Vol. 33, No. 3 (2023) | DOI: 10.37190/ord230301

OPEN ACCESS

Operations Research and Decisions

www.ord.pwr.edu.pl

OPERATIONS RESEARCH AND DECISIONS QUARTERLY

The SEKO assignment. Efficient and fair assignment of students to multiple seminars

Tobias Hoßfeld[1*]

[1] *Chair of Communication Networks, Faculty of Mathematics and Computer Science, University of Würzburg, Germany*

[*] *Email address: tobias.hossfeld@uni-wuerzburg.de*

Abstract

Seminars are offered to students for education in various disciplines. The seminars may be limited in terms of the maximum number of participants, e.g., to have lively interactions. Due to capacity limitations, those seminars are often offered several times to serve the students' demands. Still, some seminars are more popular than others and it may not be possible to grant access to all interested students due to capacity limitations. In this paper, a simple, but efficient random selection using key objectives (SEKO) assignment strategy is proposed which achieves the following goals: (i) efficiency by utilizing all available seminar places, (ii) satisfying all students by trying to assign at least one seminar to each student, and (iii) fairness by considering the number of assigned seminars per student. We formulate various theoretical optimization models using integer linear programming (ILP) and compare their solutions to the SEKO assignment based on a real-world data set. The real-world data set is also used as the basis for generating large data sets to investigate the scalability in terms of demand and number of seminars. Furthermore, the first-in first-out (FIFO) assignment, as a typical implementation of fair assignments in practice, is compared to SEKO in terms of utilization and fairness. The results show that the FIFO assignment suffers in real-world situations regarding fairness, while the SEKO assignment is close to the optimum and scales regarding computational time in contrast to the ILP.

Keywords: *assignment problem, limited capacity, utilization, at least on seminar (ALOS) ratio, fairness, integer linear programming (ILP), first-in first-out (FIFO) assignment, random selection using key objectives (SEKO)*

1. Introduction

Seminars, workshops, group work, or other interactive courses are part of university education processes, and also part of private or professional further education. Students or interested persons, in general, want to participate in such activities, which are, however, often limited in terms of the maximum number of participants for different reasons. For example, workshops and group work with lively interactions require a proper size, as noted by Al Mulhim and Eldokhny [1] in the context of project-based learning environments or Chen and Liu [4] for asynchronous online discussion in higher education. Other reasons

Received 1 June 2022, accepted 1 June 2023, published online 21 September 2023
ISSN 2391-6060 (Online)/

The costs of publishing this issue have been co-financed by the program *Development of Academic Journals* of the Polish Ministry of Education and Science under agreement RCN/SP/0241/2021/1

for a limited number of available places may be organizational reasons such as room capacities or available technical equipment. Due to the place or capacity limitations, such activities and courses are often offered multiple times (in parallel or in different time slots) to serve the students' demands. Still, some seminars are more popular than others and it may not be possible to grant access to all those interested due to capacity limitations.

In this article, we consider a general assignment problem of students to multiple seminars with limited capacity according to their preferences, taking into account possibly several offers for seminars of the same type. For the sake of simplicity, we will just speak of students and seminars, but our solution approaches can be applied in a more general context. In the planning phase of organizing the seminars in a semester, the historical popularity of seminars in terms of the number of requests is considered from the previous semester. Accordingly, several seminars of the same type may then be offered for popular seminars. Students may request several seminars of the same type, but of course, only a single seminar of the same type may be assigned. The requests by all students are captured and considered in an assignment strategy.

A common and simple strategy, which is often used in practice, is a first-in first-out (FIFO) strategy, also referred to as first-come first-serve (FCFS) strategy. In a different context of queueing systems, such a FIFO strategy is considered fair, since job requests arrive in a queue and all jobs are served in the order they arrive. This is referred to as order fairness. Avi-Itzhak et al. [2] review different types of fairness and discuss the FIFO strategy which is the fairest under the concept of order fairness. However, in the context of queues, a FIFO strategy is unfair if the job requests have different demands and require different capacities of the queue [14]. To this end, the concept of proportional fairness is also considered in queues: it is fair for large jobs to have large response times and for small jobs to have small response times. Proportional fairness is, however, not directly applicable in our scenario. Instead, for our use case of assigning students to seminars, fairness needs to consider the number of assigned seminars per student in relation to the number of requested seminars. One of the contributions of this article is a proper definition of fairness for our use case.

The FIFO strategy is a typical implementation in our case and is considered fair due to order fairness. There is a registration start time and students then may register for seminars. The places in the workshops are then assigned in a FIFO manner to students registered for that seminar until the capacity of the seminar is reached. It is assumed that all students have the same opportunities to register for seminars. This is however not true in practice. Some students may not be able to register soon after the registration starts due to technical reasons, conflicting dates, or personal reasons. Hence, this may cause some unfairness. In addition, students may be interested in several seminars and register for several workshops at the same time; this is how registration interfaces are typically implemented: students select the seminars in an online form and then register for all of the selected seminars at once by submitting the form. Then, FIFO will cause additional unfairness: students registering early for many seminars will get all of them assigned.

In this paper, a simple, but efficient random selection using key objectives (SEKO) assignment strategy is proposed which achieves the following goals: (i) efficiency by utilizing all available seminar places, (ii) satisfying all students by trying to assign at least one seminar (ALOS) to each student, and (iii) fairness by considering the number of assigned seminars per student. We formulate various theoretical optimization models and compare their solutions to the SEKO assignment based on a real-world data set. The FIFO assignment, which is often used in practice, is compared to SEKO in terms of utilization and fairness. The results show that the FIFO assignment suffers in real-world situations regarding fairness, while the SEKO assignment is close to the optimum. Furthermore, we investigate the scalability of integer linear programming (ILP) concerning computational time and compare the performance of SEKO to the results of the ILP. Thereby, the real-world data set is used as the basis for generating large data sets to investigate the scalability.

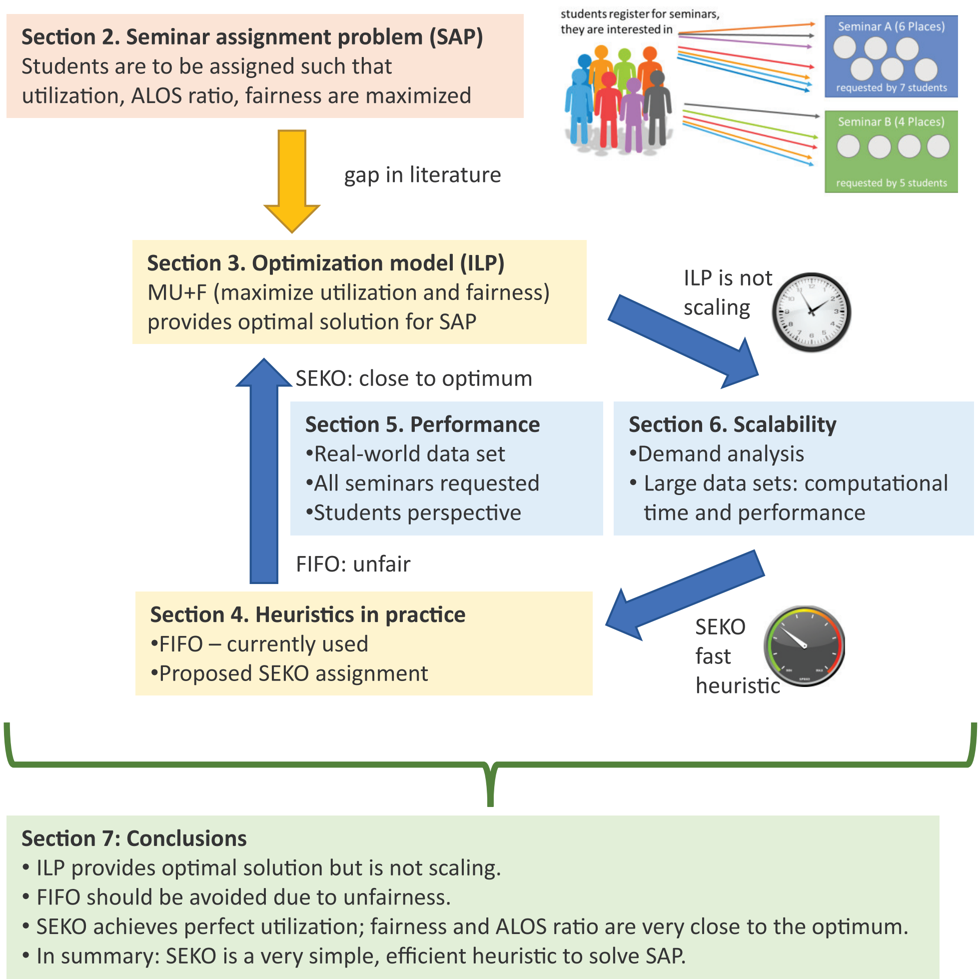

Figure 1. Structure of this article and summary of the key results

Our key contributions to this article are as follows:

- definition of quantitative measures for key objectives: utilization, ALOS ratio, fairness;
- formulation of optimization problems by means of ILP;
- formulation of SEKO assignment: random selection using key objectives;
- simulative performance evaluation of SEKO and comparison to FIFO and optimum based on a real-world data set from the year 2021;
- model to generate large data sets mimicking the characteristics of the real-world data set;
- analysis of the scalability of the ILP and SEKO concerning computational time.

The remainder of this article is structured as follows. Section 2 properly formulates the problem to be solved and reviews related work in the context of assignment strategies, related optimization problems, and quantitative measures for key objectives. Section 3 introduces three different optimization problems and formulates the corresponding ILPs for maximizing utilization, the ALOS ratio, and fairness. Section 4 explains the assignment strategies in practice and provides pseudo-code for the strategies. In particular, the SEKO assignment is described in Section 4.2, along with a motivation of its ideas and key objectives. Section 5 provides numerical data utilizing a real word data set of students' registrations for seminars and a systematic parameter sensitivity study is conducted. Section 6 evaluates the SEKO and the ILP in terms of scalability. Thereby, the computational time, but also performance is assessed. Section 7 concludes this work. A graphical visualization of the structure and the contents of this article is provided in Figure 1.

2. Related work and problem formulation

First, we review related work on optimization problems close to ours. However, there is yet no optimization problem that exactly captures our use case. Then, quantitative measures are considered and proposed on utilization, fairness, and the ALOS ratio. Finally, the formulation of the seminar assignment problem (SAP) which we are addressing in this article is given.

2.1. Assignment problem

The generalized assignment problem (GAP) is a well-known optimization problem. There are in general n items and m knapsacks. A solution to the GAP finds the optimum assignment of each item to exactly one of the m knapsacks. However, the capacity of any knapsack is limited. The generalized problem considers different costs or weights of the items. In the special case of the assignment problem, all items have the same costs. The GAP seems to be similar to our problem. There are n students who are allocated to m seminars and the capacity of the seminars is limited. Each student has the same "cost" and occupies one seminar place.

