solutions and refining them to achieve global fairness, or (2) executing the FaST-MOSA algorithm from scratch while considering the soft constraints of all stakeholders, without relying on previously computed local fair solutions.

Given the results discussed in the previous section and the absence of strict time constraints, the following analysis focuses on the FaST-MOSA approach; however, the same considerations apply analogously when FaST perturbations are integrated within AMOSA.



(a) University.

(b) Tourism.

Fig. 4. The global fairness score's variance and mean as the maximum number of perturbations per iteration increases.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 9
Student fairness divided by degree courses and year, starting from professors' timeline.

	Computer science			Bioinformatics			Biotechnology		
	1	2	3	1	2	3	1	2	3
Fairness	−2.62	−1.15	−1.57	−1.93	−1.02	−1.56	−1.0	−0.8	−2.23

7.1. From local to global fairness

The first alternative to produce a global fair solution is to start by generating a local fair solution from each stakeholder's preferences, and then add all soft constraints to create a unified timeline suitable for real task assignments.

University timetable problem. Firstly, we analyze the scenario in which the timetable is optimized from the professors' perspective, i.e., a valid schedule that focuses on satisfying the soft constraints of the 32 professors. The resulting schedule is locally fair for professors, as on average, 95% of their soft constraints are satisfied, and no hard constraints are violated. However, this locally optimal configuration does not translate into a feasible timetable for students. When this schedule is transferred to the student side, most cohorts experience significantly unfair allocations. Table 9 reports the fairness values obtained for each degree program and year under the professor solution.

Across the nine student cohorts, the resulting fairness is poor: the average fairness value is below -1.5, indicating that several soft constraints, such as avoiding early hours, long gaps, and guaranteeing the lunch break are violated; with this approach, the global fairness of this professor-driven solution is $\langle -1.02, 1.04\rangle$, where the first component $g_a(\mathcal{A})$ represents the average global fairness value and the second $g_s(\mathcal{A})$ its standard deviation, reflecting a high dispersion among stakeholders. Applying FaST-MOSA with all soft constraints to this configuration produces an improvement, yielding a global fairness of $\langle -0.56, 0.22\rangle$.

Conversely, Table 10 presents the student class fairness obtained when the timetable is initially optimized for students. In this case, the resulting solution better satisfies the students' soft constraints, but introduces violations for professors: 9 out of 32 professors have at least one unsatisfied preference. The global fairness score is $\langle -0.84, 0.87\rangle$, indicating a more balanced situation than in the professor scenario.

Starting from this last assignment, which presents better fairness characteristics, we apply FaST-MOSA considering the global problem to improve the obtained solution. The algorithm converges to a global fairness of $\langle -0.50, 0.09\rangle$, which not only increases the average fairness value but also reduces the variance across stakeholders. This reduction in dispersion reflects a more equitable distribution of constraint satisfaction, with fewer stakeholders experiencing significantly lower levels of satisfaction.

Table 11 summarizes the fairness scores obtained by using the different optimization strategies. The first two strategies compute the

Table 10
Student fairness divided by degree courses and year, starting from students' timeline.

	Computer Science			Bioinformatics			Biotechnology		
	1	2	3	1	2	3	1	2	3
Fairness	−1.55	−1.02	−1.57	−1.41	−1.15	−1.57	−1	−0.8	−1.55

Table 11
Summary of fairness results across the two optimization strategies. Average and standard deviation are computed across all stakeholders, and the global fairness is reported. The first two strategies are local fair solutions, while the last two strategies are global solutions computed from local fair ones.

Strategy	Professors (avg, std)	Students (avg, std)	Global $\langle g_a(\mathcal{A}), g_s(\mathcal{A})\rangle$
Professors' preferences	−0.49, 0.86	−1.54, 0.58	$\langle -1.02, 1.04\rangle$
Students' preferences	−0.40, 0.83	−1.29, 0.28	$\langle -0.84, 0.87\rangle$
Global from professor	0.0, 0.0	−1.11, 0.22	$\langle -0.56, 0.22\rangle$
Global from students	0.0, 0.0	−1.01, 0.09	$\langle -0.50, 0.09\rangle$

Table 12
Summary of fairness results across the optimization strategies. Average and standard deviation are computed across all stakeholders, and the global fairness is reported.

