Fairness in Academic Course Timetabling

Moritz Mühlenthaler · **Rolf Wanka**

Abstract We consider the problem of creating *fair* course timetables in the setting of a university. Our motivation is to improve the overall satisfaction of individuals concerned (students, teachers, etc.) by providing a fair timetable to them. The central idea is that undesirable arrangements in the course timetable, i. e., violations of soft constraints, should be distributed in a fair way among the individuals. We propose two formulations for the fair course timetabling problem that are based on max-min fairness and Jain's fairness index, respectively. Furthermore, we present and experimentally evaluate an optimization algorithm based on simulated annealing for solving max-min fair course timetabling problems. The new contribution is concerned with measuring the energy difference between two timetables, i. e., how much worse a timetable is compared to another timetable with respect to max-min fairness. We introduce three different energy difference measures and evaluate their impact on the overall algorithm performance. The second proposed problem formulation focuses on the tradeoff between fairness and the total amount of soft constraint violations. Our experimental evaluation shows that the known best solutions to the ITC2007 curriculum-based course timetabling instances are quite fair with respect to Jain's fairness index. However, the experiments also show that the fairness can be improved further for only a rather small increase in the total amount of soft constraint violations.

Keywords Curriculum-based Course Timetabling, Max-Min Fairness, Fairness Index

1 Introduction

We consider the problem of creating fair course timetables in the setting of a university. In academic timetabling, the courses need to be assigned to a limited number of resources (rooms and timeslots) such that certain constraints are satisfied There are typically two kinds of constraints called hard and soft constraints. The hard constraints are basic requirements, so a timetable which does not satisfy all hard constraints is considered useless. The

Research funded in parts by the School of Engineering of the University of Erlangen-Nuremberg.

Moritz Mühlenthaler, Rolf Wanka
Department of Computer Science, University of Erlangen-Nuremberg, Germany
E-mail: {moritz.muehlenthaler, rwanka}@cs.fau.de

soft constraints characterize certain undesirable properties of a timetable for students and teachers, as well as for abstract entities such as courses and curricula. The quality of a timetable is determined by the extent to which soft constraints are violated: the fewer soft constraint violations, the better. The usual approach is to use an objective function which penalizes the soft constraint violations so the goal is to find a timetable with a minimal total penalty. Situations may arise however, in which a large part of penalty hits only a small group of individuals, who would thus receive a poor timetable in comparison to others. In other words, a timetable may be unfair due to an unequal distribution of penalty.

Fairness and inequality in the distribution of resources are a major concern for instance in economics [15,36] and computer networks [4–6,17,18,20,31,35]. In the area of operations research, fairness criteria have been applied for example to the aircraft landing problem [37]. To the best of our knowledge, no previous paper on academic timetabling addresses fairness explicitly. In this paper, we investigate two approaches that avoid unfair distributions of penalty in the context of academic course timetabling. The first approach uses a purely qualitative measure of fairness, i. e., given two timetables the fairness measure determines which of the two is better. In contrast, the second approach is based on a quantitative fairness measure, which represents the fairness of a timetable as a number between zero and one.

The first approach considers groups of students ranked by the quality of the course timetable from the students' perspective. The goal is to improve the student satisfaction by imposing the following fairness conditions: The courses should be assigned to rooms and timeslots such that the worst course schedule for any of the students is as good as possible with respect to the various soft constraints. Under this condition, the second-worst course schedule for any student should be as good as possible, and so forth. This fairness concept is called lexicographic max-min fairness. For the sake of succinctness, we will refer to this fairness concept just as max-min fairness. In the literature, max-min fairness has for example been applied to network bandwidth allocation problems [31,35]. In this work, we propose the MMF-CB-CTT problem model, a max-min fair variant of the popular curriculum-based course timetabling (CB-CTT) problem formulation from [13]. We further propose MaxMinFair_SA, an optimization algorithm based on simulated annealing (SA), for solving max-min fair optimization problems. Although our evaluation of MaxMinFair_SA focuses on MMF-CB-CTT problems, the algorithm can be tailored to other max-min fair problems by choosing an appropriate neighborhood exploration mechanism and a suitable evaluation function. A delicate part of the algorithm is the energy difference function, which quantifies how much worse one solution is compared to another solution. We propose three different energy difference functions and evaluate their impact on the performance of MaxMinFair_SA on the 21 standard instances from [13].

The fairness conditions imposed by max-min fairness are rather strict in the sense that no tradeoff arises between fairness and total penalty. When creating course timetables for a university, however, it may be desirable to pick a timetable from a number of solutions with varying tradeoffs between fairness and total penalty. Our second proposed approach offers this flexibility. The approach is based on a bi-criteria problem formulation which includes fairness as an option, but does not enforce it like max-min fair optimization. In this problem formulation, referred to as JFI-CB-CTT, we use Jain's fairness index, an inequality measure proposed by Jain et al. in [17]. The fairness index allows us to quantify the fairness of a timetable explicitly. Please note that since max-min fairness is a purely qualitative fairness measure, it is not applicable in this setting. We investigate the tradeoffs between fairness and total penalty for the six standard instances from [13] whose known best solutions have the highest total penalty compared to the other instances. Our motivation for this choice of instances is simply that if the total penalty of a timetable is very small, then there is not much gain in distributing the penalty in a fair way. Our conclusion regarding this approach is that, although the known best solutions for the six instances are quite fair, we can improve the fairness further with only a rather small increase in total penalty. For a theoretical treatment of the price of fairness on so-called convex utility sets with respect to proportional fairness and max-min fairness, see the recent work by Bertsimas et al. [6].

The remainder of this paper is organized as follows. In Section 2, we will provide a brief review of the curriculum-based course timetabling (CB-CTT) problem model as well as the two fairness concepts max-min fairness and Jain's fairness index. In Section 3 we will propose two fair variants of the CB-CTT model, and in Section 4, we will introduce the optimization SA-based algorithm MAXMINFAIR_SA for solving max-min fair allocation problems. Section 5 is dedicated to our experimental evaluation of the fairness of the known best solutions to 21 standard instances from [13] with respect to max-min fairness and Jain's fairness index, and the performance of the MAXMINFAIR_SA algorithm.

2 Preliminaries

In this section, we provide a brief review of the curriculum-based course timetabling problem formulations as well as relevant definitions concerning max-min fairness and Jain's fairness index.

2.1 Curriculum-based Course Timetabling Problems

Curriculum-based Course Timetabling (CB-CTT) is a problem formulation for a class of optimization problems which arise when creating course schedules in the setting of a university. A central entity in the problem model is the *curriculum*. Each curriculum consists of a set of courses which must be attended by a common group of students and thus must not be held simultaneously. Our experimental evaluation of fairness in academic timetabling is based on the CB-CTT problem formulation introduced for Track 3 of the Second International Timetabling Competition (ITC2007) [27]. This formulation has emerged as one of the standard problem formulations in academic timetabling – both in research and in practice.

