2023 International Conference on Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS) | 979-8-3503-4053-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITQMTIS58985.2023.10346285

University Teaching Load Distribution Algorithm

Vladlena D. Markvirer
Information Technologies in Business Department
National Research University Higher School of Economics
Perm, Russia
vdmarkvirer@hse.ru

Ekaterina A. Karnaukhova
Information Technologies in Business Department
National Research University Higher School of Economics
Perm, Russia
eakarnaukhova@hse.ru

***Abstract*— This article presents an overview of the main algorithms used in resource distribution problems. The description, input and output data, as well as peculiarities and complexity of these algorithms are considered. The paper proposes a modified algorithm for finding vapor combinations in a graph, which allows solving the problem of optimal load distribution between the teachers of a structural subdivision of the university and the disciplines that this subdivision is obliged to implement in an academic year. The need for modification of the algorithm is argued by the unevenness and variability of the subject area, multi-criteria load assignment and the need to take into account subjective characteristics of the decision on allocation.**

Keywords—teaching load distribution, graph algorithms, graph algorithms application

I. Introduction

Distribution of teaching load in higher education institution is considered to be one of the fundamental elements of providing the educational process in higher education institution [1]. Due to the multicriteria selection and constraints for assigning teachers to the disciplines of the curriculum and practical training of students, the task of load distribution is non-trivial and requires new approaches to improve the efficiency of this process. The efficiency of the allocation process is characterized by [2]:

- uniformity in the distribution of hours of a structural unit under the curriculum for the year among the teachers of different levels of this unit;
- compliance with the regulatory limits on the implementation of the planned disciplines between the teaching staff of a structural unit;
- the time taken to develop individual plans for each faculty member's workload.

The accuracy, timeliness and evenness of the distributed workload is the key to improving (or at least maintaining) the quality of educational services, ensuring the stability and growth of scientific competencies of not only students, but also teachers [3]. In order to move from the outdated form of teaching load distribution, which demonstrates time losses for the process, uneven load distribution, leading to a decrease in the quality of education due to the minimization of time for rest and self-improvement of teachers, it is proposed to consider more efficient algorithms to automate the load distribution among teachers.

The purpose of the work is to describe an algorithm for the optimal distribution of teaching load, minimizing the uneven distribution, taking into account the constraints given by regulations and laws on the organization of the educational process.

In order to achieve the goal it is necessary to:

- conduct a study of existing algorithms used to minimize costs, finding the optimal allocation of resources;
- to formulate an algorithm for solving the problem of workload distribution in higher education institution.

II. Basic Algorithms on Graphs

Recently, graphs have been used to implement systems that represent work with data sets about objects of the real world. Graphs allow to describe and structure complex data sets interconnected with each other [4]. At the same time there are many efficient and accurate algorithms on graphs, the speed of which can be much higher than a simple enumeration of all possible variants and complex calculations with the correct organization of input data and given constraints.

Further in the section we consider the problems in which it is appropriate to use algorithms on graphs consisting of V vertices and E edges. The formulations of the problems, their interpretation and applied algorithms for their solution are given [5, 6].

A. Finding Vapor Combinations (Correspondences) Problem

The problem is also known as assigning performers to perform processes. To solve it a bipartite graph representation is used and in graph theory the number of vertices in each lobe is equal.

For the bipartite graph we use Hopcroft-Karp, flower compression and Hungarian algorithms finding maximal and perfect vapor combinations with speed $O(V^{0,5}\bullet E)$, $O(V^3)$ and $O(V^2\bullet E)$, respectively. The basic idea of the algorithms is to find an alternating chain or tree as long as it is possible and as long as there are free vertices. The result is a set of edges that are not adjacent to others (vapor combinations), which covers either all vertices of the graph (perfect vapor combination) or the maximum possible one (maximal vapor combination).

Algorithms are used in practice in logistics to solve the problem of resource allocation and optimization along the way, as well as assigning performers to jobs that can be performed independently of each other.