Strategy	Professors (avg, std)	Students (avg, std)	Global $\langle g_a(\mathcal{A}), g_s(\mathcal{A})\rangle$
Global from students	0.0, 0.0	−1.01, 0.09	$\langle -0.50, 0.09\rangle$
Direct-global	−0.01, 0.04	−1.07, 0.12	$\langle -0.54, 0.12\rangle$
Real	−0.09, 0.21	−1.5, 0.39	$\langle -0.80, 0.45\rangle$

global solution by considering only soft constraints of a single stakeholder group (i.e., students or professors). The latter two strategies begin with a preliminary, local fair assignment and then refine it through global optimization. Interestingly, the *Professors' preferences* configuration, which prioritizes professors' soft constraints, produces the worst professor fairness. This outcome highlights that, even though the professor-oriented solution is locally optimal for professors, combining it with the students' constraints leads to conflicts that need to be solved by identifying some compromises. Conversely, starting from the students' preferences provides a more balanced initialization, where student constraints are aligned with the overall feasibility of the timetable. As a result, the subsequent global optimization step (i.e., *Global from Students*) produces the fairest solution, with both the highest average fairness and the lowest standard deviation. This suggests that a locally fair timetable for students provides a structurally better starting point for global optimization than the professor-oriented one.

Tourism traveling problem. In the tourism scenario, the number of soft constraints defined by guides and tourists is more balanced than in the university use case. As a result, optimizing the schedule from one stakeholder's perspective does not drastically penalize the other. A timetable optimized first for guides, then with FaST-MOSA, with all soft constraints remaining reasonably fair for tourists, improving the global fairness, and vice versa, with no major divergence between the two local optima. This balanced constraint structure implies that starting from a guide-local fair or tourist-local fair solution yields comparable results. Consequently, in the tourism domain, the choice of which stakeholder to optimize first has a marginal effect on the final fairness outcome.

7.2. Global fairness construction

The second option we consider executes the FaST-MOSA algorithm by jointly considering not only all the hard, but also all the soft constraints of all the stakeholders involved. As expected, achieving a global fair solution requires a higher number of iterations, resulting in increased execution time. In the plot of Fig. 5, the trends for global fairness and execution time for the university use case are reported as the number of iterations increases.

The plot is generated by running the algorithm with seven different seeds, demonstrating that the behavior remains consistent across all trials. The best global fairness achieved for the university use case is $\langle -0.54, 0.12\rangle$. In this configuration, only 1 out of 32 has only an unsatisfied constraint, with a standard deviation of 0.04 for the professor and 0.12 for the students.

In Table 12, the global fairness, in terms of the average and standard deviation, for students and professors of the global approach are reported, together with the corresponding best results obtained from the local to global approach and the scores of the real schedule timeline from which the data were extracted.

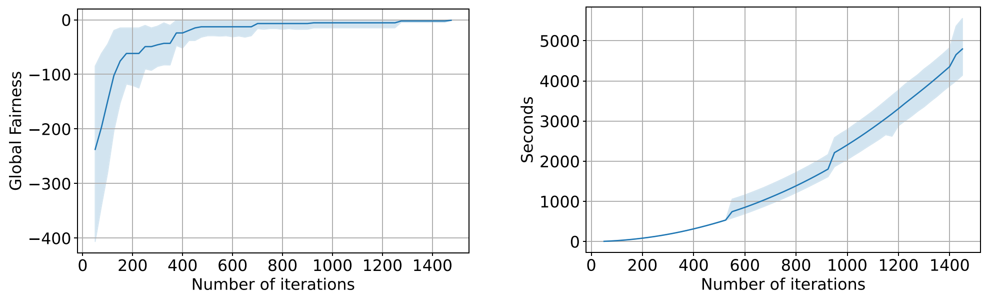
Considering the real timeline, the global fairness is $\langle -0.80, 0.45\rangle$ with a standard deviation of 0.21 for professors and of 0.39 for students. Although the professors show very good average fairness values, this comes at the cost of completely disregarding the students' preferences. Moreover, the standard deviation observed in the real timeline indicates that only a few professors have unmet constraints; in other words, fairness is unevenly distributed, and a minority of professors is disproportionately penalized.