CB-CTT problems are NP-hard and a lot of effort has been devoted to finding heuristic approaches which provide high quality solutions within reasonable time. A wide range of techniques has been employed for solving CB-CTT instances including but not limited to approaches based on Max-SAT [2], mathematical programming [24, 8], local search [12, 26], evolutionary computation [1] as well as hybrid approaches [29]. There has been a lot of progress in terms of the achieved solution quality in the recent years. Interestingly however, there seems to be no single approach which is superior to the other approaches on all (or even most) ITC2007 instances (see [13] for current results).

A CB-CTT instance consists of the following data: We are given a set of days and each day is divided into a fixed number of timeslots. A pair composed of a day and a timeslot will be referred to as *period*. A period in conjunction with a room is called a *resource*. Additionally, we are given sets of teachers, courses, rooms and curricula. Each course has a teacher and consists of a number of lectures; each *curriculum* is a set of courses. For each room, we are provided with the maximum number of students it can accommodate. Now, given a CB-CTT instance I, the task is to create a course timetable τ, i. e., to find an assignment of lectures to resources such that hard constraints are satisfied and soft constraint violations are minimal. A comprehensive description of the problem model and the rationale behind it can be found in [11].

A timetable which satisfies all hard constraints is called *feasible*. The hard constraints ensure that no student and no teacher has to be present at two lectures at the same time and no two lectures can occupy a room at the same time. Additionally, all lectures have to be assigned to a suitable resource and, for each lecture, the teacher has to be available in the given period. For our purpose of creating fair course timetables, we will consider the distribution of soft constraint violations among the curricula. The CB-CTT formulation features the following four soft constraints:

- S1 *RoomCapacity*: Each lecture should be assigned to a room of sufficient size.
- S2 *MinWorkingDays*: The lectures of each course should be distributed over a certain minimum number of days.
- S3 *IsolatedLectures*: For each curriculum, all lectures associated to the curriculum should be scheduled in adjacent timeslots.
- S4 *RoomStability*: The lectures of each course should be assigned to the same room.

Each violation of one of the soft constraints results in a penalty for the timetable. The objective function aggregates individual penalties by taking their weighted sum. Detailed descriptions of how hard and soft constraints are evaluated and how much penalty is applied for a particular soft constraint violation can be found in [11].

2.2 Fairness in Resource Allocation

Fairness issues typically arise when scarce resources are allocated to a number of individuals with demands. E. g., fair resource allocation has received much attention in economic theory [15], but also occurs in a wide range of applications in computer science including bandwidth allocation in networks [5] and task scheduling [34]. In many optimization problems related to resource allocation, the goal is to maximize the amount of resources allocated to each individual. Fairness conditions can be imposed implicitly or explicitly in order to prevent unfair distributions of the allocated resources.

Consider a resource allocation problem with n entities or, in the following, *individuals* receiving resources. A particular resource allocation (an admissible solution) induces an allocation vector $X = (X_1, \ldots, X_n)$, where each item $X_i, 1 \leq i \leq n$, corresponds to the amount of resources allocated to individual i. There are various approaches to determining the fairness of a resource allocation from the corresponding allocation vector. A fairness concept which allows for a qualitative comparison of two allocation vectors is (lexicographic) max-min fairness. It has received attention in the area of network engineering, in particular in the context of flow control [4,20,35,41]. Another class of approaches are inequality measures such as the Gini index [16] and Jain's fairness index [17,25]: an inequal distribution of resources is considered unfair. The inequality of a resources distribution is typically represented as a number, which allows for a quantitative comparison of the fairness of resource allocations. Inequality measures have been studied in economics [15], in particular in the context of income distribution. Furthermore, in many resource allocation problems, the notion of fairness is implicitly contained in the objective function. Depending on the notion of fairness, the task can be to find allocations maximizing or minimizing the sum of the individual allocations, the mean allocation, the root mean square (RMS), the smallest allocation, and so forth [30,37].

Our evaluation of fairness in academic course timetabling focuses on the two fairness criteria max-min fairness and Jain's fairness index.

Max-min Fairness. Max-min fairness can be stated as iterated application of Rawls's Second Principle of Justice [33]:

> "Social and economic inequalities are to be arranged so that they are to be of greatest benefit to the least-advantaged members of society." (the Difference Principle)

Once the status of the least-advantaged members has been determined according to the difference principle, it can be applied again to everyone except the least-advantaged group in order to maximize the utility (in the economic sense) for the second least-advantaged members, and so on. The resulting utility assignment is called max-min fair. A max-min fair utility assignment implies that no member can improve its utility at the expense of any other member who received less utility. A max-min fair resource allocation is Pareto-optimal.

In order to define max-min fairness more formally, we introduce some notation. Let X be an allocation vector. Then let $\vec{X}$ be the corresponding vector containing the values of X arranged in nondecreasing order. Let Y be another allocation vector. We write $X \preceq_{mm} Y$ if X is at least as good as Y in the max-min sense, that is if $\vec{Y} \preceq_{lex} \vec{X}$, where $\preceq_{lex}$ is the usual lexicographic comparison. A resource allocation X is called max-min optimal, if $X \preceq_{mm} Y$ holds for all feasible resource allocations Y. Since the allocations are sorted, max-min fairness does not discriminate between individuals, but only between the amounts of resources assigned to them.

A weaker version of max-min fairness results if the fairness conditions are not applied iteratively as stated in the above definition. This means that we are just concerned with chosing the best possible outcome for the least-advantaged individuals. In the literature, related optimization problems are referred to as bottleneck optimization problems [14,32]. In the context of academic timetabling however, this weaker fairness concept does not lead to desirable results: although, from the perspective of the least-advantaged group a timetable may be optimal, the quality of the timetable from the perspective of other stakeholders is not considered.

Jain's Fairness Index. While max-min fairness enforces a certain efficiency in resource utilization and provides a qualitative measure of fairness, Jain's fairness index [17] quantifies the inequality of a given resource distribution. An equal distribution of resources is considered fair, while an inequal distribution is considered unfair. It is the crucial fairness measure that is applied in the famous AIMD algorithm used in TCP Congestion Avoidance [10]. The fairness index $J(X)$ of an allocation vector X is defined as

$$J(X) = \frac{\Big(\sum\limits_{1\le i\le n} X_i\Big)^2}{n \cdot \sum\limits_{1\le i\le n} X_i^2} \quad . \tag{1}$$

It has several useful properties like population size independence, scale and metric independence, it is bounded between 0 and 1, and it has an intuitive interpretation. In particular $J(X) = 1$ means that X is a completely fair allocation, i. e., the allocation is fair for every individual, and if $J(X) = 1/n$ then all resources are occupied by a single individual. Furthermore, if $J(X) = x\%$ then the allocation X is fair for x percent of the individuals.