Cattrysse and Van Wassenhove [3] survey solution approaches for GAP and conclude that there is a shortage of effective heuristics. Öncan [10] surveys literature with a different focus on GAP and its real-life applications in various fields, like scheduling, telecommunication, facility location, transportation, production planning, etc. In the paper, search algorithms, exact solution procedures, and simple heuristic algorithms are revisited and evaluated. However, the GAP does not consider that there are preferences of the students for which seminars to attend. Furthermore, a student may want to participate in several seminars. Therefore, we cannot use existing GAP solutions for our problem.

The reviewer assignment problem (RAP) takes into account such preferences. In the RAP, reviewers are assigned to do a scientific peer review of several manuscripts. The matching degree between manuscripts and reviewers is thereby explicitly taken into account. Wanget al. [13] survey research on RAP and its variations. They additionally discuss practical challenges for developing a RAP system. Garg ey al.[6] consider the RAP for conferences where a bidding phase is included such that reviewers express their preferences for which papers to review and which papers cannot be reviewed due to missing expertise or personal conflict of interests. Garg et al. [6] also focus on fairness and load balance of the assignments. Fairness considers hereby the quality of the assigned reviewers across all papers as well as the number of assigned reviews according to the preferences of the referees. Related to our seminar assignment problem, the students are the referees and the bidding process captures the preferences of students for which seminars to attend. The seminars are then the manuscripts to be reviewed which may have a maximum number of reviews. However, the RAP is different. It takes into account coverage, i.e., every paper should be reviewed a sufficient number of times. Hence, a seminar should be allocated a sufficient number of times. This depends however on the preferences of the students. Instead of coverage, the utilization of available seminar places is a key objective in practice. Furthermore, there are seminars that are offered at different times. This does not happen in the RAP – this would result in a double submission at the same conference. In summary: the RAP problem does not match our seminar assignment problem.

Having a closer look at assignment problems regarding courses, projects, or seminars at universities, there are also various approaches existing. Roeder and Saltzman [12] propose schedule-based group assignments where students are assigned to groups based on their availability. The goal is to assign every student to exactly one group. Here, the key focus is to consider the availability of students such that they can meet in a group.

Rezaeinia et al. [11] discuss an assignment problem arising in the allocation of students to business projects in a master program. The students provide their preferences on the projects they want to work on, placing focus on efficiency and fairness in the assignment problem. For integrating fairness in the optimization problem, Jain's index is used to quantify the utility of the assignment. Utility associated with the assignment of a student to a specific project considers the preferences of the students in which project to work. The students provide a ranking of the projects which reflects their utility. Efficiency is then the sum of the utilities of the projects assigned to students. Hence, the goal is to optimize efficiency (which we refer to as utilization) and fairness. There are again some key differences with the SAP. For example, there is an upper bound and a lower bound on the number of students that are needed for a project. Thereby, the capabilities of students are required (referred to as attributes) such that a project can be successfully conducted. In our seminar assignment problem, we also have an upper bound and a lower bound for the number of attendees per seminar, but there is no need to consider specific capabilities. From that perspective, the optimization problem by Rezaeinia et al. [11] is more general and we may additionally extend our problem to take into account attributes or capabilities in the future. However, currently, this is not required in practice and we decided to skip this in our problem formulation. Furthermore, the lower bound is also not required for our real-world data set. For the sake of simplicity, we, therefore, skip this in our ILPs and assignment strategies, but this constraint can be added in a straightforward way. In contrast to the SAP, each student is assigned to a single project only in Rezaeinia et al. [11]. Similarly, Magnanti and Natarajan [9] also address the problem of assigning students to projects by considering efficiency and fairness as optimization criteria sequentially. However, they use a different kind of fairness. The lexicographic max–min fairness criterion consists of minimizing the number of students assigned to their least preferred project. This process is then repeated with the second-to-last preference. Again, students are assigned to single projects only. In contrast, students may attend several seminars in our scenario.

In summary, we conclude that none of the discussed problems and their variants matches our particular seminar assignment problem and takes into account our key objectives. We close this gap in the literature for the practically relevant seminar assignment problem.

2.2. Quantitative measures

The SAP is based on key objectives and quantitative measures for them. The definition of quantitative measures requires a proper notion of variables. The outcome of an assignment strategy is captured by the binary variable $y_{i,j}$ indicating whether student i is assigned to seminar j ($y_{i,j} = 1$); otherwise $y_{i,j} = 0$. There are n students and m seminars. Then, the assignment for all students and seminars is captured in the assignment matrix $\boldsymbol{y} = (y_{i,j})_{n\times m}$. A seminar j has s_j available places. We define the set of all students $\mathbb{I} = \{i : i = 1, 2, \ldots, n\}$ and the set of all seminars $\mathbb{J} = \{j : j = 1, 2, \ldots, m\}$, respectively.

There may be several seminars that provide the same seminar contents but are, e.g., offered at different times or in parallel. There are t different seminar types. For each seminar type k, the set of seminars belonging to type k is referred to as $\mathcal{S}_k$. Any student $i \in \mathbb{I}$ will be assigned at most one seminar for the same seminar type k. This will be ensured by any assignment strategy. We define the set of all seminar types $\mathbb{K} = \{k : k = 1, 2, \ldots, t\}$.

2.2.1. Utilization U

The utilization U or efficiency of the assignment quantifies the ratio of assigned places:

$$U(\boldsymbol{y}) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j}}{\sum_{j=1}^{m} s_j} = \frac{\sum_{i=1}^{n} A_i}{\sum_{j=1}^{m} s_j} \tag{1}$$

with A_i being the number of assigned seminars for student i:

$$A_i = \sum_{j=1}^{m} y_{i,j} \quad \forall i \in \mathbb{I} \tag{2}$$

If the assignment $\boldsymbol{y}$ is clear from the context, we simply write U instead of $U(\boldsymbol{y})$. It is clear that $0 \le U \le 1$.

Note that the maximum possible utilization may be less than 100%, if there are fewer requests r_j for a seminar j than available places s_j. In addition, a student may register for several seminars of the same seminar type, but the student will get assigned only one seminar of the same seminar type. The maximum possible utilization U^* is formulated as a linear program problem in Section 3.1, see equation (12).

2.2.2. ALOS ratio L

In practice, it may be relevant for the satisfaction of the students that at least one seminar will be assigned to each student. At the same time, fairness will be increased if the assignment strategy tries to assign at least one seminar to students. Therefore, the proposed ALOS ratio measure quantifies the ratio of students with at least one seminar

$$L(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} \operatorname{sign}\Big(A_i\Big) \tag{3}$$

The signum function $\operatorname{sign}\left(A_i\right)$ returns $\operatorname{sign}\left(A_i\right) = 1$, if $A_i > 0$; otherwise $\operatorname{sign}\left(A_i\right) = 0$. If the assignment $\boldsymbol{y}$ is clear, we simply write L instead of $L(\boldsymbol{y})$.

2.2.3. Fairness index F

Our notion of fairness considers the variance of the number of assigned seminars A_i across the students $i = 1, \ldots, n$. One prominent measure is Jain's fairness index provided by Jain et al. [8] based on the coefficient of variation of the number of assigned seminars.

The mean number $\mu_{\boldsymbol{y}}$ of assigned seminars is

$$\mu_{\boldsymbol{y}} = \frac{1}{n} \sum_{i=1}^{n} A_i \tag{4}$$

The standard deviation $\sigma_{\boldsymbol{y}}$ is defined as square root of the variance:

$$\sigma_{\boldsymbol{y}}^2 = \left(\frac{1}{n} \sum_{i=1}^{n} {A_i}^2 \right) - \mu_{\boldsymbol{y}}^2 \tag{5}$$

The coefficient of variation is then

$$c_{\boldsymbol{y}} = \frac{\sigma_{\boldsymbol{y}}}{\mu_{\boldsymbol{y}}} \tag{6}$$

and we finally arrive at Jain's index:

$$J(\boldsymbol{y}) = \frac{1}{1 + c_{\boldsymbol{y}}^2} \; . \tag{7}$$

Jain's index J ranges from 0 to 1. However, Jain's index is not considering that there is a maximum possible standard deviation. As an example, we consider $n = 100$ students and there are $t = 5$ seminar types. Suppose that half of the students do not get any seminar ($A_i = 0$ for $i = 1, \ldots, 50$); the other half of students get k seminars ($A_i = k$ for $i = 51, \ldots, 100$). Then, Jain's index is $J = 0.5$ independently of the value k. It does not matter if 50% of the students get 1, 2, 3, 4, or 5 seminars assigned.

Therefore, we rely on a metric that was originally defined to quantify QoE fairness by Hoßfeld et al. [7]. This fairness index $F(\boldsymbol{y})$ is a linear function of the standard deviation $\sigma_{\boldsymbol{y}}$ which takes into account the maximum number t of seminars a single student can get assigned.

$$F(\boldsymbol{y}) = 1 - \frac{2\sigma_{\boldsymbol{y}}}{t} \tag{8}$$

$F(\boldsymbol{y})$ returns values in the interval $[0; 1]$ with 0 being the most unfair and 1 being the most fair assignment.

Consider the example from above. If half of the students get no seminar and the other half gets the maximum number $t = 5$ assigned, then the most unfair situation is observed. The fairness index returns $F = 0$ in that case. The fairest situation is achieved when all students get the same number A_i assigned. Then, $\sigma_{\boldsymbol{y}} = 0$ and $F = 1$ (and also $J = 1$). If 50% of the students get 1, 2, 3, 4, and 5 seminars assigned, the fairness index F is 0.8, 0.6, 0.4, 0.2, and 0.0, respectively (remember Jain's index is $J = 0.5$ in those cases). Hence, the fairness index F is intuitive and properly quantifies fairness for the assignment problem.

2.3. Problem formulation: seminar assignment problem (SAP)

The following problem is addressed in this paper. There are n students requesting places for m seminars. There are t different seminar types. For each seminar type k, the set of seminars belonging to type k is referred to as $\mathcal{S}_k$. The student's requests for seminars are captured in a binary request variable $x_{i,j}$ for student $i \in \mathbb{I}$ and seminar $j \in \mathbb{J}$, where $x_{i,j} = 1$ means that the student i requests seminar j. Then, the requests for seminars by all students are captured in the request matrix $\boldsymbol{x} = (x_{i,j})_{n\times m}$. An assignment strategy $\boldsymbol{y}$ is to be defined with the following properties:

1. The students' requests $\boldsymbol{x}$ are considered:

$$y_{i,j} \leq x_{i,j} \quad \forall i \in \mathbb{I},\ \forall j \in \mathbb{J}\ . \tag{9}$$

2. Any student $i \in \mathbb{I}$ will be assigned at most one seminar for the same seminar type $k \in \mathbb{K}$:

$$\sum_{j\in\mathcal{S}_k} y_{i,j} \leq 1 \quad \forall i \in \mathbb{I},\ \forall k \in \mathbb{K}\ . \tag{10}$$

3. The utilization $U(\boldsymbol{y})$ of the available seminar places reaches the maximum utilization U^*:

$$U(\boldsymbol{y}) = U^* \tag{11}$$

4. The ALOS ratio $L(\boldsymbol{y})$ shall be maximized.
5. The fairness index $F(\boldsymbol{y})$ shall be maximized.
6. The concrete assignment of seminars to users shall be reproducible.