Conversely, the application of FaST-MOSA directly to the global problem improves fairness by reducing the standard deviation of the global score, which captures the dispersion across stakeholders. However, when comparing the approaches under the same computational budget, the strategy that starts from a locally fair solution consistently produces better global fairness outcomes than running the global optimization from scratch. In particular, for the same number of iterations (e.g., an initial temperature of 35 and a maximum number of perturbations of 10), the configuration obtained by starting from the students' preferences and then performing global refinement (*Global from Students*) achieves both a higher average global fairness and a lower standard deviation. Moreover, even when the initial temperature is further reduced to 25, starting from the local solution still yields comparable results ($\langle -0.54, 0.23\rangle$). These results suggest that local initialization provides a more effective starting point for global optimization, enabling the algorithm to achieve fairer solutions more quickly.

A similar behavior is observed in the tourism travel planning problem. The direct global optimization achieves the best overall fairness $\langle -0.88, 0.44\rangle$, this result is obtained at the cost of longer execution times compared to starting from a local solution. Nevertheless, unlike the university scenario, a real-world tourism timeline is not available. Therefore, a direct comparison against an existing schedule cannot be performed.

7.3. Validation in a real use case

To evaluate our approach, we conducted a questionnaire involving 102 Bachelor's and Master's degree students from the University of Verona. This survey serves two different purposes: (i) to evaluate the perceived importance of the objective functions, (ii) to evaluate the goodness of the approach. Regarding the first aspect, we can observe that in the proposed case study, students and tourists are treated as a single entity, and they cannot express individual preferences during the problem instantiation. Therefore, the set of objective functions has been defined to incorporate all usual preferences informally acquired from the domain experts' experience. Clearly, some students may be more



(a) Fairness trend over iterations.

(b) Execution time trend over iterations.

Fig. 5. Evolution of fairness and execution time with increasing iterations.

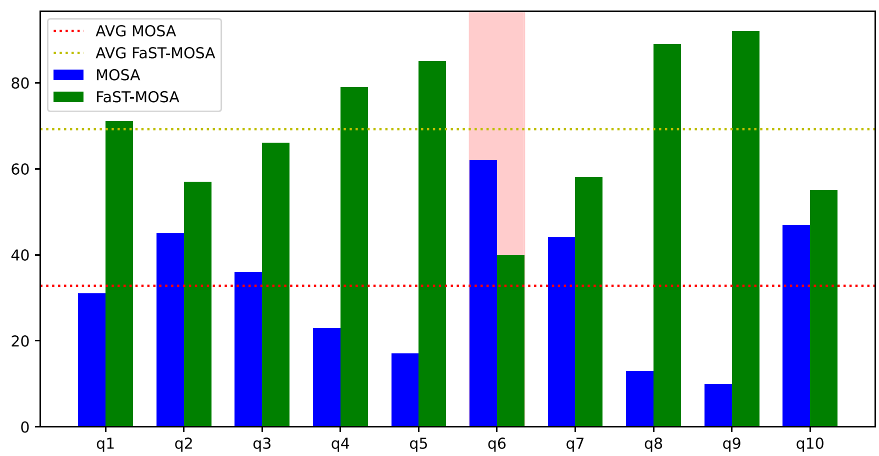


Fig. 6. Timeline chosen by the participants of the questionnaire. In red, the cases in which the participants prefer the timeline generated with MOSA are highlighted. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

sensitive to specific aspects, while others may be more interested in others. With the questionnaire, we can evaluate the perceived importance of the chosen objective functions.