3 Fairness in Academic Course Timetabling

Course timetabling problems fit quite well in the framework of fair resource allocation problems described in the previous section: A timetable is an allocation of resources (rooms, timeslots) to lectures. In this section, we will define two fair versions of the CB-CTT problem model. The first one, MMF-CB-CTT, is based on max-min fairness. Since max-min fairness enforces efficiency (maximum utility) as well as fairness at least to some extent, it is not a suitable concept for exploring the tradeoff between fairness and efficiency. Therefore we propose a second fair variant of CB-CTT called JFI-CB-CTT that is based on Jain's fairness index.

In order to employ the fairness concepts mentioned in the previous section, we need to define how to determine the allocation vector for a timetable. The central entities in the CB-CTT problem model are the curricula. Therefore, in this work, we are interested in a fair distribution of the penalty values assigned to the curricula. Please note that we interpret utility as the opposite of penalty. Hence, a timetable which receives less penalty than another timetable has a higher utility. We achieve the transformation from penalty to utility by simply changing the signs of the penalty values. Let I be a CB-CTT instance with curricula $c_1, c_2, \ldots, c_k$ and let f_c be the usual CB-CTT objective function from [11], which evaluates (S1)-(S4) restricted to curriculum c. This means f_c determines soft constraint violations only for the courses in curriculum c. For a timetable t the corresponding allocation vector is given by the allocation function

$$A(t) = (-f_{c_1}(t), -f_{c_2}(t), \ldots, -f_{c_k}(t)) \quad . \tag{2}$$

Definition 1 (MMF-CB-CTT) Given a CB-CTT instance I, the task is to find a feasible timetable t such that $A(t)$ is max-min optimal.

If a feasible timetable corresponds to a max-min optimal allocation, then any curriculum c could receive less penalty only at the expense of other curricula which receive more penalty than c. Since each student is struck only by the penalty assigned to his or her curriculum the group of students with the worst timetable receive the best possible timetable and under this condition, the students with the second-worst timetable receive the best possible timetable, and so on.

In order to explore the tradeoff between efficient and fair resource allocation in curriculum-based timetabling, we propose another fair variant of CB-CTT called JFI-CB-CTT that is based on Jain's fairness index [17]. In order to get meaningful results from the fairness index however, we need a different allocation function. Consider an allocation X, where all penalty is allocated to a single curriculum while the remaining $k-1$ curricula receive no penalty. Then $J(X) = 1/k$, which means that only one curriculum is happy with the allocation (see [17]). In our situation however, the opposite is the case: $k-1$ curricula are happy since they receive no penalty at all. The following allocation function shifts the penalty values such that the corresponding fairness index in the situation described above becomes $(k-1)/k$, which is in much better agreement with our intuition:

$$A'(t) = (f_{max} - f_{c_1}(t), f_{max} - f_{c_2}(t), \ldots, f_{max} - f_{c_k}(t)) \quad , \tag{3}$$

with

$$f_{max} = \max_{1 \le i \le k} \{f_{c_i}(t)\} \quad .$$

Definition 2 (JFI-CB-CTT) Given a CB-CTT instance I, the task is to find the set of feasible solutions which are Pareto-optimal with respect to the two objectives of the objective function

$$F(t) = (f(t), 1 - J(A'(t))) \quad , \tag{4}$$

where f is the CB-CTT objective function from [11] and J is defined in Eq. (1).

In a similar fashion, other classes of timetabling problem such as post-enrollment course timetabling, exam timetabling and nurse rostering can be turned into fair optimization problems. For example, for post-enrollment course timetabling, the central entities of interest would likely be the individual students, not the curricula as for CB-CTT problems. Therefore, the goal would be a fair distribution of penalty over all students. Once an appropriate allocation function has been def ned, we immediately get the corresponding fair optimization problems.

Our proposed problem formulations are concerned with balancing the interests within one group of indiviuals, namely the students. In practice however, there are often several groups of individuals with possibly conf icting interest, for example students, lecturers and administration. The proposed models can be extended to multiple groups of stakeholders in a straight-forward manner: The penalty values for all individuals (in all groups) are stored in the allocation vector, and since all values are just penalty values, the fairness concepts can be applied as proposed above. This approach requires some additional thought however, since the interest of all individuals are considered to be equally important, which may or may not be intended in practice (weighting can be applied of course). Another approach, which avoids the problem of giving explicit priorities to the interests of different groups extends the problem formulation based on Jain's fairness index: The fairness index can be determined independently for each group and the problem model can be extended to a $(d+1)$-objective optimization problem, where d is the number of groups of stakeholders under consideration. The set of Pareto-optimal solutions characterizes the tradeoffs between the interests of the different groups and the total penalty.

4 Simulated Annealing for Max-Min Fair Course Timetabling and Three Measures for Energy Difference

Simulated Annealing (SA) is a popular local search method which works surprisingly well on many problem domains [19]. SA has been applied successfully to timetabling problems [21,38] and some of the currently known best solutions to CB-CTT instances from the ITC2007 competition were discovered by simulated annealing-based methods [13]. Our SA for max-min fair optimization problems shown in Algorithm 1 below (algorithm MAXMINFAIR_SA) is conceptually very similar to the original SA algorithm proposed by Kirkpatrick et al. [19]. Since max-min fairness only tells us which of two given solutions is better, but not how much better, the main challenge in tailoring SA to max-min fair optimization problems is to f nd a suitable energy difference function, which quantif es the difference in quality between two candidate solutions. In the following, we propose three different energy difference measures for max-min fair optimization and provide details on the acceptance criterion, the cooling schedule, and the neighborhood exploration method used for the experimental evaluation of MAXMINFAIR_SA in the next section.

Acceptance Criterion. Similar to the original SA algorithm proposed by Kirkpatrick et al. in [19], algorithm MAXMINFAIR_SA accepts an improved or equally good solution s_{next} with probability 1. If s_{next} is worse than s_{cur} then the acceptance probability depends on the current temperature level ϑ and the energy difference ΔE. The energy difference measures the difference in quality of the allocation induced by s_{next} compared to the allocation induced

Algorithm 1: MAXMINFAIR_SA

input : s_{cur}: feasible timetable, ϑ_{max}: initial temperature, ϑ_{min}: final temperature, timeout
output: s_{best}: Best feasible timetable found so far

```
s_best ← s_cur
ϑ ← ϑ_max
while timeout not hit do
    s_next ← neighbor(s_cur)
    if P_accept ≥ random() then s_cur ← s_next
    if A(s_cur) ⪯_mm A(s_best) then s_best ← s_cur
    ϑ ← next_temperature(ϑ)
end
return s_best
```

by the current solution s_{cur}. The acceptance probability P_{accept} is defined as:

$$P_{accept} = \begin{cases} 1 & \text{if } s_{next} \preceq_{mm} s_{cur} \\ \exp\big(-\dfrac{\Delta E(X,Y)}{\vartheta}\big) & \text{otherwise,} \end{cases}$$

where $X = A(s_{cur})$ and $Y = A(s_{next})$. With max-min fair optimization in mind, we propose the following three energy different measures for the energy difference, ΔE_{lex}, ΔE_{cw}, and ΔE_{ps}, which are based on lexicographic comparison, component-wise ratios and the ratios of the partial sums of the sorted allocation vectors, respectively. Our experiments presented in the next section indicate that choosing one energy difference function over another has a clear impact on the performance of Algorithm MAXMINFAIR_SA. Hence the choice of the energy difference function is a critical design choice.

