AKADÉMIAI KIADÓ

Exploring fair scheduling aspects – Through final exam scheduling

Szilvia Jáhn-Erdős* and Bence Kővári

Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Budapest, Hungary

Pollack Periodica • An International Journal for Engineering and Information Sciences

19 (2024) 1, 151–156

DOI: 10.1556/606.2023.00780

Received: December 29, 2022 • Revised manuscript received: July 21, 2023 • Accepted: August 4, 2023
Published online: October 13, 2023

ORIGINAL RESEARCH PAPER

ABSTRACT

Fair treatment of individuals in a scheduling task is essential. Unfairness can cause dissatisfaction among workers, faster obsolescence of work tools and underutilization of others. The literature's definitions vary, and there is no clear definition of general scheduling tasks.

This article explores fair scheduling through the lens of final exams, aiming to extend decision support system methodologies. It proposes a method based on Lipschitz mapping to measure fairness and presents a pseudo-algorithm for estimating optimal trend lines.

The model and the algorithm are demonstrated using the example of final exam schedules. In this way, two feasible solutions can be measured and compared in terms of fairness.

KEYWORDS

fairness, scheduling, Lipschitz-mapping, workload, operations research, final exam scheduling

1. INTRODUCTION

The concept of fairness is becoming increasingly important in many areas of life, in decision support systems, legal and economic issues, and optimization. However, the idea of fairness is not quantifiable, and it can be challenging to define what is meant by it. According to Cambridge Dictionary, fairness is defined as the quality of treating people equally or in a way that is right or reasonable [1]. The aim is to translate this mathematically.

In scheduling tasks, especially when the assignment of people is to determine, this aspect also comes to the foreground. An example of a real-life problem dealing with the scheduling of people is the scheduling of final exams, which is a special sub-task of scheduling problems where specific requirements constrain the state space. For schedules, there is usually a state space and requirements that a schedule must meet. Based on these, an objective function is settled, containing the schedule conditions. In the case of a final exam schedule, the constraints are separated into hard and soft requirements. If a hard requirement is violated, the assignment is not considered acceptable, while the soft requirements aim to meet as many of them as possible. These are built in the objective function as a weighted sum. The fairness requirement, however, is a constraint that cannot be embedded into the other constraints because it cannot be quantified explicitly. It cannot be included in the objective function but must be considered in a separate dimension. This is because one cannot say precisely what makes a final exam distribution fair. It could consider, for example, that for examiners, all examiners should be allocated to the same number of exams. Still, some examiners are teachers in 6–7 subjects, while others may only be allocated to 1 subject. Should there be an even distribution per subject? This may not give good results either, as some subjects require 1-2 students to be examined, while others require several hundred students to be examined individually. However, the aim is to treat instructors of similar competence similarly and fairly (as it is quoted in [1]), and none of the intuitive formulations can explain this. However, if one looks at the notions of fairness found in literature, it can be found that there is no general method or convention for dealing with fairness in the context of scheduling.

*Corresponding author.
E-mail: Erdos.Szilvia@aut.bme.hu

AKJournals

In other optimization tasks, however, the need for a general definition of fairness has been raised, for example, in decision support systems Grgic-Hlaca et al. [2]. The input information of the task is similar since a decision must be made from a given set of input properties. A typical example is lending, where the input information is the customer's data and income circumstances, which should be used to decide whether he/she can get a loan. In this case, similar input characteristics (e.g., whether a person can be a chairperson or secretary in a final exam, how many subjects they can sit for, etc.) should be used to determine how many exams it would be fair for them to sit for in the final schedule. In both cases, an essential factor is to never look at fairness as an individual but to define it by comparing candidates to each other.

In this paper, an attempt is made to extend the methodologies used in decision support systems for complex scheduling tasks, and an example is provided of how it can be used through the task of final exam scheduling.

2. BACKGROUND

The following section discusses the context in which the definition of fairness has been used in literature for scheduling tasks and how fairness has been successfully defined for other optimization tasks, which can also help to create a general definition in scheduling.

2.1. Fairness definitions in scheduling problems

In many scheduling challenges, the concept of fairness is not taken into account at all in any sense (e.g., in articles of Neelakantan et al. [3] and Seyyed et al. [4]) instead, a set of requirements to be satisfied is given in the objective function. However, many articles use the word "fair", and their interpretation can be divided into two categories.