Firstly, we anonymously collected student-type information, as preferences may vary depending on the group to which they belong. Specifically, students were categorized into three groups: commuters, who live more than 40 km from the university (48%); on-site students, who live close to the university (34,3%); and off-site students, who rent an apartment near the university during class periods (17,6%). The classification based on their proximity to the campus seems to be the most informative distinction to understand the importance of the objective functions. Note that this information is used only to understand the obtained results, not to drive them. Indeed, any attempt to consider them during the perturbation could lead to a solution that prioritizes the majority class over the minority one.

After this preliminary question, participants were asked to choose between two timelines and explain their selection. Using the MOSA and FaST-MOSA algorithms, we generated 10 different timelines with both approaches. As shown in Fig. 6, in 9 out of 10 timelines, the students prefer the solution proposed by FaST-MOSA. Looking more in detail at the timeline generated in Table 13, the *q6–MOSA generated* offers a more enjoyable slot for the lunch break than the *q6–FaST-MOSA generated*. Thus, there is no situation of unfairness in none of the two timelines, hence we can consider them as equivalents.

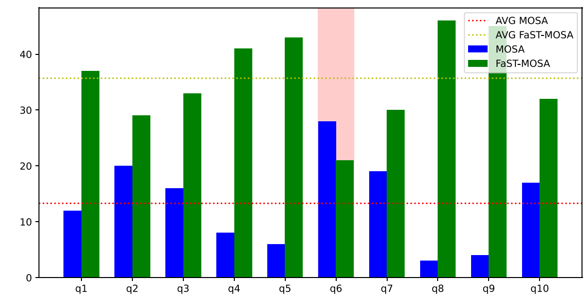
The plots in Fig. 7 display preferences categorized by student groups identified in the first question. Notably, on-site and off-site students, who live near the university, expressed different preferences from commuters in the final question. Examining Table 14, which represents the timeline of *q10*, reveals the cause of this divergence: commuters prefer the timeline with fewer days.

Additionally, participants rank the constraint in order of importance, from most to least significant. The results show that different types of students prioritize constraints differently. Notably, commuters ranked *having fewer days* as their top priority, while *lunch break* consistently placed second or third among all participants. Besides the differences in the importance of these objective functions, the questionnaire confirms the validity of the chosen objective functions, since they are perceived by students as important soft constraints in their preference expression.

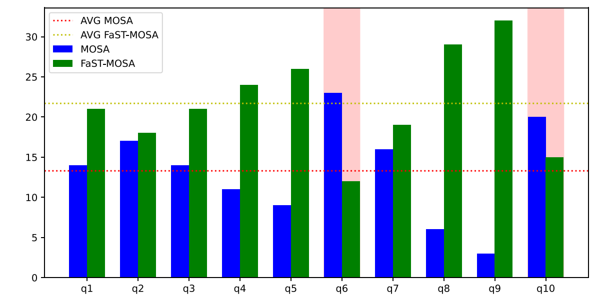
7.4. Discussion

The analysis in Section 7.1 reveals that achieving a local fair solution for each stakeholder is not enough to build a global fair solution or even a valid solution for another stakeholder. In this regard, a multi-objective optimization approach is a powerful tool for identifying a local fair solution, as each preference (transformed into an objective function) can be considered of equal importance in the dominance relation, while the set of common hard constraints defines the notion of a valid solution. Conversely, when considering global fairness, the classical MOSA technique has some limitations, as the dominance relation tends to favor the scenario where one stakeholder increases the corresponding objective function, while the other maintains its satisfaction level, rather than a more equitable situation in which the satisfaction of all stakeholders is similar.

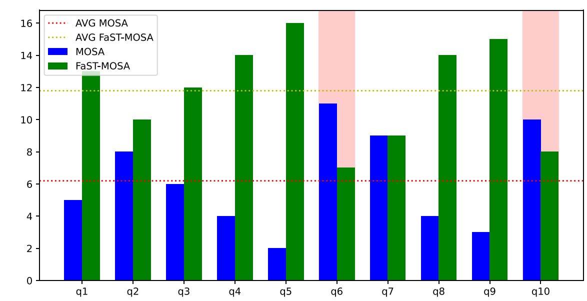
Therefore, the choice to start by identifying local fair solutions is reasonable, but once they have been achieved, a proper procedure



(a) Commuter.