For the task of subdivision’s teaching load distribution steam-combinations are directly relevant, because the relationship of a pair of vertices in the form of assignment to some teacher to teach one of the disciplines of the curriculum is clearly traceable. However, it is worth noting that in contrast to the basic interpretation of pair combinations, the relations in the load sharing problem have a more complex structure, and pair combinations can represent a set of partial graph edges, because often one discipline can be taught by several teachers, and the load sharing between teachers can take place not only by types of discipline implementation, but also can be divided into the so-called parts of discipline material by the same type of implementation.

B. *Finding the Shortest Path Problem*

The problem is to find a path in a graph from one vertex to another with minimal weight of the included edges. It is used to construct routes between cities, in routing protocols to ensure minimal delay.

Dijkstra and Bellman-Ford algorithms are used to solve the problem, with average execution time complexity estimated to be $O(V^2)$ and $O(V \cdot E)$, respectively.

The main idea of the above algorithms is to find the shortest path from one of the graph vertices to a given one, gradually reducing the path to all neighboring graph vertices, excluding the visited ones. As a result of the algorithm the shortest path, the minimum distance or the cost of the path from the initial vertex to any other vertex becomes known.

In terms of the task of distributing faculty workload, the shortest path search problem can be applied to find an option for distributing faculty workload hours that ensures that the minimum faculty rate threshold is passed and that the workload is distributed evenly among faculty members.

C. *Minimum Cost Maximum Flow Problem*

This class of problems is also known as the transport or resource allocation problem, which requires constructing a network with such an edge load and cost that it provides maximum or complete resource allocation from source to sink, minimizing the cost of travel.

The Ford-Falkerson, Edmonds-Karp, and Dinitz algorithms are used to solve the problem. These algorithms allow us to find the maximum flow in the transportation network. The maximum flow is the maximum possible loading of the network edges, found as a result of sequential loading of flows by the minimum value from the increasing path of the residual network (the path from the sink to the source).

The Ford-Falkerson algorithm can only complete its work for integer values of threads in time equal to $O(V \cdot E)$. The Edmonds-Karp and Dinitz algorithms allow to work with real values on streams, and also involve finding not any path but the shortest one. The complexity of the latter two algorithms is $O(V \cdot E^2)$ and $O(V^2 \cdot E)$, respectively.

The maximum flow problem, as applied to this paper, is directly relevant: workload allocation is to ensure that all hours are fully allocated to all faculty members.

III. Teaching Load Distribution Algorithm in an Educational Institution

The algorithms described above in theory have the simplest structure: the vertices denote the presence of some object in the problem, and the relations often show only the presence of a relation between vertices (sometimes the importance of the direction of the relation is marked with an arrow), and in some cases may have a weight or value of the relation. However, it is worth noting that real-world objects and the relationships between them contain varying amounts of additional attributes that affect decision making, such as the allocation of faculty workload to curriculum disciplines.

A. *Teaching Load Distribution Input Data*

The teaching load distribution problem has many input parameters, constraints, and rules. To begin to perform load balancing specialists need information about the faculty members in the department: the number of full-time and visiting faculty members, as well as the positions held, degrees and titles held. The next necessary information is the total number of hours of teaching load for the coming academic year, formed by the approved curriculum and student contingent and necessary for implementation. On the curriculum the list of disciplines and load units on practical training of the unit is also formed, at each discipline the hours of contact work with students on carrying out lectures, seminars, practical classes, control elements and examination, and on the element of practical training the normative load, number of hours per unit of normative load and total number of students is given.

The purpose of load distribution is an equal distribution of the total number of allocated hours of teaching load per unit among the teaching staff. In doing so, it is necessary to keep in mind and take into account the regulations on the faculty member's normative load hours by position, competencies, faculty wishes and preferences, and other additional information.

The following section presents a graphical representation of the teaching load allocation problem, provides designations and explanations, and specifies the basic formulas.