For two allocation vectors X and Y of length n, let the energy difference ΔE_{lex} be (note $\vec{X}_i$ denotes the ith entry after sorting the entries of X, $\vec{Y}_i$ is defined analoguously)

$$\Delta E_{lex}(X,Y) = 1 - \frac{1}{n} \cdot \left(\min_{1 \le i \le n} \left\{ i \mid \vec{X}_i > \vec{Y}_i \right\} + 1 \right) \quad . \tag{5}$$

ΔE_{lex} determines the energy difference between X and Y from the index of the sorted allocation vectors at which the comparison $X \preceq_{mm} Y$ shows that X is better than Y. Thus, sorted allocation vectors which differ at the most significant indices have a higher energy difference than those which differ at later indices. In order to make the numerical range of ΔE_{lex} independent of the actual size of the allocation vector, which may vary from instance to instance, the result is normalized by the length $n$of the allocation vectors.

ΔE_{lex} only considers the earliest index at which two sorted allocation vectors differ but ignores how much the actual entries differ. We additionally propose the two energy difference measures ΔE_{cs} and ΔE_{ps} which take this information into account. These two energy difference measures were inspired by the definitions of approximation ratios for max-min fair allocation problems given by Kleinberg et al. in [20]. An approximation ratio is a measure for how much worse the quality of a solution is relative to a possibly unknown optimal solution. In our case, we are interested in how much worse one given allocation is relative to another given allocation. Despite the different context, we can use the same general ideas. Let $\mu_{X,Y}$ be the smallest value of the two allocation vectors X and Y offset by a parameter $\delta > 0$, i. e.,

$$\mu_{X,Y} = \min\{\vec{X}_1, \vec{Y}_1\} - \delta \quad . \tag{6}$$

The component-wise energy difference ΔE_{cw} of two allocations X and Y is defined as:

$$\Delta E_{cw}(X,Y) = \max_{1\le i\le n}\left\{\frac{\mu_{X,Y} - \vec{X}_i}{\mu_{X,Y} - \vec{Y}_i}\right\} - 1 \tag{7}$$

Unlike ΔE_{lex}, the component-wise energy difference does not take into account explicitly which entries of the sorted allocation vectors are responsible for $X \preceq_{mm} Y$. There is however a bias towards the ratios of entries which occur early in the sorted allocation vectors. Since all entries are subtracted from $\mu_{X,Y}$, the ratios of the most significant entries with respect to $\preceq_{mm}$ tend to govern the value component-wise energy difference. Consider for example the situation that Y is much worse than X, say, $\min\{\vec{X}_1, \vec{Y}_1\}$ occurs more often in X than in Y. Then for $\delta \ll 1$ the energy difference $\Delta E_{cw}(X,Y)$ becomes large. On the other hand, if X is nearly as good as Y then the ratios are all close to one and thus $\Delta E_{cw}(X,Y)$ is close to zero.

The third proposed energy difference measure ΔE_{ps} is based on the ratios of the partial sums $\sigma_i(X)$ of the sorted allocation vectors.

$$\sigma_i(X) = \sum_{1\le j\le i} X_j \ .$$

The intention of using partial sums of the sorted allocations is to give the individuals who receive the most penalty, and hence occur early in the sorted allocation vectors, more influence on the resulting energy difference compared to ΔE_{cw}. The energy difference ΔE_{ps} is defined as

$$\Delta E_{ps}(X,Y) = \max_{1\le i\le n}\left\{\frac{i\cdot\mu_{X,Y} - \sigma_i(\vec{X})}{i\cdot\mu_{X,Y} - \sigma_i(\vec{Y})}\right\} - 1 \ . \tag{8}$$

Cooling Schedule. In algorithm MAXMINFAIR_SA, the function `next_temperature` updates the current temperature level ϑ according to the cooling schedule. We use a standard geometric cooling schedule

$$\vartheta = \alpha^t \cdot \vartheta_{max} \ ,$$

where α is the cooling rate and t is the elapsed time. Geometric cooling schedules decrease the temperature level exponentially over time. It is a popular class of cooling schedules which is widely used in practice and works well in many problem domains including timetabling problems [23,22,39]. Geometric cooling was chosen due to its simplicity, since the main focus of our evaluation in Section 5 is the performance impact of the different energy difference functions. We have made a slight adjustment to the specification of the geometric cooling schedule in order to make the behavior more consistent for different timeouts. Instead of specifying the cooling rate α, we determine α from ϑ_{max}, the desired minimum temperature ϑ_{min} and the timeout according to:

$$\alpha = \left(\frac{\vartheta_{min}}{\vartheta_{max}}\right)^{\frac{1}{\text{timeout}}} \ . \tag{9}$$

Thus, at the beginning ($t = 0$) the temperature level is ϑ_{max} and when the timeout is reached ($t =$ timeout), the temperature level becomes ϑ_{min}. We chose to set a timeout rather than a maximum number of iterations since this setting is compliant with the ITC2007 competition conditions, which are a widely accepted standard for comparing results.

Table 1: Fairness of the known best timetables from [13] for the ITC2007 CB-CTT instances.