These goals are partly contrary. Perfect fairness is achieved if all users get no seminar assigned.

3. Optimization model: integer linear programming (ILP)

The assignment of the seminars with respect to the key objectives is found by means of integer linear programming (ILP).

Table 1. Notations and variables used for the optimization problem

Variable	Explanation
$y_{i,j}$	target assignment variable (binary); $y_{i,j} = 1$ means that the seminar j is assigned to student i
$x_{i,j}$	request input data (binary); $x_{i,j} = 1$ means that the student i requests seminar j
A_i	assigned seminars (integer) for student i; it is $A_i = \sum_{j=1}^{m} y_{i,j}$
$A_{\max}$	maximum number of seminars being assigned to a student [$A_{\max} = 99$]
n	number of different students [$n = 103$]
$\mathbb{I}$	set of all students $\mathbb{I} = \{i : i = 1, 2, \ldots, n\}$
m	number of different seminars [$m = 9$]
$\mathbb{J}$	set of all seminars $\mathbb{J} = \{j : j = 1, 2, \ldots, m\}$
s_j	number of places in seminar j for $j = 1, \ldots, m$ [$s_j = 12$ for all j]
t	number of different seminar types [$t = 5$]
$\mathbb{K}$	set of all seminar types $\mathbb{K} = \{k : k = 1, 2, \ldots, t\}$
$\mathcal{S}_\omega$	set of seminars for seminar type ω for $\omega = 1, \ldots, t$ [see Table 2]

[1] The parameters of the real data set are given in [brackets].

The first ILP maximizes the utilization, i.e., the number of assigned places to students, and is referred to as MU. The resulting maximum utilization U^{MU} is then used as a constraint in the other ILP problems. The ILP MU + 1 provides an assignment that maximizes the ratio of students who are assigned at least one seminar with the maximum utilization which can be achieved added as a constraint. The third ILP MU + F aims at maximizing fairness while considering the maximum utilization as a constraint. Table 1 introduces the variables which are used in the optimization problems. An implementation of the three different ILPs is available on Github[1].

3.1. Maximum utilization problem (MU)

The first ILP maximizes the number of assigned seminars, i.e., the available seminar places are utilized in the best possible way. The binary variable $y_{i,j}$ indicates whether student i is assigned to seminar j which means $y_{i,j} = 1$; otherwise $y_{i,j} = 0$. Then, the objective function is the maximization of $y_{i,j}$ over all n students and all m seminars. The resulting absolute utilization is U^{MU} and the (relative) maximum utilization is

$$U^* = \frac{U^{\mathrm{MU}}}{\sum_{j=1}^{m} s_j} \tag{12}$$

The constraints of the MU problem are the number of available seminar places (equation (13b)) and the students' registrations to seminars (equation (13c)). A seminar j has s_j places and therefore the sum of assignments to all students must be less than the number of places: $\sum_{i=1}^{n} y_{i,j} < s_j$. This constraint is added for any seminar $j \in \mathbb{J}$. The final constraint considers the registrations $x_{i,j}$ of student i to seminar j. A seminar is only assigned if a student has registered for it: $y_{i,j} \leq x_{i,j}$. However, there may be several seminars that provide the same seminar contents and belong to the same seminar type. There are t different seminar types. For each seminar type k, the set of seminars belonging to type k is referred to as $\mathcal{S}_k$. Any student $i \in \mathbb{I}$ will be assigned at most one seminar for the same seminar type $k \in \mathbb{K}$. Hence, $n \cdot t$ constraints are added, see equation (13d). Finally, the maximum number of seminars which may be assigned to a student may be limited. Hence, for each student $i \in \mathbb{I}$, the sum of assigned seminars must be less than $A_{\max}$, see equation (13e). This results in optimization problem (13).

$$
\begin{aligned}
U^{\mathrm{MU}} = \underset{y_{i,j}}{\text{maximize}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} && && \text{(13a)}\\
\text{subject to} \quad & \sum_{i=1}^{n} y_{i,j} \leq s_j && \forall j \in \mathbb{J}, && \text{(13b)}\\
& y_{i,j} \leq x_{i,j} && \forall i \in \mathbb{I},\ \forall j \in \mathbb{J}, && \text{(13c)}\\
& \sum_{j \in \mathcal{S}_k} y_{i,j} \leq 1 && \forall i \in \mathbb{I},\ \forall k \in \mathbb{K}, && \text{(13d)}\\
& \sum_{j=1}^{m} y_{i,j} \leq A_{\max} && \forall i \in \mathbb{I} && \text{(13e)}
\end{aligned}
$$

[1] https://github.com/hossfeld/sekoAssignment

3.2. Assigning at least one seminar problem (MU + 1)

The ratio of students who are assigned at least one seminar is now maximized. To this end, we consider the number A_i of seminars assigned to student i.

$$A_i = \sum_{j=1}^{m} y_{i,j} \ , \quad \forall i \in \mathbb{I} \tag{14}$$

The objective function is the maximization of $\text{sign}\Big(A_i\Big)$ over all n students. The signum function just indicates if a student has one or more seminars assigned and returns $\text{sign}\Big(A_i\Big) = 1$ in this case. The constraints are identical to the MU optimization problem, but additionally, we ensure that the MU + 1 problem achieves the same utilization U^{MU} as the MU problem (equation (15f)).

$$\begin{aligned}
\underset{y_{i,j}}{\text{maximize}} \quad & \sum_{i=1}^{n} \text{sign}\Big(A_i\Big) && && (15a)\\
\text{subject to} \quad & \sum_{i=1}^{n} y_{i,j} \le s_j && \forall j \in \mathbb{J}, && (15b)\\
& y_{i,j} \le x_{i,j} && \forall i \in \mathbb{I},\ \forall j \in \mathbb{J}, && (15c)\\
& \sum_{j \in \mathcal{S}_k} y_{i,j} \le 1 && \forall i \in \mathbb{I},\ \forall k \in \mathbb{K}, && (15d)\\
& \sum_{j=1}^{m} y_{i,j} \le A_{\max} && \forall i \in \mathbb{I}, && (15e)\\
& \sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} \ge U^{\text{MU}} && \text{with solution } U^{\text{MU}} \text{ from ILP (13)} && (15f)
\end{aligned}$$

Please note that the signum function is not supported by default by many ILP solvers. Therefore, we need to reformulate the MU + 1 optimization problem by linearizing the signum function. The ILP

$$\underset{y_{i,j}}{\text{maximize}} \quad \sum_{i=1}^{n} \text{sign}\Big(A_i\Big) \tag{16}$$

is identical to the following ILP with $\gamma > m$, e.g., $\gamma = 1000$ and $m = 9$.

$$\begin{aligned}
\underset{b_i}{\text{maximize}} \quad & \sum_{i=1}^{n} b_i && (17a)\\
\text{subject to} \quad & b_i \le A_i \le \gamma b_i \quad \forall j \in \mathbb{J} && (17b)
\end{aligned}$$

The constraint in equation (17b) ensures that at least one seminar is assigned if possible (otherwise $A_i = 0$ and $b_i = 0$), while the objective function in equation (17) maximizes the ratio of students with at least one seminar.

3.3. Maximizing fairness and utilization problem (MU + F)

Fairness considers here the absolute deviation of the number A_i of assigned seminars to student i from the mean number μ of assigned seminars. The total number of assigned seminars is reflected by the solution of the MU optimization problem, i.e., U^{MU}, again considered as a constraint in the fairness ILP (equation (19f)).

$$\mu = \frac{1}{n} \sum_{i=1}^{n} A_i = \frac{1}{n} U^{\mathrm{MU}} \tag{18}$$

The objective function of the maximizing fairness and utilization problem (MU + F) is therefore to minimize the sum of the absolute deviations from the mean across the students. The constraints are identical to the MU + 1 problem

$$\begin{aligned}
\underset{y_{i,j}}{\text{minimize}} \quad & \sum_{i=1}^{n} |A_i - \mu| && && \text{(19a)} \\
\text{subject to} \quad & \sum_{i=1}^{n} y_{i,j} \le s_j && \forall j \in \mathbb{J}, && \text{(19b)} \\
& y_{i,j} \le x_{i,j} && \forall i \in \mathbb{I},\ \forall j \in \mathbb{J}, && \text{(19c)} \\
& \sum_{j \in \mathcal{S}_k} y_{i,j} \le 1 && \forall i \in \mathbb{I},\ \forall k \in \mathbb{K}, && \text{(19d)} \\
& \sum_{j=1}^{m} y_{i,j} \le A_{\max} && \forall i \in \mathbb{I}, && \text{(19e)} \\
& \sum_{i=1}^{n} \sum_{j=1}^{m} y_{i,j} \ge U^{\mathrm{MU}} && \text{with solution } U^{\mathrm{MU}} \text{ from ILP (13)} && \text{(19f)}
\end{aligned}$$

Since the absolute value function is nonlinear, the ILP is reformulated to linearize the objective function properly. The ILP

$$\underset{y_{i,j}}{\text{minimize}} \quad \sum_{i=1}^{n} |A_i - \mu| \tag{20a}$$

is identical to the following ILP

$$\begin{aligned}
\underset{y_{i,j},\, T_i}{\text{minimize}} \quad & \sum_{i=1}^{n} T_i && && \text{(21a)} \\
\text{subject to} \quad & T_i \ge A_i - \mu && \forall i \in \mathbb{I}, && \text{(21b)} \\
& -T_i \le A_i - \mu && \forall i \in \mathbb{I} && \text{(21c)}
\end{aligned}$$

Hence, the final ILP, which maximizes fairness in terms of the sum of absolute deviations of the mean assigned seminars per student for a given utilization, considers the ILP from equation (21) plus the constraints from equation (19).

Please note that the solution of the optimization problem MU + F includes the maximization of utilization, i.e., the solution of the problem MU, as a constraint (19f). Furthermore, the maximization of fairness, i.e., the minimization of the sum of the absolute deviations from the mean number of assigned seminars across the students is the objective function (19a) of the problem MU + F. This implicitly takes into account the ALOS ratio as we will see in the performance evaluation in Sections 5 and 6. In summary, MU + F leads to the desired optimal solution for the SAP problem.

4. Assignment algorithms in practice

For the SAP problem, the MU + F yields the optimal solution. However, the solution of the ILP may not scale with the problem size in terms of the number of seminars or students due to computational time. Therefore, we propose the SEKO assignment algorithm in practice, which is a heuristic with low computational time, but which is close to the optimal solution by design. To be more precise, we propose the SEKO assignment algorithm in practice to achieve efficiency and fairness. In this section, we introduce and motivate the SEKO strategy. Later, the performance of the SEKO assignment strategy and its computational time are discussed and compared to the ILPs in Sections 5 and 6.