In many scheduling problems, the task is to allocate a resource in which the definition of fairness is straightforward. In this case, a given resource is allocated equally or according to some given objective function weighting scheme. Some examples are presented below.

The processor utilization of computers is an example where the aim is to distribute the tasks or memory equally. A fair resource scheduler was presented by Shen et al. [5] for flash-based solid-state drives. There are also many recent kinds of research for scheduling algorithms for persistent memory, e.g., the article by Zhao et al. [6]. Even neural network-based control mechanisms were examined by Budhiraja et al. [7]. Fair queuing is also considered in the problem of allocating networks' bandwidth. For example, adaptive bandwidth binning is considered by Hong et al. [8], and weighted fair queues by Pan et al. [9].

Sometimes, the shift scheduling problem could be considered the same problem when the employees should be scheduled for equal timeslots. In his article, Ikeda et al. [10] worked on the same problem.

Another set of definitions of fairness in the literature deals with defining fairness in a concrete problem-specific case. In these cases, the notion of fairness is complex, it is not just the distribution of a resource, but fairness is formulated in terms specific to a given problem.

Woumans et al. [11] article considers timetabling fair if students have sufficient study time between exams.

According to Mansini et al. [12], fairness is accomplished when all doctors are assigned to all kinds of tasks in a hospital.

In his theory, Uhde et al. [13] considered collaboration with stakeholders as the key to achieving fair scheduling. Before assignment, interviews are conducted with hospital staff, and fairness is measured against their criteria.

In his paper Vetschera et al. [14] discusses how partial information in group decisions impacts fairness. He claims that fairness depends on how many different individual's interests are reflected fairly.

In his article, Jütte et al. [15] defined fairness as a soft constraint where the goal was an even distribution of the unpopular duties among depots. While according to Breugem et al. [16], fairness relates to the distribution of work among the roster groups, and he modeled fairness via hard and soft constraints in his context.

These cases have in common that they all gave a problem-specific definition and translated the problem of fairness into a specific task requirement, which thus can neither give a general definition nor measure actual fairness.

2.2. Fairness in decision support systems

The question of fairness can also be found in many other areas of life, e.g., social, ethical and legal issues. There is even a conference [17], where all areas of fairness are covered.

One of the most significant areas of optimization is the decision support systems to focus on fairness. Most of these efforts are in this area, so the following discusses the most relevant concepts.

The article of Verma et al. [18] on definitions of fairness summarizes what is considered fair in a credit decision support system (examines racial and gender discrimination in banking systems). He discusses different ethical issues and provides mathematical formulations of definitions. One of the most significant of these models is Dwork et al. [19] article, which uses the Lipschitz mapping.

Dwork applies the Lipschitz continuity property of functions to probability distributions in the following way, which he calls the Lipschitz mapping. Here, the mapping M is interpreted for each input V, on outputs $\Delta(A)$, with distance definitions d and D. (He considers different distance metrics.) This relation is shown in in Eq. (1).

Fairness is interpreted as a linear programming problem to optimize a given loss function considering d and D distances,

$$\text{If } M : \mathbf{V} \rightarrow \Delta(A) \text{ and } d : \mathbf{V} \times \mathbf{V} \rightarrow \mathbb{R}, \text{ then } D(M(x), M(y)) \leq d(x,y) \forall x, y \in \mathbf{V}. \tag{1}$$

In previous research, Erdős et al. [20] examined the model for scheduling tasks based on Dwork's article. So far, only the challenges have been interpreted; the actual model and algorithm are presented in the following.

3. PROPOSED METHOD

The task is to apply the concepts of fairness for planning schedules so that the fairness of individual schedules can be measured. Based on Lipschitz mapping, the goal is to ensure that Eq. (1) is applied to the individuals to be scheduled.

The scope of the study is scheduling tasks where the resources are not homogeneous, so either different types of resources have to be assigned or each resource has different capabilities, i.e. not all individuals can perform all tasks.