Instance	Curricula	$f(s_{best})$	$J(A'(s_{best}))$	$-\vec{A}(s_{best})$
comp01	14	5	0.8571	$5^2, 0^{12}$
comp02	70	24	0.9515	$4, 2^{10}, 0^{59}$
comp03	68	66	0.9114	$13, 10^3, 9, 7^2, 6^4, 5^{13}, 4, 2^6, 0^{37}$
comp04	57	35	0.8964	$7, 6^3, 5^4, 4^2, 2, 0^{46}$
comp05	139	291	0.8277	$41^2, 36^7, 35^5, 32^5, 31^6, 30^9, 28, 27^7, 26^2, 25^{14}, \ldots, 2, 0^3$
comp06	70	27	0.9657	$12, 7^2, 5^4, 2^3, 0^{60}$
comp07	77	6	0.9870	$6, 0^{76}$
comp08	61	37	0.9020	$7, 6^3, 5^4, 4^2, 2^2, 0^{49}$
comp09	75	96	0.8047	$10^5, 9, 7^{10}, 6^6, 5^{10}, 4, 2, 0^{41}$
comp10	67	4	0.9701	$2^2, 0^{65}$
comp11	13	0	–	0^{13}
comp12	150	300	0.9128	$45, 30^{14}, 28, 27^2, 26^5, 25^{19}, 22^4, 21^6, 20^8, 19, \ldots, 2^2, 0^3$
comp13	66	59	0.8830	$8, 7, 6^5, 5^7, 4^2, 2^3, 0^{47}$
comp14	60	51	0.9023	$8^4, 7, 5^2, 2^6, 0^{47}$
comp15	68	66	0.8495	$10^3, 9^3, 7, 6^4, 5^{13}, 4, 2^7, 0^{36}$
comp16	71	18	0.9176	$7^2, 5^7, 4, 0^{61}$
comp17	70	56	0.9248	$10^2, 6^3, 5^9, 2^4, 0^{52}$
comp18	52	62	0.9009	$17, 15, 14, 13, 11, 10, 9^2, 5^{19}, 2^2, 0^{23}$
comp19	66	57	0.9612	$13, 7, 6^4, 5^2, 4, 2^7, 0^{50}$
comp20	78	4	0.9744	$2^2, 0^{76}$
comp21	78	76	0.8838	$12, 11, 10^4, 9, 7^4, 6^4, 5^{12}, 4, 2^3, 1^2, 0^{45}$

Neighborhood. In our max-min fair SA implementation, the function `neighbor` picks at random a neighbor in the Kempe-neighborhood of s_{cur}. The Kempe-neighborhood is the set of all timetables which can be reached by performing a single Kempe-move, which is a well-known and widely used operation for swapping events in a timetable [7,26,28,39,40]. A prominent feature of Kempe-moves is that they preserve the feasibility of a timetable. Since the algorithm MAXMINFAIR_SA only uses Kempe-moves to modify timetables the output is guaranteed to be feasible if the input timetable is feasible. In the future, more advanced neighborhood exploration methods similar to those proposed for example in [12,26] could be used, which may well lead to an improved overall performance of MAXMINFAIR_SA.

5 Evaluation

In this section, we will f rst address the question how fair or unfair the known best timetables for the ITC2007 CB-CTT instances are with respect to Jain's fairness index and max-min fairness. Table 1 shows our measurements of how fair the best existing solutions to the CB-CTT instances `comp01`, `comp02`, ..., `comp21` are (see [13] for instance data). Please note that the known best timetables were not created with fairness in mind, but the objective was to create timetables with minimal total penalty. In Table 1, s_{best} refers to the known best solution for each instance. A and A' refer to the allocation functions given in (2) and (3), respectively. The data indicates that the timetables with a low total penalty are also rather fair. This can be explained by the fact that these timetables do not have a large amount of penalty to distribute over the curricula. Thus, most curricula receive little or no penalty and consequently, the distribution is fair for most curricula. We will show in Section 5.2

Table 2: The performance of MAXMINFAIR_SA with $\Delta E = \Delta E_{cw}$ for different values of δ.

δ	10^0	10^{-2}	10^{-3}	10^{-6}
10^0	–	02	02,05	02
10^{-2}	10,19,20	–	09	19
10^{-3}	01,10,19,20	–	–	03,19
10^{-6}	01,10,20	–	–	–

however, that for timetables with a comparatively large total penalty there is still some room for improvement concerning fairness.

The rightmost column of Table 1 contains the sorted allocation vectors of the best solutions. For a more convenient presentation, all entries of the sorted allocation vectors are multiplied by -1. The exponents denote how often a certain number occurs. For example, the sorted allocation vector $(-5,-5,0,0,0)$ would be represented as $5^2,0^3$. The sum of the values of an allocation vector is generally much larger than the total penalty shown in the second column. The reason for this is that the penalty assigned to a course is counted for each curriculum the course belongs to. With a few exceptions the general theme seems to be that the penalty is assigned to only a few curricula while a majority of curricula receives no penalty. In the next section we will show that the situation for the curricula which receive the most penalty can be improved with max-min fair optimization for 15 out of 21 instances.

5.1 Max-Min Fair Optimization

In Section 4, we presented a SA-based algorithm for solving max-min fair resource allocation problems. A crucial part of this algorithm is the energy difference measure which determines how much worse a given solution is compared to another solution, i.e. the energy difference of the solutions. We evaluate the impact of the three energy difference measures (5), (7) and (8) on the performance of MAXMINFAIR_SA.

Our test setup was the following: For each energy difference function we independently performed 50 runs with MAXMINFAIR_SA. The temperature levels were determined experimentally, we set $\vartheta_{max} = 5$ and $\vartheta_{min} = 0.01$; the cooling rate α was set according to (9). In order to establish consistent experimental conditions for fair optimization, we used a timeout, which was determined according to the publicly available ITC2007 benchmark executable. On our machines (i7 CPUs running at 3.4GHz, 8GB RAM), the timeout was set to 192 seconds. The MAXMINFAIR_SA algorithm was executed on a single core. We generated feasible initial timetables for MAXMINFAIR_SA as a preprocess using standard sequential heuristics [9]. The soft constraint violations were not considered at this stage. Since the preprocess was performed only once per instance (not per run), it is not counted in the timeout. However, the time it took was negligible compared to the timeout (less than 1 second per instance).

Table 2 shows the impact of the parameter δ on the performance of MAXMINFAIR_SA with energy difference measure ΔE_{cw}. For each pair of values we performed the one-sided Wilcoxon Rank-Sum test with a signif cance level of 0.01. The data indicates that for best performance, δ should be small, but not too small. For $\delta = 1$, MAXMINFAIR_SA beats the other shown conf gurations on instance `comp02` but performs worse than the other conf gurations on instances `comp10` and `comp20`. For $\delta = 10^{-6}$ the overall performance is better than for $\delta = 1$, but worse than for the other conf gurations. With $\delta = 10^{-2}$ and $\delta = 10^{-3}$,

Table 3: The performance of MAXMINFAIR_SA with energy difference measures ΔE_{lex}, ΔE_{ps} and ΔE_{cw}.

ΔE	ΔE_{lex}	ΔE_{ps}	ΔE_{cw}
ΔE_{lex}	–	–	–
ΔE_{ps}	all except 01,06,08,11,17	–	18
ΔE_{cw}	all except 11	06,07,08,17,21	–

MAXMINFAIR_SA shows the best relative performance. Thus, for our further evaluation we set $\delta = 10^{-3}$.

Table 3 shows the relative performance of Algorithm MAXMINFAIR_SA for the proposed energy difference measures (5), (7) and (8). The table shows for any choice of two energy difference measures i and j, for which instances MAXMINFAIR_SA with measure i performs signif cantly better than MAXMINFAIR_SA using measure j. Again, we used the Wilcoxon Rank-Sum test with a signif cance level of 1 percent. The data shows that ΔE_{cw} is the best choice among the three alternatives, since it is a better choice than ΔE_{lex} on all instances except `comp11` and a better choice than ΔE_{ps} on f ve out of 21 instances. However, although ΔE_{cw} shows signif cantly better performance than the other energy difference measures, it did not necessarily produce the best timetables on all instances. For the instances `comp03`, `comp15`, `comp05` and `comp12` for example, the best solution found with $\Delta E = \Delta E_{ps}$ was better than with $\Delta E = \Delta E_{cw}$.