In practice, however, the FIFO assignment is widely used. In fact, the simplest assignment strategy is FIFO, such that registered students are assigned to seminars until the seminars' capacities are exceeded. However, FIFO fails to achieve a high ALOS ratio and fairness, which will be also shown in Sections 5 and 6. A description of the FIFO and SEKO algorithms is provided below. An implementation of both algorithms is available on Github[2].

4.1. FIFO assignment

The FIFO assignment works as follows. The seminars are sorted according to the number of requests in ascending order. The assignment algorithms start with the least requested seminar j_1 and finally assign the students to the most popular seminar j_m. The number of requests of seminar j is r_j. The request matrix $\boldsymbol{x} = (x_{i,j})_{n \times m}$ indicates if a student i has requested seminar j; then $x_{i,j} = 1$; otherwise $x_{i,j} = 0$.

$$r_j = \sum_{i=1}^{n} x_{i,j} \text{ for seminar } j \in \mathbb{J} \tag{22}$$

The order in which the students are assigned to seminars is

$$\mathcal{J} = (j_1, j_2, \ldots, j_m) \text{ with } r_{j_1} \leq r_{j_2} \leq \cdots \leq r_{j_m} \tag{23}$$

The assignment of students to seminars in ascending order ensures that maximum utilization is achieved. There is a maximum number $A_{\max}$ of seminars being assigned to students. Consider that a particular student i is assigned to a very popular seminar and thereby reaching $A_{\max}$. The student i is also interested in a less popular seminar j, for which fewer requests than seminar places are obtained: $r_j \leq s_j$. In that case, $r_j - 1$ students are assigned and the capacity of the seminar j is not utilized. However, assigning first students to unpopular seminars avoids that situation and results in reaching maximum utilization.

Note that the request matrix $\boldsymbol{x}$ as input for the assignment algorithm is sorted according to the time τ_i the requests were made by the students i. Hence, $\tau_1 < \tau_2 < \cdots < \tau_n$ and the first row represents the first student requesting for seminars at time τ_1. Then, the FIFO assignment strategy simply iterates over the seminars j in the order $\mathcal{J}$ and assigns the s_j seminar places to the first requesting students. Note that $\min(s_j, r_j)$ seminar places are assigned, since less than s_j places may be requested. Obviously, the FIFO algorithm assigns all possible seminar places to interested students and achieves the maximum

[2] https://github.com/hossfeld/sekoAssignment

utilization U^* according to the MU ILP in equation (13). If a student i is assigned to seminar j, then $y_{i,j} = 1$; otherwise $y_{i,j} = 0$.

In addition, the FIFO assignment also considers that several seminars of the same type may be offered. If a student i is assigned to seminar j which is of seminar type ω, then the requests for seminars of the same type for this student are removed in this iteration step: $x_{i,\kappa} = 0$ for $\kappa \in \mathcal{S}_\omega$. The set $\mathcal{S}_\omega$ contains all seminars of the type ω and the seminar $j \in \mathcal{S}_\omega$. Due to changing $x_{i,j}$ and r_j, the order of seminars has to be determined in each iteration step again. This is implemented in the FIFO assignment by simply identifying in each iteration step the seminar j with the minimum number of requests r_j, see lines 16–19 in Listing 2. Note that the FIFO assignment considers the maximum number $A_{\max}$ of seminars being assigned to a student. Hence, in each iteration step, the list of eligible students is determined for the corresponding seminar i. Eligible students have been assigned so far less than $A_{\max}$ seminars and are interested in the seminar ($x_{i,j} = 1$).

The assignment is captured in the assignment matrix $\boldsymbol{y} = (y_{i,j})_{n\times m}$. A description of the algorithm is Python-like pseudo code and is provided in Listing 2. As already mentioned, the maximum utilization is reached: $U(\boldsymbol{y}) = U^*$. However, fairness $F(\boldsymbol{y})$ and the ALOS ratio $L(\boldsymbol{y})$ are unclear for the FIFO assignment $\boldsymbol{y}$.

4.2. Proposed SEKO assignment

The proposed SEKO assignment follows a random selection approach to assign which students are assigned to a seminar. While the FIFO assignment considers the time when students are requesting seminars, the SEKO approach ignores the timestamps. The underlying idea is to improve fairness in student assignments to seminars. Not all students have the opportunity to register for seminars early after the registration starts. In that case, such students will not be able to get to seminar places.

SEKO stands for random lection using key objectives that are

O1 – to maximize utilization U,

O2 – to maximize the ratio of students with at least one seminar L,

O3 – to maximize fairness F.

The utilization objective O1 is reached by iterating over the seminars in ascending order of the number of requests, as in the FIFO assignment. Hence, in each iteration step, the seminar j with the minimum number of requests r_j across all remaining seminars, not yet considered, is selected (see lines 17–20 in Listing 1).

The fairness objective O3 is taken into account by adapting the probability p_i to select a student i based on the number of assigned seminars $A_i^{(k)}$ so far in the kth assignment step. The assignments in the kth assignment step are denoted as $y_{i,j}^{(k)}$ and $A_i^{(k)} = \sum\limits_{j=1}^{m} y_{i,j}^{(k)}$. Again, the set of eligible students is determined in an iteration step for the seminar j. Eligible students $\mathcal{E}$ have not been assigned so far $A_{\max}$ seminars and the students are interested in the seminar ($x_{i,j} = 1$ for $i \in \mathcal{E}$)

$$\forall i \in \mathcal{E} : A_i^{(k)} = \sum_{j=1}^{m} y_{i,j}^{(k)} < A_{\max} \text{ and } x_{i,j} = 1 \tag{24}$$

First, the maximum number $A^{(k)}$ of assigned seminars is determined from the eligible students for seminar j in this iteration step k

$$A^{(k)} = \max_{i \in \mathcal{E}}(A_i) \tag{25}$$

Then, we define the variable $\tilde{p}_i$ which is the difference between the maximum number $A^{(k)}$ of seminars being assigned in round k and the number $A_i^{(k)}$ of currently assigned seminars for that student i. Note that $A_i^{(k)} \in \{0, 1, \dots, A^{(k)}\}$. This variable $\tilde{p}_i$ is later used to obtain a probability p_i for randomly selecting students to be assigned to a seminar. To obtain a probability, $\tilde{p}_i$ will be normalized by the sum of all variables $\tilde{p}_i$ over all students $i = 1, \dots, n$, cf. equation (28). If students have already been assigned the maximum number of seminars in round k, i.e., $A_i^{(k)} = A^{(k)}$, we ensure a (small) chance to be randomly selected in this round by adding 1 in the following equation

$$\tilde{p}_i = A^{(k)} - A_i^{(k)} + 1 \quad \forall i \in \mathbb{I} \tag{26}$$

For example, the maximum number of assigned seminars for eligible students is $A^{(k)} = 3$ in step k. Then, all students with $A_i^{(k)} = 3$ have the same $\tilde{p}_i = 1$; for students with $A_i^{(k)} = 2$, the value (and thus the probability p_i) is doubled ($\tilde{p}_i = 2$); for students with $A_i^{(k)} = 1$, the value is tripled ($\tilde{p}_i = 3$). An illustrative example is provided below showing the effect of taking into account the number of already assigned seminars.

The ALOS objective O2 aims at maximizing the ratio of students who are assigned at least one seminar. The SEKO assignment, therefore, multiplies the values $\tilde{p}_i$ by a factor α if a student i has not been assigned any seminar so far ($A_i^{(k)} = 0$). As default value, we use $\alpha = 100$ in our evaluations

$$\tilde{p}_i = \alpha(A^{(k)} - A_i^{(k)} + 1) = \alpha(A^{(k)} + 1) \quad \text{if } A_i^{(k)} = 0 \tag{27}$$

Finally, the values $\tilde{p}_i$ are normalized and we obtain the probability p_i that a student is assigned to seminar j in step k

$$p_i = \frac{\tilde{p}_i}{\sum\limits_{l \in \mathcal{E}} \tilde{p}_l} \text{ such that } \sum_{l \in \mathcal{E}} p_l = 1 \tag{28}$$

Now, we randomly select s_j students out of the set of eligible students (without replacement) based on the probabilities p_i[3]. Since we can pass a seed to the random number generator, the results are reproducible – which may be important in practice to have reproducible pseudo-random assignments. Of course, if the number of eligible students is less than s_j, all eligible students are assigned. If a student i is assigned to seminar j which is of seminar type ω, then the requests for seminars of the same type for this student are removed in the iteration step. The assignment is captured in the assignment matrix $\boldsymbol{y} = (y_{i,j})_{n \times m}$. A description of the algorithm's Python-like pseudo code is provided in Listing 1.

[3] The NumPy random number generator `random.number.choice` was used, but any algorithm for weighted random sampling without replacement can be used, e.g., [5].

Illustrative example of the SEKO strategy

A simple scenario is considered to show how the SEKO strategy works and how the probabilities change depending on the number of already assigned seminars. There are $n = 25$ students and $m = 5$ different seminars. All students request a place in each of the five seminars. Each seminar has a capacity of 12 places. We consider now the probability $\pi_{j,k}$ that every student who has already been assigned j seminars will be assigned the seminar in round k. Note $\pi_{j,k}$ is the sum of probabilities p_i (equation (28)) for all students with j seminars in round k. Assume z students have been assigned j seminars in round k, then $\pi_{j,k} = zp_i$.

Round 1. No seminar has been assigned to any student. Hence, all students have the same probability of $\pi_{0,1} = 1/n = 4\%$ to be assigned to the first seminar. As a result, 12 students will be assigned to the first seminar, while the other 13 students still have zero seminars.

Round 2. Now, the ALOS objective gets relevant. The 13 students without any seminar get a significantly higher probability to be selected due to the factor $\alpha = 100$. It is $\pi_{0,2} = 7.66\%$, while for the other students with one seminar the probability is only $\pi_{1,2} = 0.04\%$. Thus, the probability is about 200 times higher for students without any seminar. It is very likely that 12 students without any seminar get one assigned in this round.

Round 3. There are 24 students with one seminar and $\pi_{1,3} = 0.45\%$. There is one student without any seminar and $\pi_{0,3} = 89.29\%$. Hence, it is likely that the student will get one seminar assigned. Then, there are 11 more places which will be assigned to the students with one seminar so far.

Round 4. Hence, there will be 11 students with two seminars and $\pi_{2,4} = 2.56\%$. The remaining 14 students have one seminar and $\pi_{1,4} = 5.13\%$. Thus, the probability is twice as high, i.e., $\pi_{1,4} = 2 \cdot \pi_{2,4}$ in this round. This ensures that the objective on fairness is taken into account. Let us assume that 8 students with one seminar get a place in this round. Then four students with two seminars will get a third seminar assigned.