Formally, a criterion taken from decision support systems (see Dwork et al. [19]) is that the distance between the distributions of the outputs of any two individuals is always less than or equal to the distance between the input properties of the same individuals. If the concept of Lipschitz mapping is defined in this way, it can also be interpreted for scheduling tasks. On this basis, it can be seen that the task is to schedule individuals with similar properties in a similar way, i.e., the closer two individuals are to each other in terms of properties (e.g., abilities), the more critical it is for their final scheduling (e.g., load number) to be close to each other. In this case, this means that the input information (e.g., how many and what tasks each individual can perform, how many of the tasks to be performed) determines what the ideal number of tasks to be taken on in the fair assignment would be. However, this can never be given individually; this information needs to be interpreted by comparing individuals in relation to each other.

The aim of this research is not to produce fair schedules but to find out how fair a completed schedule is. However, to interpret this, the definition of a fair schedule is needed based on the concept described by Dwork et al. [19].

Introduce the following definitions and notations: Let $\mathbf{V}$ denote the set of individuals, N the number of individuals, and n the number of input attributes, i.e., all the information is taken as a basis for the computation of the fairness to be interpreted on it (this means all the influencing factors). The $\mathbf{x}_i \in \mathbb{R}^n$ vector aggregates the input properties of an individual i. Given these, $\mathbf{V}$ is defined as it can be seen in (2):

$$\mathbf{V} = \{\mathbf{x}_i |\ \mathbf{x}_i \in \mathbb{R}^n, i \in N\}. \tag{2}$$

The $s(\mathbf{x})$ gives the output information that is considered in terms of fairness, $S(\mathbf{V})$ denotes the same for the whole schedule, i.e., $S(\mathbf{V}) := \{s(\mathbf{x}_1), s(\mathbf{x}_2), \cdots\}$. It is important to interpret this for a schedule S that satisfies all the basic requirements of the scheduling task, i.e., only feasible solutions are considered (S gives the feasible solution to the given scheduling task).

Definition: $T(\mathbf{x})$ trend-line tells us what the output of an individual would be in the ideal case, i.e., in the optimal case, $s(\mathbf{x}) = T(\mathbf{x})$.

Definition: A distributed scheduling is *K-fair*, if and only if for $\mathbf{x} \in \mathbf{V}$, $s(\mathbf{x})$ is the element of the hyperspace bounded by $T(\mathbf{x}) \pm K$, $K \in \mathbb{R}$. In this case, $S_k(\mathbf{V})$ denotes a *K-fair* scheduling.

Notion: K_T denotes K value of a *K-fair* scheduling, which belongs to $T(\mathbf{x})$.

Definition: Fair-optimal scheduling is a *K-fair* scheduling, for which it is true that the value K is the smallest for all other schedules. This means that the schedule S_{k0} is fair-optimal, where $k_0 \leq k, \forall S_k(\mathbf{V})$.

However, finding the trend line $T(\mathbf{x})$ is not trivial. Finding the optimal value of $T(\mathbf{x})$ is an iterative process in which the goal is to estimate the optimal value of $T(\mathbf{x})$ in a reverse algorithm.

The steps of the pseudo algorithm are the following.

1. S schedule is given, looking for $T(\mathbf{x})$, the corresponding K_T;
2. Denote T^* the polynomial curve that best approximates $T(\mathbf{x})$, $\mathbf{V}_0 := \mathbf{V}$;
3. Normalize the data, so the distance calculation between dimensions of different orders of magnitude does not cause distortion or numerical error;
4. Detect outliers and remove them from the data set. $\mathbf{V}_0 := \mathbf{V}_0 \setminus \{\mathbf{x}_{outlier}\}$. Practical algorithms for very outliers would significantly distort the expected results. Furthermore, $i := 1$ and $\mathbf{V}^* := \mathbf{V}_0$;
5. Fit a polynomial to V_0 based on a heuristic; this trend line will be T_i. Calculate the K-fairness of T_i. $T_0 := T_i$;
6. $\mathbf{V}^* := \mathbf{V}^* \setminus \{\mathbf{x}_j\}$, where $\mathbf{x}_j$ element is the “furthest” from the trend curve. The most significant improvement in the error function can be achieved by removing this element. (The furthest element causes the most significant deviation in absolute value in the error function).
7. Fit a new polynomial (T_{i+1}) to the new $\mathbf{V}^*$. Calculate K-fairness of T_{i+1} for $\mathbf{V}^*$;
8. If $K_{T_{i+1}} < K_{T_i} - \varepsilon$, ($\varepsilon > 0, \varepsilon \in \mathbb{R}$ is an arbitrary small number), then $i := i + 1$ and GOTO 5;
9. $T^* = T_0$, to obtain the curve of the fair-optimal scheduling trend, which is K_{T_0}-fair over $\mathbf{V}_0$.