(b) On-site.



(c) Off-site.

Fig. 7. Detailed plots about the timeline chosen from each group of students. The cases where the participants prefer the timeline generated with MOSA are highlighted in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 13
Timelines proposed in the 6th questionnaire question.

<table>
<tr><th colspan="12">(a) MOSA</th></tr>
<tr><th></th><th>8–9</th><th>9–10</th><th>10–11</th><th>11–12</th><th>12–13</th><th>13–14</th><th>14–15</th><th>15–16</th><th>16–17</th><th>17–18</th><th>18–19</th></tr>
<tr><td>Mon</td><td></td><td></td><td></td><td></td><td>$C_{ProgT-1}$</td><td></td><td colspan="2">C_{Math-2}</td><td colspan="2">$C_{ProgT-2}$</td><td></td></tr>
<tr><td>Tue</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Wed</td><td></td><td colspan="2">C_{Net-2}</td><td colspan="2">$C_{ProgL-2}$</td><td></td><td>C_{Net-1}</td><td colspan="3">C_{Math-3}</td><td></td></tr>
<tr><td>Thu</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Fri</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><th colspan="12">(b) FaST-MOSA</th></tr>
<tr><th></th><th>8–9</th><th>9–10</th><th>10–11</th><th>11–12</th><th>12–13</th><th>13–14</th><th>14–15</th><th>15–16</th><th>16–17</th><th>17–18</th><th>18–19</th></tr>
<tr><td>Mon</td><td></td><td></td><td></td><td></td><td colspan="2">$C_{ProgT-2}$</td><td colspan="2">C_{Net-2}</td><td>$C_{ProgT-1}$</td><td></td><td></td></tr>
<tr><td>Tue</td><td></td><td colspan="3">C_{Math-3}</td><td>C_{Net-1}</td><td></td><td colspan="2">$C_{ProgL-2}$</td><td colspan="2">C_{Math-2}</td><td></td></tr>
<tr><td>Wed</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Thu</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Fri</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Table 14
Timelines proposed in the 10th questionnaire question.

<table>
<tr><th colspan="12">(a) MOSA</th></tr>
<tr><th></th><th>8–9</th><th>9–10</th><th>10–11</th><th>11–12</th><th>12–13</th><th>13–14</th><th>14–15</th><th>15–16</th><th>16–17</th><th>17–18</th><th>18–19</th></tr>
<tr><td>Mon</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Tue</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Wed</td><td></td><td></td><td></td><td></td><td colspan="3">C_{Math-3}</td><td>$C_{ProgT-1}$</td><td></td><td></td><td></td></tr>
<tr><td>Thu</td><td></td><td></td><td></td><td></td><td colspan="2">$C_{ProgT-2}$</td><td colspan="2">$C_{ProgL-2}$</td><td></td><td>C_{Net-1}</td><td></td></tr>
<tr><td>Fri</td><td></td><td></td><td></td><td></td><td colspan="2">C_{Math-2}</td><td colspan="2">C_{Net-2}</td><td></td><td></td><td></td></tr>
<tr><th colspan="12">(b) FaST-MOSA</th></tr>
<tr><th></th><th>8–9</th><th>9–10</th><th>10–11</th><th>11–12</th><th>12–13</th><th>13–14</th><th>14–15</th><th>15–16</th><th>16–17</th><th>17–18</th><th>18–19</th></tr>
<tr><td>Mon</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Tue</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Wed</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Thu</td><td></td><td>C_{Net-1}</td><td colspan="2">C_{Math-2}</td><td></td><td></td><td colspan="2">$C_{ProgL-2}$</td><td></td><td></td><td></td></tr>
<tr><td>Fri</td><td></td><td colspan="2">C_{Net-2}</td><td>$C_{ProgT-1}$</td><td></td><td colspan="3">C_{Math-3}</td><td colspan="2">$C_{ProgT-2}$</td><td></td></tr>
</table>

is necessary to combine them to obtain a global fair solution for all stakeholders. In the identification of global fairness, a modification of the notion of dominance and acceptance probability is needed, as well as a measure of global fairness (see Definition 9) which cannot be considered a mere summation or average of the local fairness values $f()$ individually computed. Indeed, the standard deviation of each $f()$ is also included to measure the spread or dispersion of such values. A fair solution is not a solution that completely satisfies one at the expense of others, but a solution that eventually does not save anyone in the same proportion.