Round 5. Now, the places for the last seminar are assigned. There are still 6 students with one seminar, 15 students have two seminars, 4 students have 3 seminars. The probabilities are $\pi_{1,5} = 5.77\% = 3\pi_{3,5}$, $\pi_{3,5} = 3.85\% = 2\pi_{3,5}$, and $\pi_{3,5} = 1.92\%$. The probabilities take into account the number of already assigned seminars in a proper way. For example, the outcome of the random draw in this round may lead to 5, 6, 13, 1 students with 1, 2, 3, 4, seminars, respectively.

As a result, all seminar places are assigned ($U = 1$) and every student has at least 1 seminar ($L = 1$). The fairness index is in this case only about 0.66. It cannot be avoided that all students do not get the same number of seminars assigned. There are $5 \times 12 = 60$ places, but 25 students requesting all seminars. If all places are assigned, the minimal standard deviation is obtained if 15 students get 2 seminars and 10 students get 3 seminars, respectively. The mean number of seminars per student is 2.4. The standard deviation of the optimal assignment is $\sigma = 0.49$. Hence, the fairness index is $F = 1 - 2\sigma/5 = 0.80$ for a maximum number of 5 seminars to be assigned to a single student.

5. Performance evaluation for real-world data set

The performance evaluation is based on a real-world data set from the year 2021 (scenario REAL). We additionally investigate a scenario where every student requests all seminars and modify the requests in the real-world data set (scenario ALL). The SEKO assignment is compared to the FIFO assignment as well as the results from the optimization problems MU, MU + 1, MU + F.

5.1. Real-world data set

Our real word data set contains the requests $\boldsymbol{x} = (x_{i,j})_{n\times m}$ from 2021 for $n = 103$ students and $m = 9$ seminars. There are $t = 5$ different seminar types. The seminars of type $\omega = 1, 3, 4, 5$ are offered two times, e.g., $\mathcal{S}_1 = \{1, 2\}$ or $\mathcal{S}_5 = \{8, 9\}$. Table 2 summarizes the parameters of the data set and the popularity of seminars, expressed as number r_j of requests per seminar j. Further the distribution n_k of the number of students who are requesting k seminars is provided.

Table 2. Real data set with requests of students for the different seminar types

Variable	Parameter	Value								
n	number of students	103								
m	number of seminars	9								
t	number of different seminar types	5								
Ω	total number of places	108								
R	total number of requests	171								
ρ	requests to places ratio	1.53								
U^*	maximum utilization	98.14%								
$\boldsymbol{x}$	request matrix $\boldsymbol{x} = (x_{i,j})_{n\times m}$									
k	number of requests	1	2	3	4	5				
n_k	number of students with k requests	64	20	12	4	3				
j	seminar ID	1	2	3	4	5	6	7	8	9
	seminar content	1	1	2	3	3	4	4	5	5
c_j	number of places per seminar j	12	12	12	12	12	12	12	12	12
r_j	number of requests per seminar j	10	13	19	15	13	29	27	20	25

5.2. Performance of assignment for real-world data set

Figure 2a shows the cumulative distribution function (CDF) of the ALOS ratio L (labeled as a ratio in the legend) and the fairness index F for the SEKO assignment for 1000 different assignments. This is compared to the deterministic optimal values from the optimization problem MU + F. We observe that the performance of the SEKO strategy is close to the optimum and that the variance of the SEKO assignment is rather small. We additionally compare this to the deterministic results from the FIFO assignment. We see that the ALOS ratio and the fairness index are significantly lower for FIFO than for SEKO.

Figure 2b shows the key performance indicators for the different assignment strategies and optimization problems. To be more precise, we are plotting the mean values for the SEKO assignment; the other values are deterministic. In all cases, the maximum utilization is reached as already discussed. However, we see that FIFO assignment leads to poor performance values regarding the ALOS ratio and fairness. The optimization problem MU is not considering fairness and returns an arbitrary solution, yielding in lower values for F and L. The differences between MU + 1 and MU + F are negligible. Again, we see that the SEKO assignment is close to the optimum. Hence, we conclude that SEKO assignment is performing very well in practice.

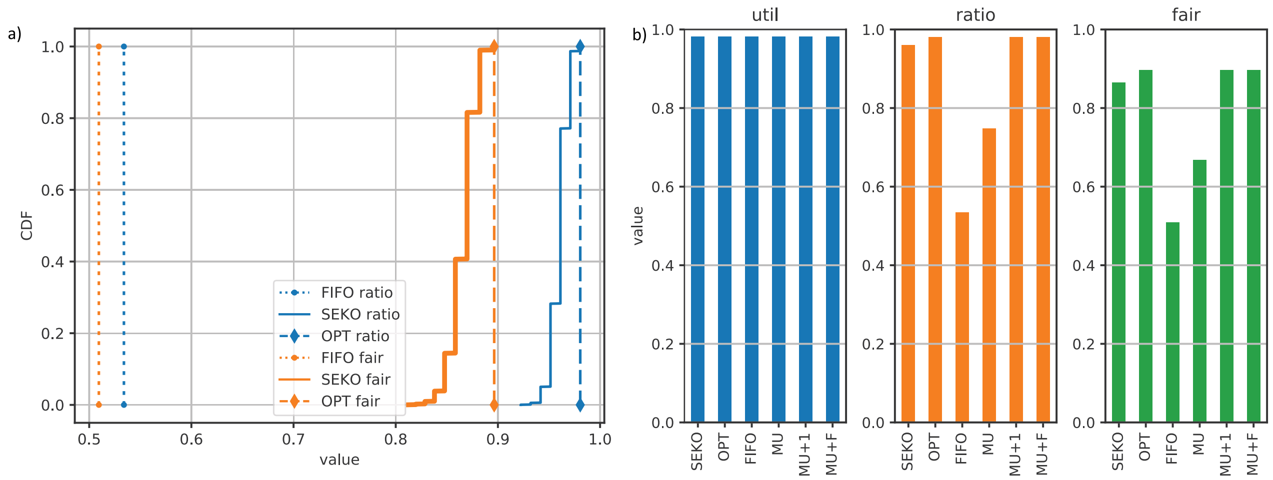

Figure 2. Real-world scenario with given demands for the various assignment strategies: a) the CDF of the ALOS ratio (RATIO) and the fairness index (FAIR) over 1000 different assignments of the SEKO strategy compared to FIFO and the optimal assignment, b) for SEKO, the mean key performance indicators, averaged over 1000 runs, is computed and compared to the deterministic values for FIFO and the solution of the optimization problems; RATIO is an abbreviation for the ratio of students who are assigned at least one seminar (ALOS); FAIR stands for the fairness index; MU reflects the results for the optimization problem which maximizes the number of assigned places; MU + 1 is the solution when at least one seminar is assigned per student, while the capacity of the MU problem is ensured; MU + F is the solution when fairness across students is maximized, while the capacity of the MU problem is ensured; OPT provides the maximum, i.e., optimal, value over all three optimization problems

5.3. Student perspective on assignments

Figure 3a computes the probability that a student gets i seminars assigned for each simulation run. The boxplot is over the simulation runs for the real data set. We see that most students get 1 seminar; however, to utilize the capacity, some students get several seminars assigned. This requires of course that several seminars are requested, cf. Table 2.

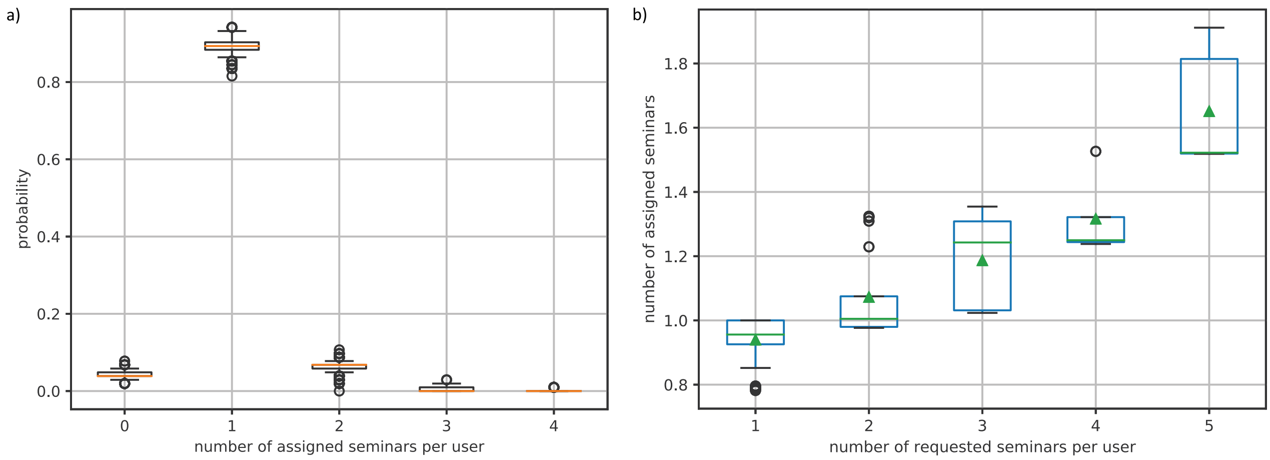

Figure 3. Student-centric evaluation of the real-world scenario with given demands for SEKO assignment: a) boxplot over 1000 runs of the probability of getting assigned i seminars vs. i, b) boxplot over 1000 runs of the number of assigned seminars vs. the number of requested seminars per user; there are $n = 103$ students and the SEKO assignment strategy is conducted 1000 times

Figure 3b correlates the number of requested and the number of assigned seminars. The more seminars are requested the more seminars are assigned on average. The Pearson correlation coefficient is 0.81.

However, please note the absolute numbers are close. For k requested seminars, the mean number m_k of assigned seminars is: $m_1 = 0.939$, $m_2 = 1.072$, $m_3 = 1.186$, $m_4 = 1.316$, $m_5 = 1.650$.

The question arises whether students therefore will simply demand many more seminars to improve the chances to get more seminars assigned. In practice, this may not be the case, since students will only register for seminars where they can participate if there are, e.g., some kind of registration fees. However, the 'all students request all seminars' (ALL) scenario gives interesting insights into fairness.

5.4. All students request all seminars scenario

We now consider that every student is requesting all seminars. There are 108 places that can be filled by 103 students. Hence, in the fairest scenario, 98 students get 1 seminar and 5 get 2 seminars. This leads to a maximum fairness $F^* = 0.914$. Figure 4a compares the optimal results from the MU + F assignment with the SEKO and FIFO assignment for the real-world scenario and the ALL scenario. While the FIFO assignment has worse performance in the ALL scenario than in the real scenario, the SEKO assignment is still very close to the optimum.

Figure 4b shows the mean number of assigned seminars per user over 5000 simulation runs. The real scenario reflects the requests by the users, yielding in several peaks. The ALL scenario nicely shows that there are no differences among users.