4. CASE STUDY

The example of the final exam scheduling illustrates the practical application of the methodology described above. This study is based on a real data set, the final exam schedules of the Department of Automation and Applied Informatics, Budapest University of Technology and Economics. The schedule is an oral exam session, where students take the exam individually in front of a board of 5–6 members in parallel rooms. Up to 300 students may sit the exam in a semester, which must be completed in two weeks. The schedule has to comply with several rules described in the examination regulations. In addition, several human factors must be considered, e.g., considering availability or assigning instructors in blocks if possible. Another important aspect was to try to distribute the instructors’ workload as fairly as possible. But as this should be interpreted as a separate dimension, for the reasons explained earlier, the model presented below will be used to test the fairness of previous years' manual final exam distributions.

Since only schedules are considered that satisfy all requirements except fairness, (and are therefore outside the scope of this present analysis), they can be taken as the default. Thus, it is possible to examine only the fairness conditions, filtering out the irrelevant factors.

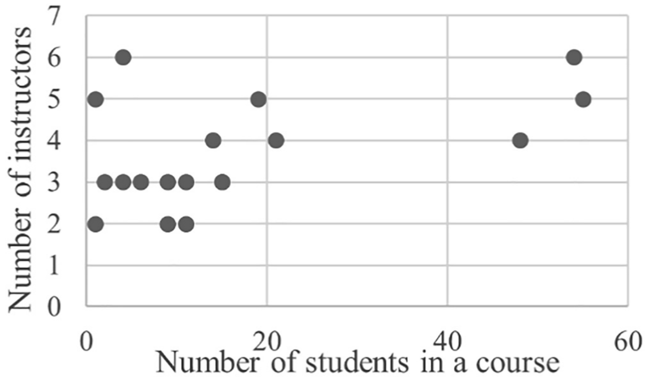


Fig. 1. Statistical information about courses

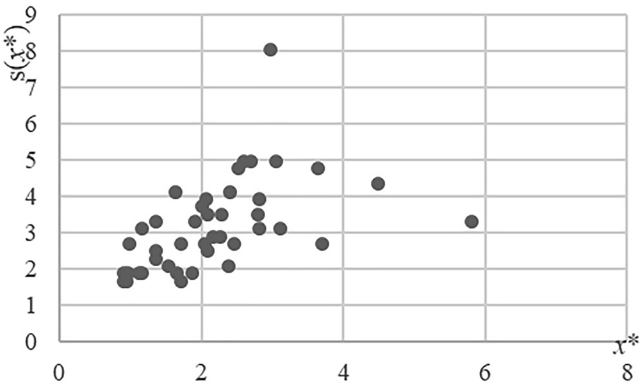


Fig. 2. The normalized input and output data of instructors

For the purpose of the present case study, let us take the examiners as a basis, one or two of whom must sit on each committee, depending on whether the student who is taking his final examination is required to take one or two courses. The set of optional examination courses is given, as well as the instructors who can examine a course. Only an instructor who is also an instructor of a course can be an examiner for a subject. It is also important to note that some instructors may be examiners for more than one subject, depending on the number of subjects they teach. In most cases, an instructor is expected to be an examiner for more than one subject, but the distribution of this is quite variable, as it is shown in Fig. 1, where every course belongs to a dot.