The data in Table 4 shows a comparison of the sorted allocation vectors of the known best solutions from [13] with the best solutions found by the 50 runs of MAXMINFAIR_SA with $\Delta E = \Delta E_{cw}$. First of all, for instances `comp01` and `comp11`, the allocation vectors of the best existing solutions and the best solution found by MAXMINFAIR_SA are identical. This means that MAXMINFAIR_SA f nds reasonably good solutions despite the certainly more complex f tness landscape due to max-min fair optimization. We can also observe that the maximum penalty any curriculum receives is signif cantly less for most instances and the penalty is more evenly distributed across the curricula. This means that although max-min fair timetables may have a higher total penalty, they might be more attractive from the students' perspective, since in the f rst place each student notices an unfortunate arrangement of his/her timetable, which is tied to the curriculum. Furthermore, we can observe that if the total penalty of a known best solution is rather low, then it is also good with respect to max-min fairness. For several instances in this category, (`comp01`, `comp04`, `comp07`, `comp10` and `comp20`), the solution found by MAXMINFAIR_SA is not as good as the known best solution with respect to max-min fairness. We can conclude that if there is not much penalty to distribute, it is not necessary to bother about a fair distribution of penalty.

5.2 The Tradeoff Between Fairness and Eff ciency

We proposed the JFI-CB-CTT problem formulation in Section 3, which allows us to investigate the tradeoff between fairness and eff ciency which arises in course timetabling. We can observe in column 4 of Table 1 that for all of the best solutions from [13] the fairness index (1) is greater than 0.8, i. e., the known best solutions are also fair for more than 80 percent of the curricula. In order to solve the corresponding JFI-CB-CTT instances, we use the multi-objective optimization algorithm AMOSA proposed in [3] that is based on simulated annealing like Algorithm MAXMINFAIR_SA. Since we do not expect from a general multi-objective optimization algorithm to produce solutions as good as the best CB-CTT

Table 4: Comparison of the sorted allocation vectors of the known best solutions from [13] with the allocation vectors found by MAXMINFAIR_SA with respect to max-min fairness.

Instance	Known best solution	MAXMINFAIR_SA ($\Delta E = \Delta E_{cw}$)
comp01	$5^2, 0^{12}$	$5^2, 0^{12}$
comp02	$4, 2^{10}, 0^{59}$	$4^2, 2^{31}, 1^7, 0^{30}$
comp03	$13, 10^3, 9, 7^2, 6^4, 5^{13}, 4, 2^6, 0^{37}$	$6^4, 4^{11}, 2^{22}, 1^3, 0^{28}$
comp04	$7, 6^3, 5^4, 4^2, 2, 0^{46}$	$6^4, 4^2, 2^4, 1, 0^{46}$
comp05	$41^2, 36^7, 35^5, 32^5, 31^6, 30^9, 28, \ldots, 2, 0^3$	$19^2, 18^3, 17^3, 16^5, 15^2, 14^{15}, 13^5, \ldots, 4^8, 3^3, 2$
comp06	$12, 7^2, 5^4, 2^3, 0^{60}$	$12, 4^2, 2^{30}, 1^{13}, 0^{24}$
comp07	$6, 0^{76}$	$6, 2^{23}, 1^{24}, 0^{29}$
comp08	$7, 6^3, 5^4, 4^2, 2^2, 0^{49}$	$6^4, 4^2, 2^7, 1^5, 0^{43}$
comp09	$10^5, 9, 7^{10}, 6^6, 5^{10}, 4, 2, 0^{41}$	$6^9, 4^{14}, 2^{17}, 0^{35}$
comp10	$2^2, 0^{65}$	$2^{19}, 1^6, 0^{42}$
comp11	0^{13}	0^{13}
comp12	$45, 30^{14}, 28, 27^2, 26^5, 25^{19}, 22^4, \ldots, 2^2, 0^3$	$10^3, 9^6, 8^{31}, 7^7, 6^{43}, 5^2, 4^{36}, 3^2, 2^{16}, 1, 0^3$
comp13	$8, 7, 6^5, 5^7, 4^2, 2^3, 0^{47}$	$6^6, 4^4, 2^{13}, 1^6, 0^{37}$
comp14	$8^4, 7, 5^2, 2^6, 0^{47}$	$8^4, 4^2, 3, 2^{18}, 0^{35}$
comp15	$10^3, 9^3, 7, 6^4, 5^{13}, 4, 2^7, 0^{36}$	$6^4, 4^{11}, 2^{23}, 1^2, 0^{28}$
comp16	$7^2, 5^7, 4, 0^{61}$	$4^5, 2^{16}, 1^4, 0^{46}$
comp17	$10^2, 6^3, 5^9, 2^4, 0^{52}$	$10^2, 6^2, 4^7, 3, 2^{25}, 1^7, 0^{26}$
comp18	$17, 15, 14, 13, 11, 10, 9^2, 5^{19}, 2^2, 0^{23}$	$4^{20}, 2^{11}, 1^5, 0^{16}$
comp19	$13, 7, 6^4, 5^2, 4, 2^7, 0^{50}$	$6^4, 4^6, 2^{15}, 1^{14}, 0^{27}$
comp20	$2^2, 0^{76}$	$4^5, 3^3, 2^{31}, 1^7, 0^{32}$
comp21	$12, 11, 10^4, 9, 7^4, 6^4, 5^{12}, 4, 2^3, 1^2, 0^{45}$	$10, 6^4, 5, 4^{15}, 3, 2^{36}, 1^3, 0^{17}$

solvers, we will consider the following scenario to explore the tradeoffs between fairness and eff ciency: starting from the known best solution we examine how much increase in total penalty we have to tolerate in order to increase the fairness further. We will take as examples the six instances with the highest total amount of penalty, `comp03`, `comp05`, `comp09`, `comp12`, `comp15` and `comp21`.

The temperature levels for the AMOSA algorithm were set to $\vartheta_{max} = 20$ and $\vartheta_{min} = 0.01$; α was set according to (9) with a timeout determined by the off cial ITC2007 benchmark. The plots in Figure 1 show the (Pareto-) non-dominated solutions found by AMOSA. The arrows point to the starting point, i.e. the best available solutions to the corresponding instances. For instances `comp05` and `comp21` solutions with a lower total cost than the the previously known best solutions were discovered by this approach. The plots show that the price for increasing the fairness is generally not very high – up to a certain level, which depends on the instance. In fact, for `comp09` and `comp21`, the fairness index can be increased by 3.5 percent and 1.4 percent, respectively, without increasing the total penalty at all.