7.5. Limitations

In both application domains, the individual preferences of students and tourists were not modeled explicitly, but the soft constraints associated with these stakeholders were defined at an aggregated level, relying on common general preferences, also coming from the literature [42].

The choice to have different granularities in stakeholder characterization and preference definition (e.g., all students in the same cohort together, versus each professor alone) allows us to demonstrate the flexibility of the approach and the possibility of describing preferences at different aggregation levels. However, as discussed in Section 7.3, within the student group, there may be sub-groups with different sensitivities to the chosen objective functions. Moreover, the proposed formalization simplifies the underlying real-world problems in order to focus on the interaction between local and global fairness. For instance, in the university timetable case study, the classroom availability and capacity are not considered. Similarly, in the tourism domain, we do not account for factors such as delays or unexpected changes in visit availability. These simplifications make it possible to initiate the analysis of fairness-related topics, but pose an important direction for future work in refining both hard and soft constraints for applicability to more complex operational settings.

Finally, global fairness is quantified through a specific aggregation that combines average fairness and its standard deviation across stakeholders. This design choice explicitly captures both overall satisfaction and balance, but alternative aggregation strategies could reflect different normative interpretations of fairness. Exploring such alternatives constitutes a natural extension of this work.

8. Conclusion

This work provides a comprehensive framework for multi-sided sequential task assignment that integrates formal definitions of local and global fairness, practical implementation, privacy preservation, and real-world applicability. Indeed, integrating personalized constraints without requiring stakeholders to reveal their motivations is important in today's data-sensitive environment. Discussing potential conflicts between local and global fairness highlights the complexities and trade-offs involved in multi-stakeholder environments. This insight is essential for designing systems that effectively balance individual and collective fairness.

The paper formulates the goal of identifying both the local and global fair solutions as a Multi-Objective Optimization problem and proposes an extension of the MOSA algorithm, called FaST-MOSA, to effectively address both sub-problems by exploiting fairness principles to speed up convergence. The validation of the proposed solution in two real-world domains shows the practical utility and robustness of our approach.

In future work, we are extending the notion of soft constraints to include preferences unrelated to calendar slots. Another possible extension is to consider the impact of external factors, as it is well known that the context greatly enhances the benefits of multi-stakeholder Recommendation Systems. On the other hand, the order in which tasks are completed may be relevant (e.g., hard lectures are better in the morning than in the late afternoon). However, these aspects can be easily implemented through additional constraints that may generalize our proposal.

CRediT authorship contribution statement

Anna Dalla Vecchia: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sara Migliorini:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Elisa Quintarelli:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Kostas Stefanidis:** Supervision, Investigation, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Grammarly in order to correct and improve the grammar and readability of the current manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset and the code are available in the GitHub linked in the manuscript.