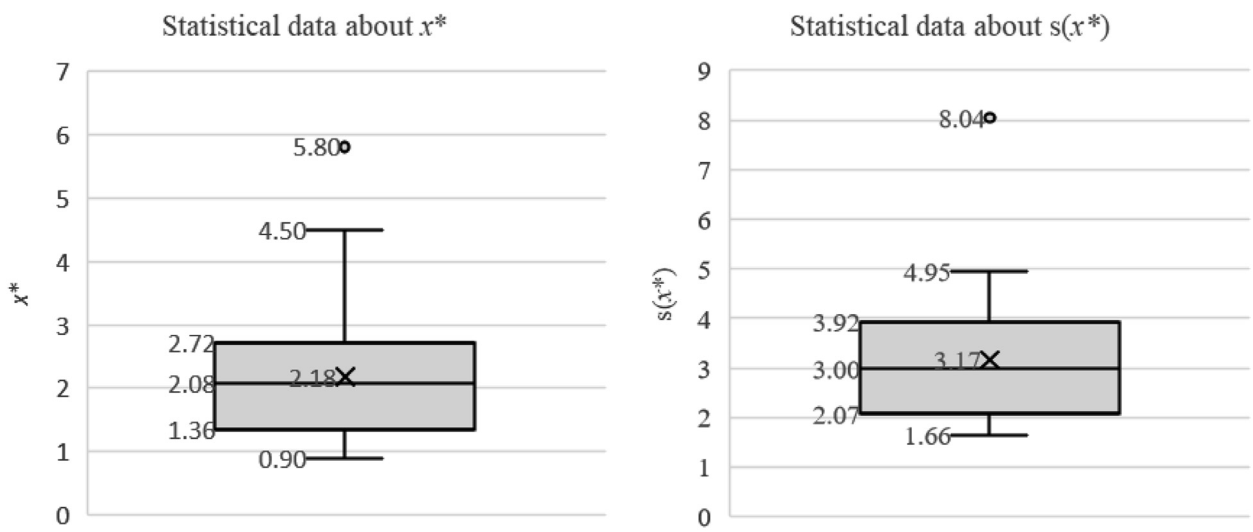
The fairness of a given schedule *S* is determined as follows, based on the algorithm presented in the previous section.

Initially, $\mathbf{V}_0 = \mathbf{V}$, i.e., the entire set of individuals, in this case all examiners in the final exam schedule.

In the final examination schedule, three attributes determine the input data of an examiner (how many courses he/she can examine, how many students are examined in each course, and how many additional instructors can examine each course). These data compose the $\mathbf{x}$ attribute vector of an examiner. In the following example, $\mathbf{x}$ is normalized and then the mean value is calculated from these data as the input property of each examiner, which is $\mathbf{x}^*$.

In this example, the function $s(\mathbf{x})$ gives the number of times in the final assignment that instructor with $\mathbf{x}$ attribute was scheduled. These can also be normalized and reduced to two dimensions to plot $s(\mathbf{x}^*)$, as it is shown in Fig. 2.

The next step is to filter out outliers. To do this, the distribution of the above data is examined and shown in Fig. 3.



a) x^* b) $s(x^*)$

Fig. 3. Distribution and locals

These also show that there are both $\mathbf{x}^*$ and $s(\mathbf{x}^*)$ that fall outside the $Q3 + 1.5 \cdot IQR$, which is used for outlier detection Schwertman et al. [21].

Let us take these from $\mathbf{V}_0$ and, after filtering them out, fit the first polynomial to the data of the set $\mathbf{V}_0$, this will be $T_1(\mathbf{x})$. The value of the error of $T_1(\mathbf{x})$ in this example is calculated as R^2, the square of the Pearson correlation coefficient, and is denoted by *Err*. From the possible polynomials, polynomial P_k of degree k is chosen, where $Err(Pk+1) <= Err(Pk) \cdot 1.03$, i.e., the error function would not improve by more than 3% for a degree higher than this, thus eliminating overfitting. The R^2 values and improvements for each polynomial are it is shown in Table 1. It shows that the polynomial of degree 3 will be chosen since an improvement of less than 3% can be achieved at degree 4. After 1st iteration $R^2 = 0.44$, $T_1(x^*) = -0.32x^{*3} + 1.88x^{*2} - 2.45x^* + 2.96$, and $K_T = 1.54$.

In the 2nd iteration the individual causing the largest deviation in the error function is removed from $\mathbf{V}^*$, as R^2 can then be improved to 0.4895 and K-fairness reduced to 1.43 in the new $\mathbf{V}^*$. Since this improvement is significant, the 2nd iteration is on hold.

Table 2 shows the $T(\mathbf{x})$ curve fitted after each iteration, the R^2 value obtained, the K value obtained, and the improvement over the previous iteration.