B. *Teaching Load Distribution Algorithm Graph Representation*

Figure 1 shows a graph describing the teaching load allocation problem.

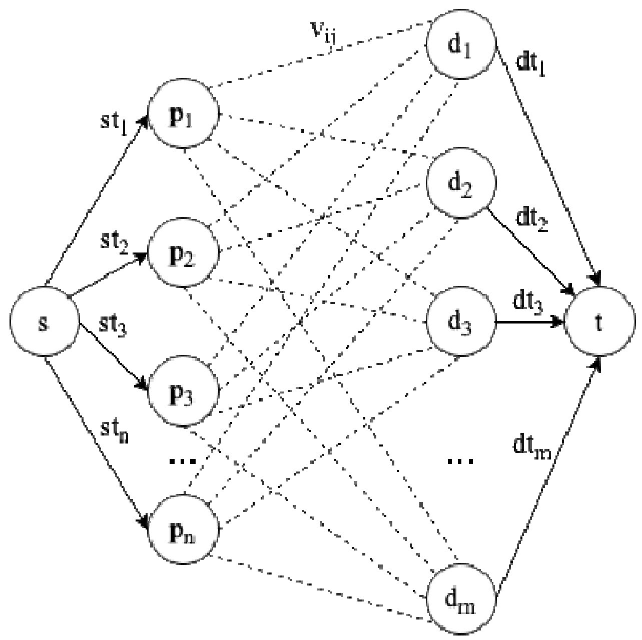

Fig. 1. General view of the load distribution graph between the teachers on the disciplines of the curriculum

The graph has two types of vertices: "s" and "t" – the vertices that characterize the number of hours of teaching load needed to be allocated and the allocated hours, respectively, and the vertices p_n and d_m that represent teachers and courses on the curriculum, respectively. Also in the graph the vertices are connected by relations $st_{1..n}$ – hours for $p_{1..n}$, $dt_{1..m}$ – hours for $d_{1..m}$, v_{ij} – possible relations connecting $p_{1..n}$ and $d_{1..m}$ (where i and j are order numbers of teachers and disciplines from 1 to n and m respectively), the relations are formed based on regulations, competences and preferences.

Nodes p can contain such attributes as teacher's full name, position (lecturer, senior lecturer, associate professor, professor, visiting lecturer), academic degree, academic rank, lowering rate, number of rates between (0;1], qualification. preferences for teaching certain disciplines.

Vertices d can contain such attributes as the name, type of discipline load (classroom or additional for practical training). If the discipline object refers to the classroom load, then this object will be additionally characterized by the number of hours for lectures (hl), seminars (hs), practicals (hp), the number of streams (kp), groups (kg) and subgroups (ku). If the object of the discipline refers to the additional load, then – the number of units of students or groups of students (count), the standard load (k), the number of hours per standard load (h), as well as the type of additional work (course work, course project, study, industrial or pregraduate practice, graduate qualification work).

The position of the teacher, the availability of lecture hours classroom load and the type of additional load depends on what disciplines, in what format the teacher can be assigned (this is reflected in the presence or absence of links v_{ij}).

Rate hours ($st_{1..n}$) are determined by the sum of the allotted load per instructor ($p_{1..n}$), taking into account the limitations, qualifications and preferences of the instructor. At that $st_{1..n}$ cannot exceed the number of rates for $p_{1..n}$ taking into account the position and possible load reduction index. Vertex s has a weight equal to the sum of all sti, where i is the teacher's ordinal number from 1 to n.

The total number of hours per discipline is formed according to formula (1), where j is the ordinal number of the discipline from 1 to m.

$$d_j = hl_j * kp_j + hs_j * kg_j + hp_j * ku_j + count * h_j * k_j \quad (1)$$

Load hours for the discipline ($dt_{1..m}$) is determined by the sum of the allocated hours for the discipline ($d_{1..m}$). When a discipline's hours are fully allocated, $dt_{1..m}$ must equal $d_{1..m}$. Otherwise, the unit needs to take steps to update the staff or to find options for allocating other faculty members in the unit. The vertex "t" has a weight equal to the sum of all dt_j, where j is the ordinal number of the discipline from 1 to m.