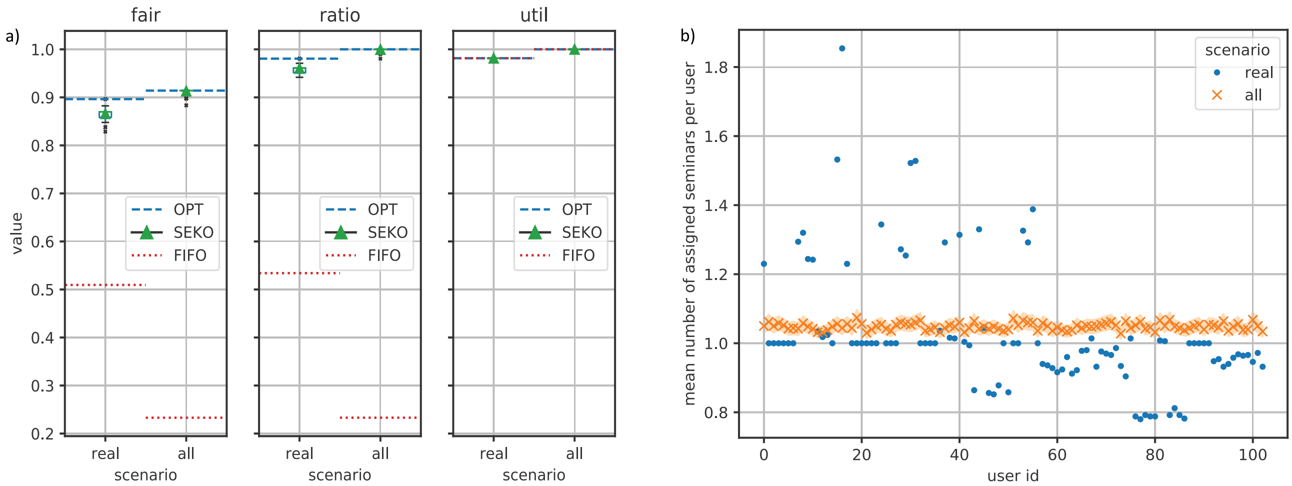

Figure 4. Real world scenario (labeled as real) vs. scenario where all students request all seminars (all): a) fairness, at least one seminar (ALOS) ratio, utilization; the figure shows the box plot for 5000 simulation runs of the SEKO assignment strategy; the triangle visualizes the mean; the values for the optimal assignment and the FIFO assignment are given; b) a mean number of assigned seminars per student; the 95% confidence interval around the mean for the scenario all s plotted; however, the confidence intervals are very small and hardly visible

6. Scalability and parameter sensitivity study

The performance of the assignment strategies is evaluated for increasing demand and larger scenarios. The performance of SEKO is compared to FIFO and the optimization problem MU + F. First, a parameter sensitivity study and demand analysis is conducted. Then, the performance of SEKO is compared to the optimal solution for data sets with a large number of seminars. We explicitly consider also the computational time indicating the limits of the ILP in practice and the need for heuristics in large scenarios.

6.1. Parameter sensitivity study and demand analysis

The demand analysis is conducted in which we are varying the number of requests per seminar and the maximally assigned places $A_{\max}$. Using the real-world data set, which has a ratio ρ of the total number of requests to a total number of available seminar places of $\rho = 1.53$. Then, we vary the demand ratio ρ and randomly draw requests according to the seminar popularity r_j for the seminars $j = 1, \ldots, m$, see Table 2.

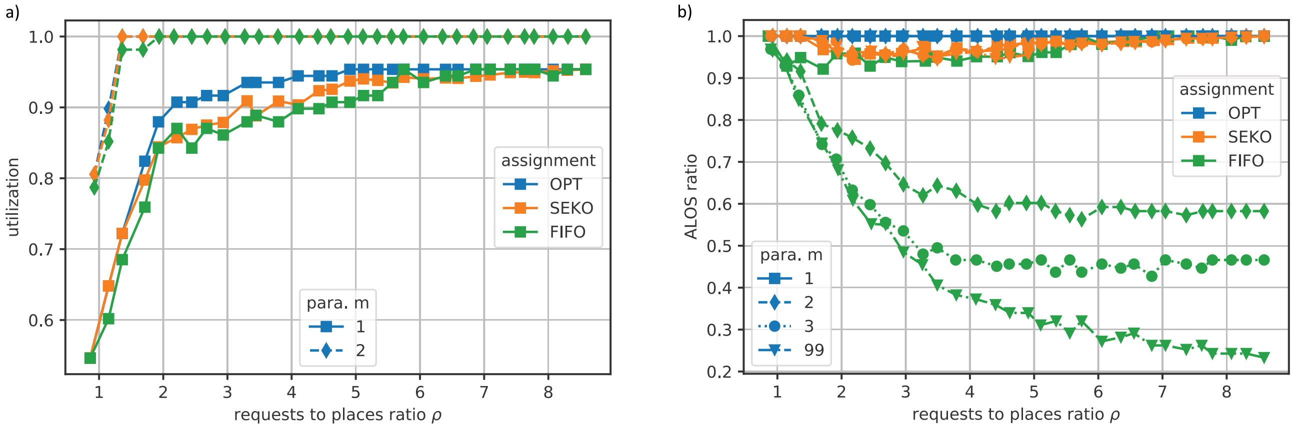

Figure 5. Real-world scenario with varying demands and maximally assigned places for the assignment strategies: a) utilization of the available places, b) ratio of students with at least one seminar; the demand is expressed as the ratio ρ of the number of requests to the number of available places; with $\rho = 1$, the original demands from the data set are obtained; the parameter m indicates the maximum number of places which are assigned to an individual student.

Figure 5a shows the utilization of the assignments depending on ρ. We see that for $A_{\max} = 2$, the real-world scenario utilizes all seminar places for all strategies if $\rho \geq 2$. However, if only a single seminar is allowed per user ($A_{\max} = 1$ or $m = 1$ in the legend), then we see small differences. SEKO is closer to the optimum than FIFO. However, this situation may be unrealistic, since we aim for 100% utilization without such restrictions. Taking a closer look at the ALOS ratio, for $m = 1$, the behavior is similar for the assignment strategies. However, for $m \geq 1$, the FIFO strategy shows a significantly worse performance than SEKO which is even worse with increasing demand ρ. The SEKO strategy, however, is in all scenarios close to the optimum.

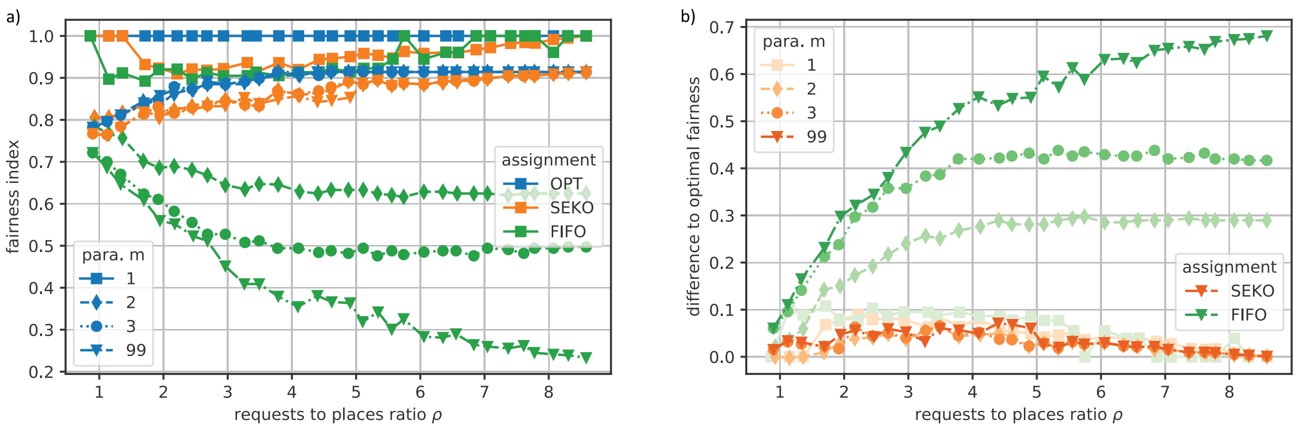

Figure 6. Fairness in a real-world scenario with varying demands and maximally assigned places: a) fairness index of different assignments, b) difference to fairness index of optimal assignment

Figure 6a is now considering the fairness index of the different assignments depending on the requests to places ratio ρ. Figure 6b visualizes the difference between the optimal fairness index of the SEKO and the FIFO assignment, taking into account a varying number of maximally assigned seminars per user ($A_{\max} = 1$ or $m = 1$ in the legend). Again, we see that the FIFO strategy strongly suffers in terms of fairness besides the ALOS ratio. In contrast, the SEKO strategy is very close to the optimum. Smaller differences are due to the random drawing and assigning of students to seminars.

In summary: the SEKO assignment achieves perfect utilization or efficiency, while fairness and user satisfaction concerning the ALOS ratio are very close to the optimum. Hence, we propose a very simple heuristic to solve the seminar assignment problem.

6.2. Assessment of scalability and performance for larger data sets

One critical aspect of solving ILPs is computational time. In order to stress the computation of the ILPs, a data set with a large number of seminars and students is required. To this end, we have created a model to generate large data sets based on the real-world data set. Then, we evaluated the computational time of the ILP "Maximizing Fairness and Utilization Problem" (MU + F).

6.2.1. Model for realistic larger data sets

Based on the real data, we have developed a synthetic model to generate a data set with an arbitrary number of seminars, while keeping the characteristics of the real data set as provided in Table 2: number of requests per seminar, how often a seminar is offered, constant number of seminar places, constant ratio of number of students to number of seminars. To scale the data set, the number of seminars is adjusted which is the input parameter of the large data set model.

[h]

Table 3. Generated large data set with requests of students for the different seminar types

Variable	Parameter	Value
m^*	number of seminars	input parameter
n^*	number of students	$\frac{nm^*}{m}$
c_j^*	number of places for a seminar $j = 1, \ldots, m^*$	12
Ω^*	total number of seminars	$12m^*$
r_j^*	number of requests per seminar j	drawn randomly from a discrete uniform distribution $r_j^* \sim \mathrm{DU}(8, 30)$, see fitting in Figure 7a
j	seminar content is offered j times	1 2 3
d_j^*	with probability d_j^*	0.3 0.6 0.1
R^*	total number of requests	$R^* = \sum_{j=1}^{m^*} r_j^*$ with mean $E[R^*] = E[r_j^*]m^* = 19m^*$
ρ^*	requests to places ratio	$\rho^* = R^*/\Omega^*$ with expected ratio $E[R^*]/(12m^*) \approx 1.583$
k	number of requests	1 2 3 4 5
n_k^*	probability that student requests k seminars	$\frac{64}{103}$ $\frac{20}{103}$ $\frac{12}{103}$ $\frac{4}{103}$ $\frac{3}{103}$

[1] We use the same notion for the variables and parameters of the model as for the real data set in Table 2, but mark them with an asterisk, e.g., m^* is the number of seminars of the large scale data set, while m is the number of seminars of the real-world data set. The original parameters of the real-world data set are provided in Table 2.