In the 4th iteration, no improvement is achieved, and there is degradation, so at the end of iteration 3, the algorithm stops, and the $T(x^*)$ obtained in iteration 3 becomes the best approximation searched by the algorithm. Resetting $\mathbf{V}_0$, it can be obtained that the scheduling S is *1.63-fair*. The final trend fit on the aggregate x^* is shown in Fig. 4.

Of course, this data alone does not provide much information, but the method itself is essential, as the determination of fairness can be achieved. By doing this, a fairness value is assigned to schedule S, and several different schedules can be compared to each other in terms of fairness.

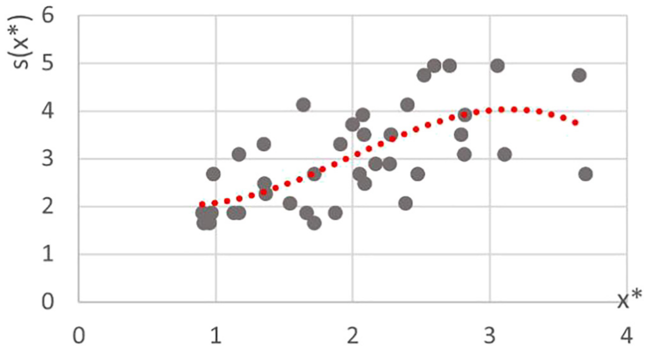


Fig. 4. Fitting $T(x^*)$ curve for $s(x^*)$

5. CONCLUSION

Previously, there was no uniform approach or definition of fairness in the literature for scheduling tasks. The possibility of creating a general mathematical concept is explored for evaluating and comparing the fairness of heterogeneous-resource allocating scheduling algorithms independently of the specific problem.

A model was created based on the Lipschitz mapping used in decision theory. The basic idea and the mathematical background of this model were interpreted for general scheduling problems, and an algorithm was constructed to compute the fairness of scheduling independent of the properties of the specific scheduling. With this algorithm, an approximation was also provided of what ideally expected output information would be for an individual with specific input data. The algorithm's operation on a real data set was also derived using a concrete example. Namely, this method was used to measure the inter-examiner fairness of a final exam assignment. This algorithm helps to compare how several different scheduling algorithms perform in terms of fairness.

Table 1. Performances of the trend-polynomials in 1st iteration

Degree of the polynomial	R^2	Improvement compared to the previous degree
1	0.3960	
2	0.4209	6.29%
3	0.4425	5.13%
4	0.4527	2.31%
5	0.4623	2.12%
6	0.4667	0.95%

Table 2. Iterations of the proposed algorithm

Iter.	$T(\mathbf{x}^*)=$	R^2	K_T over $\mathbf{V}^*$	Improve
1.	$-0.3158\,x^{*3} + 1.8819\,x^{*2} - 2.4521\,x^* + 2.9568$	0.4425	1.5310	
2.	$-0.4209\,x^{*3} + 2.626\,x^{*2} - 4.0261\,x^* + 3.8791$	0.4895	1.4311	6.52%
3.	$-0.4572\,x^{*3} + 2.8219\,x^{*2} - 4.2947\,x^* + 3.9785$	0.5354	1.1080	22.57%
4.	$-0.5216\,x^{*3} + 3.2554\,x^{*2} - 5.1429\,x^* + 4.4065$	0.5695	1.1104	-0.21%

All this could measure actual fairness because it was not just defined as a concrete requirement in the scheduling process but as general information handled in addition to it. For example, in the case of final exams, it is not just the workload number alone that provides information. However, all the available information is aggregated to determine how fair a person's assignment is compared to the others. After that, the aggregation of all this information is used to measure the fairness of the schedule.

Thus in the future, if this algorithm is included along with the scheduling process, it will be possible to select the one from several different schedules that are the fairest to individuals.

ACKNOWLEDGEMENT

The work presented in this paper has been carried out in the frame of project no. 2019-1.1.1-PIACI-KFI-2019-00263, which has been implemented with the support provided by the National Research, Development, and Innovation Fund of Hungary.

REFERENCES

[1] *Fairness Meaning in the Cambridge English Dictionary*. Cambridge University Press, 2021. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/fairness. Accessed: Dec. 20, 2022.