In Figure 1, the straight lines that go through the initial solutions show a possible tradeoff between fairness and eff ciency: the slopes were determined such that a 1 percent increase in fairness yields a 1 percent increase in penalty. For the instances shown in Figure 1, the solutions remain close to the tradeoff lines up to a fairness of 94 to 97 percent, while a further increase in fairness demands a signif cant increase in total cost. For the instances `comp05`, `comp09` and `comp15`, there are several solutions below the tradeoff lines. Picking any of the solutions below these lines would result in an increased fairness without an equally large increase in the amount of penalty. This means picking a fairer solution might well be an attractive option in a real-world academic timetabling context. For `comp05` for example, the fairness of the formerly best known solution with a total penalty of 291 can be increased by 5.4 percent at 302 total penalty, which is a 3.8 percent increase.

In summary, improving the fairness of an eff cient timetable as a post-processing step seems like a viable approach for practical decision making. Using a very eff cient solution as a starting point means that we can benef t from the existing very good approaches to creating timetables with minimal total cost and provide improved fairness depending on the actual, instance-dependent tradeoff.

6 Conclusion

In this paper we introduced two new problem formulations for academic course timetabling based on the CB-CTT problem model from track three of the ITC2007, MMF-CB-CTT and JFI-CB-CTT. Both problem formulations are aimed at creating fair course timetables in the setting of a university but include different notions of fairness. Fairness in our setting means that the penalty assigned to a timetable is distributed in a fair way among the different curricula. The MMF-CB-CTT formulation aims at creating max-min fair course timetables while JFI-CB-CTT is a bi-objective problem formulation based on Jain's fairness index. The motivation for the JFI-CB-CTT formulation is to explore the tradeoff between a fair penalty distribution and a low total penalty.

Furthermore, we proposed an optimization algorithm based on simulated annealing for solving MMF-CB-CTT problems. A critical part of the algorithm is concerned with measuring the energy difference between two timetables, i.e., how much worse a timetable is compared to another timetable with respect to max-min fairness. We evaluated the performance of the proposed algorithm for three different energy difference measures on the 21 CB-CTT benchmark instances. Our results show clearly that the algorithm performs best with ΔE_{cw} as energy difference measure.

Additionally, we investigated the fairness of the known best solutions of the 21 CB-CTT instances with respect to max-min fairness and Jain's fairness index. These solutions were not created with fairness in mind, but our results show that all of the solutions have a fairness index greater than 0.8. This means they can be considered quite fair. Nevertheless, our results show that some improvements are possible with respect to both max-min fairness and Jain's fairness index. The timetables produced by our proposed MAXMINFAIR_SA algorithms are better than the known best ones with respect to max-min fairness for 15 out of 21 instances. Our investigation of the tradeoff between fairness and the total amount of penalty using the JFI-CB-CTT problem formulation shows that the fairness of the known best timetables can be increased further with only a small increase of the total penalty.

Acknowledgements

We would like to thank the anonymous referees for their valuable remarks on this paper.

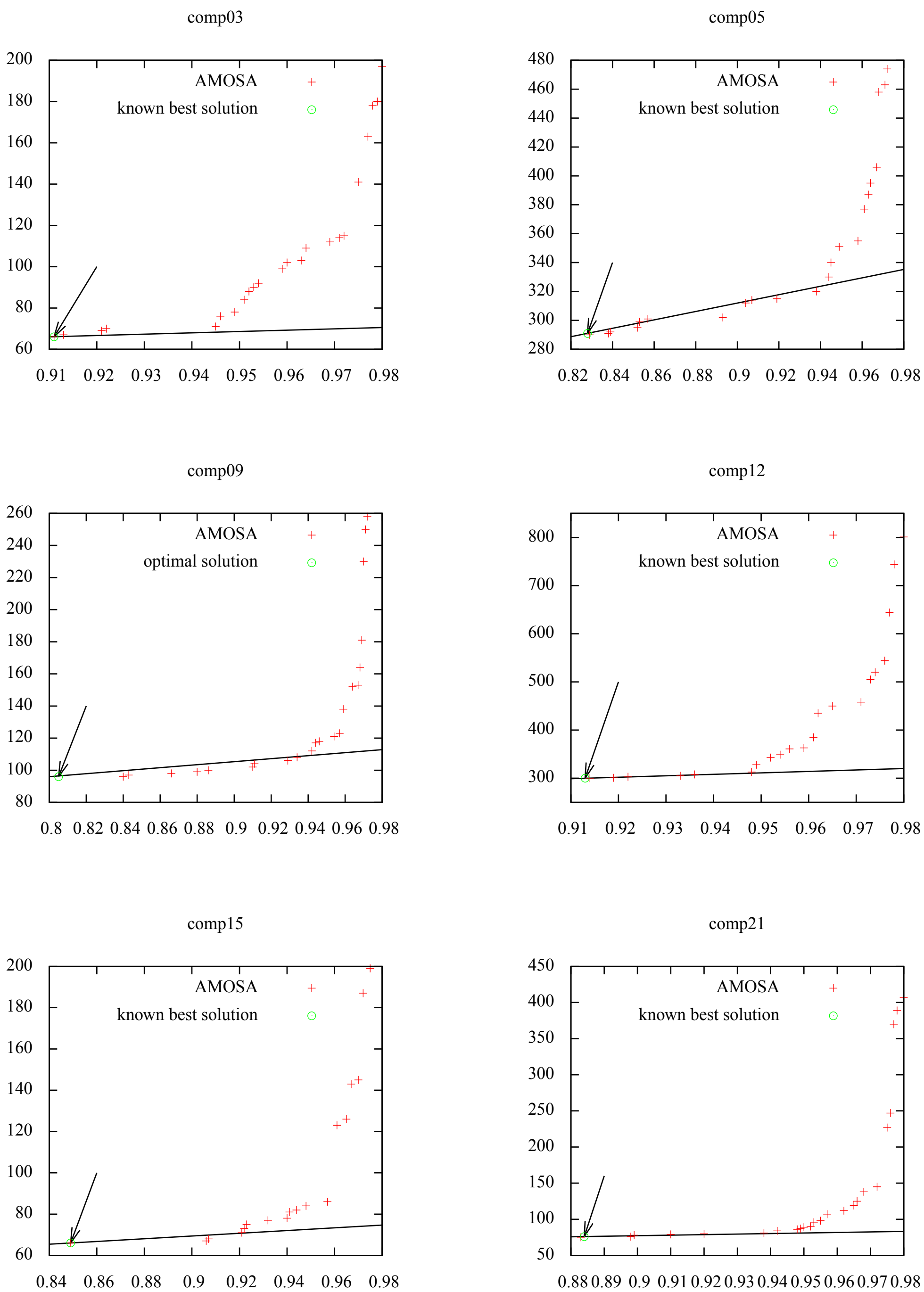

Fig. 1: Non-dominated solutions found by the AMOSA algorithm for the JFI-CB-CTT versions of instances `comp03`, `comp05`, `comp09`, `comp12`, `comp15` and `comp21`. All graphs show the fairness index on the horizontal axis and the amount of penalty on the vertical axis.