Table 3 provides an overview of the model parameters and how we adjusted them related to the real-world data set. We use the same notion for the variables and parameters of the model as for the real data set in Table 2, but mark them with '*'. For example, the key input parameter m^* reflects the number of seminars of the large scale data set, while m is the number of seminars of the real-world data set. The number of students generated in the larger data set is then $n^* = n \cdot \frac{m^*}{m}$ with the parameters n, m as specified in Table 2. Hence, we vary the number of seminars and keep the ratio of students to seminars constant. The number c_j^* of places per seminar j is kept constant.

Note that we sample how often a seminar is offered multiple times according to the distribution in Table 3. Thereby, the distribution is selected according to the real-world data set. A seminar is offered j times with probability d_j^*; we set $d_1^* = 0.3, d_2^* = 0.6, d_3^* = 0.1$. Hence, on average a seminar is offered 1.8 times, which is the average value from the real-world data set. The probability distribution function is depicted in Figure 7b.

Furthermore, the requests per seminar are randomly drawn. To this end, the number of requests per seminar is fitted with a discrete uniform distribution. Figure 7a shows the cumulative distribution function (CDF) of the number of requests per seminar. The square markers indicate the measurements from the real-world data set, while the solid line and the dot markers show the CDF for the discrete uniform distribution in the range between 8 and 30 requests per seminar j: $r_j^* \sim \mathrm{DU}(8, 30)$. Then, the expected number of requests per seminar is $E[r_j^*] = 19$. As we can see, the fitting is appropriate for our purposes to randomly draw the number of requests per seminar for an arbitrary number m^* of seminars. Thus, the expected characteristics of the generated data set are identical to the real-world data set.

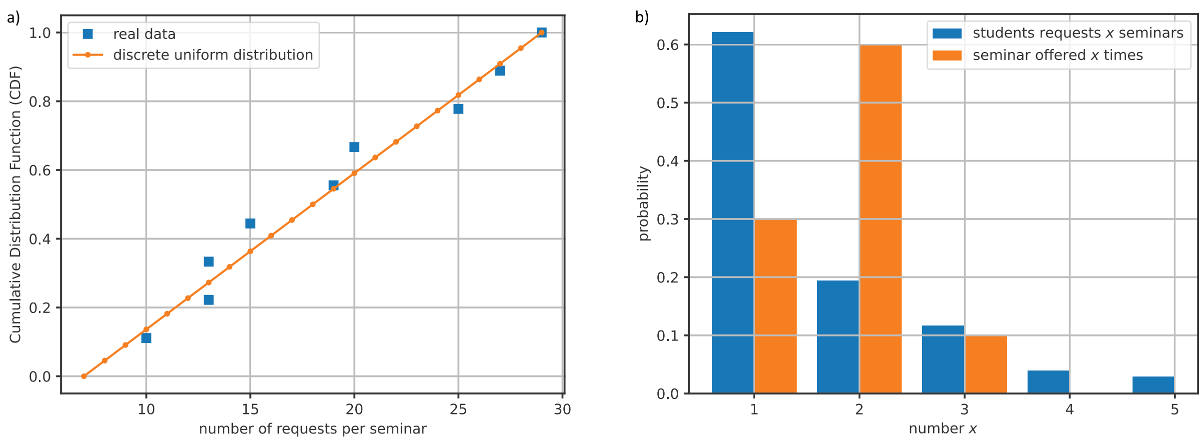

Figure 7. Probability distributions for generating the large data set based on real-world data set in Table 2: a) fitting the number of requests per seminar with a discrete uniform distribution in the range between 8 and 30 requests per seminar, b) discrete random probability distributions that a student requests several seminars (left blue bar) and that a seminar is offered multiple times (right orange bar)

6.2.2. Assessment of the computational time for larger data sets

In the following, we compare the computational times for solving the MU + F optimization problem as well as the SEKO strategy to assign students to seminars according to their requests. The number of seminars was varied from 25 to 500 in steps of 25. In addition, we considered 10 seminars as smallest number of seminars, reflecting the size of the real-world data set. The software and hardware specifications of the machines for conducting the computational time measurements are in the appendix in Table 4.

Figure 8a shows a box plot of the computational time to solve the ILP MU + F over several runs on a logarithmic scale. The maximum computational time was limited to 1200 s to solve the ILP. This results in 21 values of the number of seminars which were repeated 17 times. In total, solving the ILP for all parameters and repetitions required $31\,\mathrm{h}$ and $14.4\,\mathrm{min}$. In contrast, the SEKO strategy required a total computational time of $1.5\,\mathrm{min}$ only. The boxplots show that the computational time is significantly increasing with respect to the number of seminars and therefore the number of students requesting seminars. The computational time for MU + F and SEKO show similar behavior but on different time scales.

We take a closer look at the scalability of both assignment strategies. Figure 8b shows the average computational time over the several runs on a linear scale and we see a strong increase for the ILP. For the SEKO strategy, the increase is not visible on the linear scale. For the ILP, we can see that for 500 seminars, the maximum computational time of $1200\,\mathrm{s}$ is already reached. Solving the ILP is simply not scaling with the problem size. To support this claim, we have fitted the measurement points and obtained the following power functions $f(m) = am^k$ to obtain the computational time (in s) depending on the number m of seminars.

$$f_{\mathrm{MU+F}}(m) = 3.08 \times 10^{-6} m^{3.17} \tag{29}$$

$$f_{\mathrm{SEKO}}(m) = 9.49 \times 10^{-8} m^{2.59} \tag{30}$$

Although both fitting functions follow a power function, the exponent k is significantly larger for solving the ILP than for using a simple heuristic. In addition, the scaling factor a is more than 32 times larger for solving the ILP. As a consequence, the computational time for solving the ILP is not scaling with the number of seminars. For $m = 5000$ seminars, the computational time is $27\,569\,\mathrm{min} = 459.48\,\mathrm{h}$ for the ILP and $6.14\,\mathrm{min}$ for the SEKO strategy, respectively.

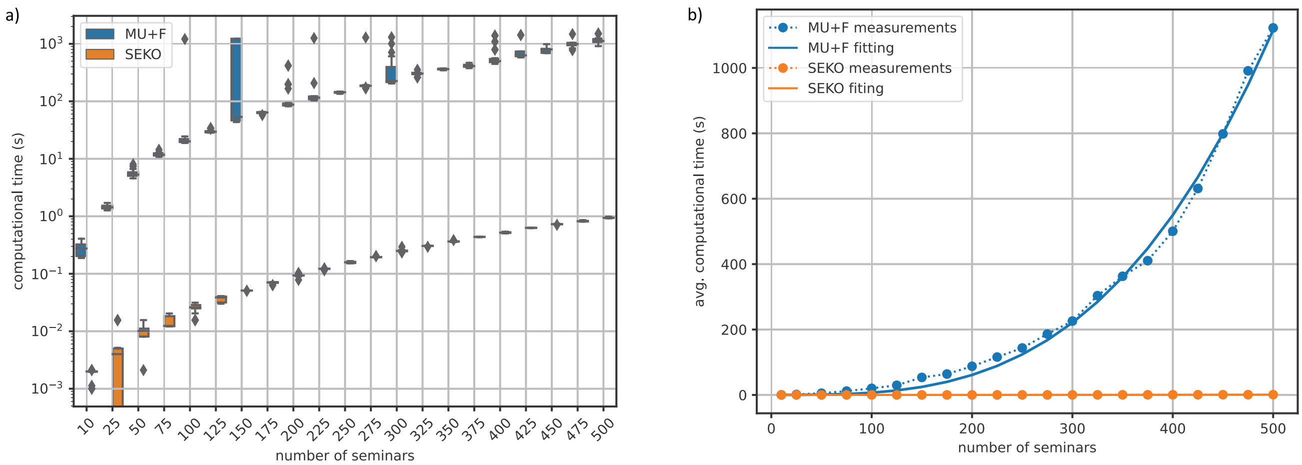

Figure 8. Computational time over several runs; the maximum computational time was limited to $1200\,s$: a) box plot of the computational time, b) mean computational time

The strong increase in the computational time to solve the ILP shows the need for heuristics like the SEKO strategy in case of a large number of seminars. However, the question remains if the SEKO strategy is close to the optimum in the large-scale scenario.

6.2.3. Performance comparison for larger data sets

The optimization problem (MU + F) which maximizes fairness while maximizing utilization returns the desired solution in practice. Per definition, the MU + F problem provides the maximum utilization, since this is a constraint of the optimization problem. Furthermore, the fairness index is maximized for MU + F, which includes also that the ALOS ratio is high.

As already discussed, the SEKO strategy leads to maximum utilization per design. For the real-world data set, we have already seen that the SEKO strategy is very close to this optimal solution (Section 5). However, the question is how the SEKO strategy performs for larger data sets. To this end, we now consider the ALOS ratio and the fairness index of the SEKO strategy in comparison to MU + F for the large data set in Figure 9.

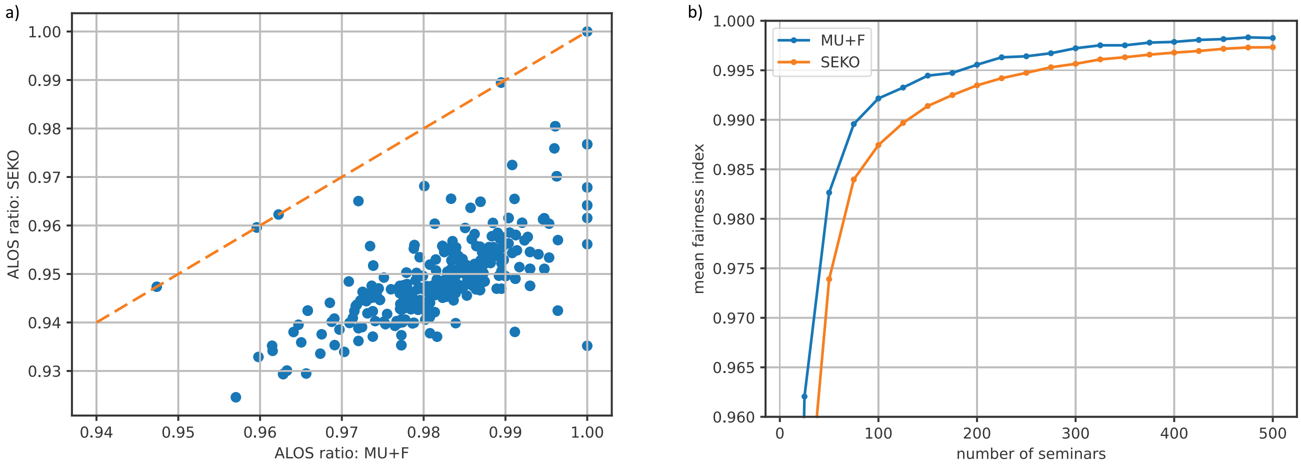

Figure 9. Performance comparison for larger data sets generated according to the model in Section 6.2: a) scatter plot of the ALOS ratio, b) fairness index averaged over several runs. Note that the range on the y-axis is very small to make the differences visible

Figure 9a shows the ALOS ratio for MU + F on the x-axis and for the SEKO strategy on the y-axis, respectively. Each plot in the scatter plot reflects a single generated request matrix, where the number of seminars is varied from 25 to 500 in steps of 25. Each configuration is repeated several times resulting in different request matrices. The dashed line indicates the ALOS ratio from the MU + F optimization. It can be seen that the SEKO strategy is always below MU + F, but the difference is rather small. In particular, the ALOS ratio is above 90% in all cases.