References

[1] S. Migliorini, E. Quintarelli, M. Gambini, A. Belussi, D. Carra, Sequence recommendations for groups: A dynamic approach to balance preferences, Inf. Syst. 108 (2022) 102023.
[2] R. Burke, Multisided fairness for recommendation, in: Workshop on Fairness, Accountability, and Transparency in Machine Learning, 2017.
[3] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: AMOSA, IEEE Trans. Evol. Comput. 12 (3) (2008) 269–283, http://dx.doi.org/10.1109/TEVC.2007.900837.
[4] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, fifth ed., Springer, New York, 2016.
[5] P. Brucker, Scheduling Algorithms, second ed., Springer, Berlin, 1999.
[6] A. Allahverdi, The third comprehensive survey on scheduling problems with setup times/costs, European J. Oper. Res. 246 (2) (2015) 345–378.
[7] A. Schaerf, A survey of automated timetabling, Artif. Intell. Rev. 13 (2) (1999) 87–127.
[8] P. Vansteenwegen, A. Gunawan, Operations Research Problems in Tourism, Springer, Cham, 2019.
[9] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Innovations in Theoretical Computer Science Conference, 2012.
[10] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, in: NIPS, 2016.
[11] E. Pitoura, K. Stefanidis, G. Koutrika, Fairness in rankings and recommendations: An overview, VLDB J. (2021).
[12] J. Bulmer, M. Fritter, Y. Gao, B. Hui, FASTT: team formation using fair division, in: Canadian AI, 2020.
[13] G. Barnabò, A. Fazzone, S. Leonardi, C. Schwiegelshohn, Algorithms for fair team formation in online labour marketplaces, in: Companion of WWW, 2019.
[14] L. Machado, K. Stefanidis, Fair team recommendations for multidisciplinary projects, in: WI, 2019.
[15] R. Borges, O. Sahlgren, S. Koivunen, K. Stefanidis, T. Olsson, A. Laitinen, Multi-objective fairness in team assembly, in: ADBIS, 2023.
[16] B. Suman, Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem, Comput. Chem. Eng. 28 (9) (2004) 1849–1871.
[17] S. Rossi, F. Barile, C. Galdi, L. Russo, Recommendation in museums: paths, sequences, and group satisfaction maximization, Multimedia Tools Appl. 76 (2017) 26031–26055.
[18] A. Piliponyte, F. Ricci, J.P.J. Koschwitz, Sequential music recommendations for groups by balancing user satisfaction, in: UMAP 2013 Extended Proceedings: Late-Breaking Results, Project Papers and Workshop Proceedings of the 21st Conference on User Modeling, Adaptation, and Personalization, 2013.
[19] D. Herzog, Recommending a sequence of points of interest to a group of users in a mobile context, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 402–406.
[20] J. Huang, Y. Liu, Y. Chen, C. Jia, Dynamic recommendation of POI sequence responding to historical trajectory, ISPRS Int. J. Geo-Inf. 8 (10) (2019) 433.

[21] Y. Zhao, K. Zheng, Z. Wang, L. Deng, B. Yang, T.B. Pedersen, C.S. Jensen, X. Zhou, Coalition-based task assignment with priority-aware fairness in spatial crowdsourcing, VLDB J. 33 (1) (2024) 163–184.

[22] F. Basık, B. Gedik, H. Ferhatosmanoğlu, K.-L. Wu, Fair task allocation in crowdsourced delivery, IEEE Trans. Serv. Comput. 14 (4) (2021) 1040–1053, http://dx.doi.org/10.1109/TSC.2018.2854866.

[23] Q.C. Ye, Y. Zhang, R. Dekker, Fair task allocation in transportation, Omega 68 (2017) 1–16, http://dx.doi.org/10.1016/j.omega.2016.05.005, URL https://www.sciencedirect.com/science/article/pii/S0305048316302523.

[24] A. Banerjee, P. Banik, W. Wörndl, A review on individual and multistakeholder fairness in tourism recommender systems, Front. Big Data 6 (2023) http://dx.doi.org/10.3389/fdata.2023.1168692.

[25] R. Shafiloo, M. Stratigi, J. Peltonen, T. Olsson, K. Stefanidis, The many facets of fairness in recommender systems: Consumers, providers and items, Inf. Syst. 137 (2026) 102643, http://dx.doi.org/10.1016/j.is.2025.102643, URL https://www.sciencedirect.com/science/article/pii/S0306437925001292.

[26] R. Shafiloo, M. Stratigi, J. Peltonen, K. Stefanidis, Multisided fairness under limited item availability in recommender systems, Inform. Sci. 731 (2026) 122926, http://dx.doi.org/10.1016/j.ins.2025.122926, URL https://www.sciencedirect.com/science/article/pii/S002002552501062X.