References

1. Salwani Abdullah, Edmund K. Burke, and Barry McCollum. A hybrid evolutionary approach to the university course timetabling problem. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1764–1768, 2007.
2. Roberto Asín Acha and Robert Nieuwenhuis. Curriculum-based course timetabling with SAT and MaxSAT. In *Proc. 8th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT)*, pages 42–56, 2010.
3. Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12:269–283, 2008.
4. Yair Bartal, Martin Farach-Colton, Shibu Yooseph, and Lisa Zhang. Fast, fair and frugal bandwidth allocation in ATM networks. *Algorithmica*, 33(3):272–286, 2002.
5. Dimitri P. Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, 2nd edition, 1992.
6. Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. The price of fairness. *Operations Research*, 59(1):17–31, February 2011.
7. Edmund K. Burke, Adam J. Eckersley, Barry McCollum, Sanja Petrovic, and Rong Qu. Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206(1):46–53, 2010.
8. Edmund K. Burke, Jakub Mareček, Andrew J. Parkes, and Hana Rudová. A branch-and-cut procedure for the udine course timetabling problem. *Annals of Operations Research*, 194(1):71–87, 2011.
9. Edmund K. Burke, Barry McCollum, Amnon Meisels, Sanja Petrovic, and Rong Qu. A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176(1):177–192, 2007.
10. Dah-Ming Chiu and Raj Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.
11. Luca Di Gaspero, Barry McCollum, and Andrea Schaerf. The second international timetabling competition (ITC-2007): Curriculum-based course timetabling (Track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT/v1.0/1, School of Electronics, Electrical Engineering and Computer Science, Queens University, Belfast (UK), August 2007.
12. Luca Di Gaspero and Andrea Schaerf. Neighborhood portfolio approach for local search applied to timetabling problems. *Journal of Mathematical Modelling ans Algorithms*, 5:65–89, 2006.
13. Luca Di Gaspero and Andrea Schaerf. Curriculum-based course timetabling site. `http://satt.diegm.uniud.it/ctt/`, January 2012.
14. Jack Edmonds and D.R. Fulkerson. Bottleneck extrema. *Journal of Combinatorial Theory*, 8(3):299 – 306, 1970.
15. Allan M. Feldman and Roberto Serrano. *Welfare economics and social choice theory*. Springer, New York, NY, 2nd edition, 2006.
16. Corrado Gini. Measurement of inequality of incomes. *The Economic Journal*, 31(121):124–126, January 1921.
17. Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC-TR-301, Digital Equipment Corporation, September 1984.
18. Frank Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.
19. Scott Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
20. Jon Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. *Journal of Computer and System Sciences*, 63(1):2–20, 2001.
21. Philipp Kostuch. The university course timetabling problem with a three-phase approach. In *Proc. 5th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT)*, pages 109–125. Springer, 2004.
22. Christos Koulamas, Solomon Antony, and R. Jaen. A survey of simulated annealing applications to operations research problems. *Omega*, 22(1):41–56, 1994.
23. P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, 1987.
24. Gerald Lach and Marco E. Lübbecke. Curriculum based course timetabling: new solutions to Udine benchmark instances. *Annals of Operations Research*, pages 1–18, 2010.
25. Tian Lan, David Kao, Mung Chiang, and Ashutosh Sabharwal. An axiomatic theory of fairness in network resource allocation. In *INFOCOM*, pages 1343–1351. IEEE, 2010.

26. Zhipeng Lü and Jin-Kao Hao. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235–244, 2010.
27. Barry McCollum, Andrea Schaerf, Ben Paechter, Paul McMullan, Rhyd Lewis, Andrew J. Parkes, Luca Di Gaspero, Rong Qu, and Edmund K. Burke. Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing*, 22:120–130, 2010.
28. Liam T. G. Merlot, Natashia Boland, Barry D. Hughes, and Peter J. Stuckey. A hybrid algorithm for the examination timetabling problem. In *Proc. 4th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT)*, pages 207–231. Springer, 2002.
29. Tomáš Müller. ITC2007 solver description: a hybrid approach. *Annals of Operations Research*, 172(1):429–446, 2009.
30. Włodzimierz Ogryczak. Bicriteria models for fair and eff cient resource allocation. In *Proceedings of the 2nd International Conference on Social Informatics (SocInfo)*, pages 140–159. Springer, 2010.
31. Włodzimierz Ogryczak and Adam Wierzbicki. On multi-criteria approaches to bandwidth allocation. *Control and Cybernetics*, 33:427–448, 2004.
32. Abraham P. Punnen and Ruonan Zhang. Quadratic bottleneck problems. *Naval Research Logistics (NRL)*, 58(2):153–164, 2011.
33. John Rawls. *A Theory of Justice*. Belknap Press of Harvard University Press, revised edition, 1999.
34. Thomas Repantis, Yannis Drougas, and Vana Kalogeraki. Adaptive resource management in Peer-to-Peer middleware. In *Proc. 19th IEEE Int. Parallel and Distributed Processing Symposium (IPDPS)*, 2005.
35. Ronaldo M. Salles and Javier A. Barria. Lexicographic maximin optimisation for fair bandwidth allocation in computer networks. *European Journal of Operational Research*, 185(2):778 – 794, 2008.
36. Amartya Sen and James E. Foster. *On economic inequality*. Clarendon Press ; Oxford University Press, Oxford : New York :, enl. ed., with a substantial annexe ”on economic inequality after a quarter century” / james foster and amartya kumar sen. edition, 1997.
37. M. J. Soomer and G. M. Koole. Fairness in the aircraft landing problem. In *Proceedings of the Anna Valicek Competition 2008*, 2008.
38. Jonathan Thompson and Kathryn A. Dowsland. General cooling schedules for a simulated annealing based timetabling system. In *Proc. 1st Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT)*, pages 345–363, 1996.
39. Jonathan M. Thompson and Kathryn A. Dowsland. A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25(7-8):637–648, 1998.
40. Mauritsius Tuga, Regina Berretta, and Alexandre Mendes. A hybrid simulated annealing with Kempe chain neighborhood for the university timetabling problem. In *Proceedings of the 6th ACIS Intinternational Conference on Computer and Information Science (ACIS-ICIS)*, pages 400–405, 2007.
41. Liang Zhang, Wen Luo, Shigang Chen, and Ying Jian. End-to-end maxmin fairness in multihop wireless networks: Theory and protocol. *Journal of Parallel and Distributed Computing*, 72(3):462–474, 2012.