[2] N. Grgic-Hlaca, E. Redmiles, K. P. Gummadi, and A. Weller, "Human perceptions of fairness in algorithmic decision making: A case study of criminal risk prediction," in *Proceedings of the World Wide Web Conference*, Lyon, France, April 23–27, 2018, pp. 903–912.

[3] P. Neelakantan and A. Reddy, "Decentralized load balancing in distributed systems," *Pollack Period.*, vol. 9, no. 2, pp. 15–28, 2014.

[4] M. Seyyed, M. Javadi, and K. Ali, "A combined approach for cloud tasks scheduling based on NSGA-II and harmony search," *Pollack Period.*, vol. 17, no. 3, pp. 1–6, 2022.

[5] K. Shen and S. Park, "FlashFQ: A fair queueing I/O scheduler for flash-based SSDS," in *2013 Annual Technical Conference*, San Joce, CA, June 26–28, 2013, pp. 67–78.

[6] J. Zhao, O. Mutlu and Y. Xie, "FIRM: Fair and high-performance memory control for persistent memory systems," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, Cambridge, UK, December 13–17, 2014, pp. 153–165.

[7] I. Budhiraja, N. Kumar, and S. Tyagi, "Deep-reinforcement-learning-based proportional fair scheduling control scheme for underlay D2D communication," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3143–3156, 2020.

[8] G. Hong, J. Martin, and J. Westall, "Adaptive bandwidth binning for bandwidth management," *Comput. Network.*, vol. 150, pp. 150–169, 2019.

[9] J. Pan, G. Chen, H. Wu, X. Peng and L. Xia, "Deep reinforcement learning-based dynamic bandwidth allocation in weighted fair queues of routers," in *2022 IEEE 18th International Conference on Automation Science and Engineering*, Mexico City, Mexico, August 20–24, 2022, pp. 1580–1587.

[10] K. Ikeda, Y. Nakamura, and T. S. Humble, "Application of quantum annealing to nurse scheduling problem," *Scientific Rep.*, vol. 9, no. 1, 2019, Art no. 12837.

[11] G. Woumans, L. De Boeck, J. Belien, and S. Creemers, "A column generation approach for solving the examination-timetabling problem," *Eur. J. Oper. Res.*, vol. 253, no. 1, pp. 178–194, 2016.

[12] R. Mansini and R. Zanotti, "Optimizing the physician scheduling problem in a large hospital ward," *J. Schedul.*, vol. 23, no. 3, pp. 337–361, 2020.

[13] A. Uhde, N. Schlicker, D. P. Wallach and M. Hassenzahl, "Fairness and decision-making in collaborative shift scheduling systems," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, April 25–30, 2020, pp. 1–13.

[14] R. Vetschera, P. Sarabando, and L. Dias, "Levels of incomplete information in group decision models–a comprehensive simulation study," *Comput. Operations Res.*, vol. 51, pp. 160–171, 2014.

[15] S. Jütte, D. Müller, and U. W. Thonemann, "Optimizing railway crew schedules with fairness preferences," *J. Schedul.*, vol. 20, pp. 43–55, 2017.

[16] T. Breugem, T. Schlechte, C. Schulz, and R. Borndörfer, "A three-phase heuristic for the fairness-oriented crew rostering problem," *Comput. Operations Res.*, vol. 154, 2023, Art no. 106186.

[17] *ACM Conference on Fairness, Accountability, and Transparency*, Seoul, South Korea, June 21–24, 2022.

[18] S. Verma and J. Rubin, "Fairness definitions explained," in *Proceedings of the International Workshop on Software Fairness*, Gothenburg, Sweden, May 29, 2018, pp. 1–7.

[19] C. Dwork, M. Hardt, T. Pitassi, O. Reingold and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, New York, NY, US, January 8–10, 2012, pp. 214–226.

[20] S. Erdős and B. Kővári, "Examination of fairness in scheduling tasks with heterogeneous resources," in *8th International Conference on Soft Computing & Machine Intelligence*, Cario, Egypt, November 26–27, 2021, pp. 155–159.

[21] N. C. Schwertman, M. A. Owens, and R. Adnan, "A simple more general boxplot method for identifying outliers," *Comput. Stat. Data Anal.*, vol. 47, no. 1, pp. 165–174, 2004.