[27] H. Wu, C. Ma, B. Mitra, F. Diaz, X. Liu, A multi-objective optimization framework for multi-stakeholder fairness-aware recommendation, ACM Trans. Inf. Syst. 41 (2) (2023) 47:1–47:29.

[28] R. Burke, N. Mattei, V. Grozin, A. Voida, N. Sonboli, Multi-agent social choice for dynamic fairness-aware recommendation, in: UMAP '22: 30th ACM Conference on User Modeling, Adaptation and Personalization, Barcelona, Spain, July 4 - 7, 2022, Adjunct Proceedings, ACM, 2022, pp. 234–244.

[29] G. Balakrishnan, W. Wörndl, Multistakeholder recommender systems in tourism, in: J. Neidhardt, W. Wörndl, T. Kuflik, M. Zanker (Eds.), Proceedings of the Workshop on Recommenders in Tourism Co-Located with the 15th ACM Conference on Recommender Systems (RecSys 2021), Online, September 26, 2021, in: CEUR Workshop Proceedings, vol. 2974, CEUR-WS.org, 2021, pp. 39–53.

[30] M. Mühlenthaler, M. Mühlenthaler, Fairness in Academic Course Timetabling, Springer, 2015.

[31] A. Muklason, A.J. Parkes, E. Özcan, B. McCollum, P. McMullan, Fairness in examination timetabling: Student preferences and extended formulations, Appl. Soft Comput. 55 (2017) 302–318, http://dx.doi.org/10.1016/j.asoc.2017.01.026, URL https://www.sciencedirect.com/science/article/pii/S1568494617300418.

[32] A.H. Khan, T. Imtiaz, A novel genetic algorithm based timetable generator for optimized university timetable solution, in: 2024 International Conference on Engineering & Computing Technologies, ICECT, 2024, pp. 1–6, http://dx.doi.org/10.1109/ICECT61618.2024.10581296.

[33] T. Müller, H. Rudová, Z. Müllerová, Real-world university course timetabling at the International Timetabling Competition 2019, J. Sched. 28 (2) (2025) 247–267, http://dx.doi.org/10.1007/S10951-023-00801-W.

[34] A.D. Selbst, D. Boyd, S.A. Friedler, S. Venkatasubramanian, J. Vertesi, Fairness and abstraction in sociotechnical systems, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, in: FAT* '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 59–68, http://dx.doi.org/10.1145/3287560.3287598.

[35] R. Burke, N. Mattei, V. Grozin, A. Voida, N. Sonboli, Multi-agent social choice for dynamic fairness-aware recommendation, in: Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, in: UMAP '22 Adjunct, Association for Computing Machinery, New York, NY, USA, 2022, pp. 234–244, http://dx.doi.org/10.1145/3511047.3538032.

[36] G.K. Patro, A. Biswas, N. Ganguly, K.P. Gummadi, A. Chakraborty, FairRec: Two-sided fairness for personalized recommendations in two-sided platforms, in: Proceedings of the Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1194–1204, http://dx.doi.org/10.1145/3366423.3380196.

[37] T. Sühr, A.J. Biega, M. Zehlike, K.P. Gummadi, A. Chakraborty, Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 3082–3092, http://dx.doi.org/10.1145/3292500.3330793.

[38] T. Müller, H. Rudová, Z. Müllerová, University course timetabling and International Timetabling Competition 2019, in: Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling, PATAT 2018, 2018, pp. 5–31.

[39] C. Bettini, X.S. Wang, S. Jajodia, Temporal granularity, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, 2009, pp. 2968–2973, http://dx.doi.org/10.1007/978-0-387-39940-9_397.

[40] E. Aarts, J. Korst, Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing, John Wiley & Sons, Inc., 1989.

[41] E.H.L. Aarts, J.H.M. Korst, Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing, John Wiley & Sons, New York, NY, USA, 1989.

[42] E. Rappos, E. Thiémard, S. Robert, J.-F. Hêche, A mixed-integer programming approach for solving university course timetabling problems, J. Sched. 25 (4) (2022) 391–404.