Figure 9b compares the fairness index depending on the number of seminars for the MU + F and the SEKO assignment, respectively. To be more precise, the average fairness index over the several runs is plotted. Again, we see that the SEKO performance is close to the optimum in the large data set scenario. The higher the number of seminars is, the higher the expected fairness index is. This is reasonable since a higher number of seminars gives more flexibility for distributing the available seminar places. From the results, we conclude that the performance of the SEKO strategy is scaling with the number of seminars and is close to the optimal results of the MU + F strategy. However, the computational efforts are much lower for the SEKO assignment than for solving the MU + F optimization problem.

7. Conclusions

The problem investigated in this article is the assignment of students to seminar places taking into account the requests of students, the limited capacity of seminar places, as well as multiple offers of the same seminar. We define quantitative measures for key objectives of the assignment strategies which are the utilization of available seminar places, the ratio of users with at least one seminar, and fairness in terms of the number of assigned seminars. Various optimization problems were formulated by means of ILP to obtain the maximum performance values for utilization, ALOS ratio, and fairness.

A popular assignment strategy is FIFO assignment due to its simplicity. However, based on a real-world data set and a parameter sensitivity study we showed that the FIFO strategy strongly suffers in terms of fairness and ALOS ratio. Therefore, we provided the SEKO strategy which is based on an iterative random selection of students to seminars by taking into account the number of already assigned seminars. This achieves almost the maximum potential fairness in real word scenarios, but also in artificial scenarios where all users are requesting all seminars. In particular, the SEKO strategies enforces that the ALOS ratio is maximized which is also improving fairness and satisfaction of students with the assignment. Our numerical results of SEKO are based on a simulative performance evaluation while the ILP are numerically solved with a linear program. The results show that the FIFO assignment suffers in real-world situations regarding fairness, while the SEKO assignment is close to the optimum obtained by the ILP. To investigate the scalability concerning the computational time to solve the ILP, a model is derived to generate large data sets containing a desired number of seminars. The model reflects the characteristics of the real-world data set. As a result, we have seen that the ILP is not scaling with the problem size, which requires a heuristic like our SEKO approach. In contrast, the SEKO approach scales concerning the computational time and is also close to the optimum for larger scenarios.

In practice, however, the MU + F optimization problem may be used if the problem size is limited in terms of number of seminars and students. This will provide the best assignment in terms of utilization and fairness. Still, it may be difficult to explain how the assignment was done for some audiences. This is another advantage of the SEKO approach which allows a simple, but meaningful explanation to the students or other people – besides its simplicity and low computational efforts, although yielding very good performance. SEKO randomly selects students by drawing a dice which takes into account how many seminars are already assigned. This is easy to understand and may be well perceived by people which are not familiar with optimization problems and their solutions.

References

[1] Al Mulhim, E. N., and Eldokhny, A. A. The impact of collaborative group size on students' achievement and product quality in project-based learning environments. *International Journal of Emerging Technologies in Learning (iJET) 15*, 10 (2020), 157–174.

[2] Avi-Itzhak, B., Levy, H., and Raz, D. Quantifying fairness in queuing systems: Principles, approaches, and applicability. *Probability in the Engineering and Informational Sciences 22*, 4 (2008), 495–517.

[3] Cattrysse, D. G., and Van Wassenhove, L. N. A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research 60*, 3 (1992), 260–272.

[4] Chen, L.-T., and Liu, L. Social presence in multidimensional online discussion: The roles of group size and requirements for discussions. *Computers in the Schools 37*, 2 (2020), 116–140.

[5] Efraimidis, P. S. Weighted random sampling over data streams. In *Algorithms, Probability, Networks, and Games*. vol. 9295 of Lecture Notes in Computer Science, C. Zaroliagis, G. Pantziou, and S. Kontogiannis, Eds., Springer, Cham 2015, pp. 183–195.

[6] Garg, N., Kavitha, T., Kumar, A., Mehlhorn, K., and Mestre, J. Assigning papers to referees. *Algorithmica 58*, 1 (2010), 119–136.

[7] Hossfeld, T., Skorin-Kapov, L., Heegaard, P. E., and Varela, M. A new QoE fairness index for QoE management. *Quality and User Experience 3*, 1 (2018), 4.

[8] Jain, R. K., Chiu, D.-M. W., and Hawe, W. R. *A quantitative measure of fairness and discrimination for resource allocation in shared computer systems*. DEC Research Report TR-301, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, 1984.

[9] Magnanti, T. L., and Natarajan, K. Allocating students to multidisciplinary capstone projects using discrete optimization. *Interfaces 48*, 3 (2018), 204–216.

[10] Öncan, T. A survey of the generalized assignment problem and its applications. *INFOR: Information Systems and Operational Research 45*, 3 (2007), 123–141.

[11] Rezaeinia, N., Góez, J. C., and Guajardo, M. Efficiency and fairness criteria in the assignment of students to projects. *Annals of Operations Research 319*, 2 (2021), 1717– 1735.

[12] Roeder, T. M., and Saltzman, R. M. Schedule-based group assignment using constraint programming. *INFORMS Transactions on Education 14*, 2 (2014), 63–72.

[13] Wang, F., Chen, B., and Miao, Z. A survey on reviewer assignment problem. In *New Frontiers in Applied Artificial Intelligence. 21st International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. IEA/AIE (Wrocław, 2008)*, vol. 5027 of Lecture Notes in Computer Science, N. T. Nguyen, L. Borzemski, A. Grzech, and M. Ali, Eds., Springer, Berlin 2008, pp. 718–727.

[14] Wierman, A. Fairness and scheduling in single server queues. *Surveys in Operations Research and Management Science 16*, 1 (2011), 39–48.

A. Pseudo code of FIFO and SEKO assignment strategy

Listing 1 The SEKO assignment strategy in Python-like pseudo code

```
def assignmentSEKO(n, # number of users (int)
                   m, # number of seminars (int)
                   s, # number of places in seminar j (array of size m)
                   x, # request matrix for user i and seminar j (size n×m)
                   S, # set of seminars with same content type k (array of size t)
 A_max , # maximum number of seminars being assigned to user (int)
 α # factor increases prob. to have at least one seminar (float)
                  ):

    # initialize assignment matrix of size n×m
    y = zeros((n,m)) # y[i,\,j]==1 means that user is is assigned to seminar j

 # list of remaining seminars
 L = range(m) # seminars are numbered from 0,..,m-1

 while L is not empty: # iterate over seminars in ascending order
        # determine seminar from remaining with minimum number of requests
        r = x[:,L].sum(axis=0) # number of requests r_i for i in L
        w = argmin(r) # determine next (least popular) seminar out of L
        i = L.pop(w) # seminar i is returned and removed from list L

 registeredUsers = argwhere(x[:,i]>0) # list of users registered for i
 # set of users with max. number of seminars assigned
 alreadyMaximumAssignedSeminars = argwhere(y.sum(axis=1) >= A_max)
 eligibleUsers = setdiff(registeredUsers, alreadyMaximumAssignedSeminars)

        # assign all eligible users to seminar, if enough places available
 if len(eligibleUsers) <= s[i]:
 y[eligibleUsers,i] = 1 # users are assigned to seminar i

            # delete other seminars of same content type from user requests
 for s in S_i: x[eligibleUsers,s] =0

 else: # randomly select users for seminar i based on probabilities
            A = y[eligibleUsers,:].sum(axis=1) # list of assigned seminar places
            curMax = max(A) # currently maximum number of assigned seminars
            p = (curMax+1.0-A) # array of (unnormalized) prob's to select user

            # take care of users who were not yet assigned any seminars
            noseminarsofar = (A==0)
            p[noseminarsofar] *= α # increase prob. by factor α
            p = p/p.sum() # normalize probabilities

            # randomly select s_i users out of the set of eligible users (without replacement) based on the probabilities
            select = random.choice(eligibleUsers, replace=False, size=s[i], p=p)

 y[select,i] = 1 # randomly selected users are assigned to seminar i

 # delete other seminars of same content type from user requests
 for s in S_i: x[select,s] =0

 return y # assignment matrix for user i and seminar j (size n×m)
```

Listing 2 The FIFO assignment strategy in Python-like pseudo code

```
def fifoAssignment(n, # number of users (int)
                   m, # number of seminars (int)
                   s, # number of places in seminar j (array of size m)
                   x, # request matrix for user i and seminar j (size n×m)
                   S, # set of seminars with same content type k (array of size
                                                t)
 Amax # maximum number of seminars being assigned to user (int)
                  ):

 # initialize assignment matrix of size n×m
    y = zeros((n,m)) # y[i,\,j]==1 means that user is is assigned to seminar j

 # list of remaining seminars
 L = range(m) # seminars are numbered from 0,..,m-1

 while L is not empty: # iterate over seminars in ascending order
        # determine seminar from remaining with minimum number of requests
        r = x[:,L].sum(axis=0) # number of requests r_i for i in L
        w = argmin(r) # determine next (least popular) seminar out of L
        i = L.pop(w) # seminar i is returned and removed from list L

        registeredUsers = argwhere(x[:,i]>0) # list of users registered for i
 # set of users with max. number of seminars assigned
 alreadyMaximumAssignedSeminars = argwhere(y.sum(axis=1) >= Amax)
 eligibleUsers = setdiff(registeredUsers, alreadyMaximumAssignedSeminars)

        # take the first s[i] user requests from the request matrix
        select = eligibleUsers[:s[i]]
 y[select,i] = 1 # # users are assigned to seminar i

 # delete other seminars of same content type from user requests
 for s in S_i: x[select,s] =0

 return y # assignment matrix for user i and seminar j (size n×m)
```

B. Software and hardware specifications for assessing the computational times

Table 4. Specification of the hardware and software used to solve the ILP and assign students using the SEKO strategy

Hardware	Specification
Processor	Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz
CPU clock speed	3.60 GHz
RAM	32 GB
Operating system	Windows 10 Pro, Version 21H2, 64bit OS
Python version	3.7.11
NumPy version	1.21.5
SciPy version	1.7.3
Pandas version	1.3.5
MIPs version	1.13.0 (Python tools for Modeling and Solving Mixed-Integer Linear Programs)