Ideally, the hours in vertex "s" and "t" should be equal. Otherwise, management decisions must be made to execute the unit's curriculum.

C. Teaching Load Distribution Algorithm Description

Taking into account the constructed graph for the problem about the distribution of teachers to disciplines, and the specified notations, the algorithm for the distribution of teaching load is constructed (Fig. 2).

The algorithm takes as input already prepared vertices – objects describing teachers and disciplines from the curriculum. Then it is necessary to calculate the boundaries of the allowable teaching load: the minimum and maximum values are formed on the basis of the position held, the number of rates occupied in the unit, the possible reduction of the load, as well as regulated by the established standards for the organization of work of teaching staff at the educational institution and the higher standards.

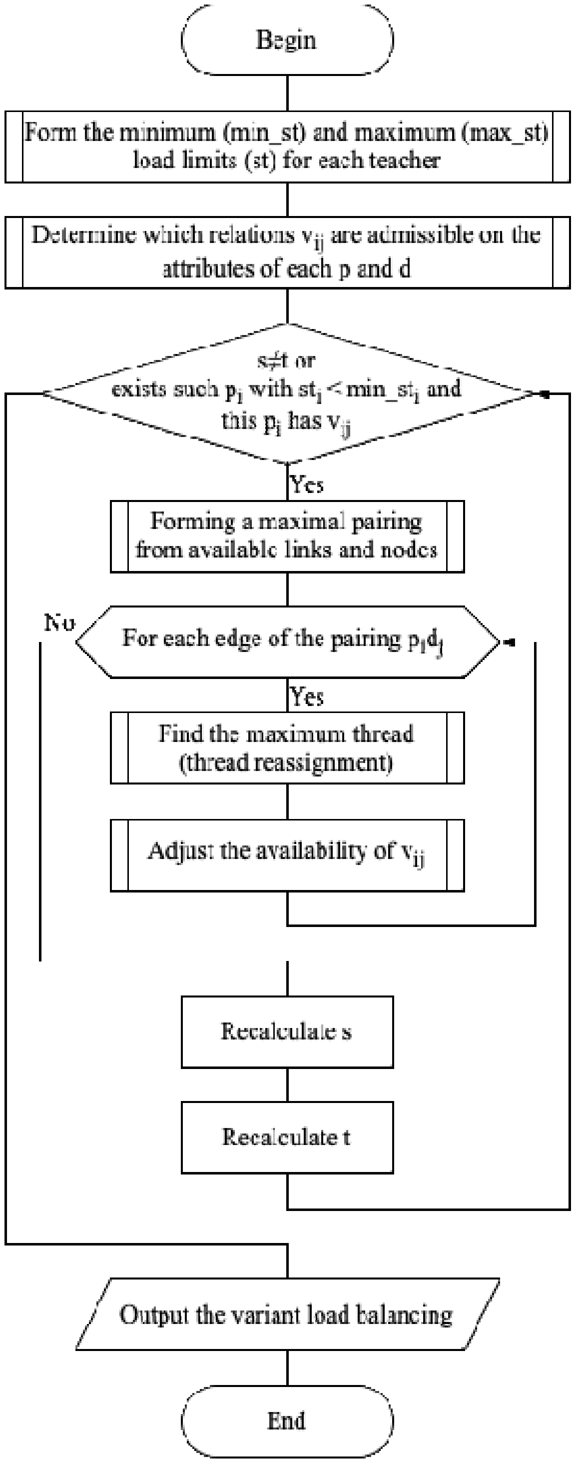

Fig. 2. Algorithm of workload distribution among teachers by curriculum disciplines

The next step of the algorithm is to determine the permissibility of links between teachers and disciplines. The link is acceptable if the position of the teacher, academic degree and rank allow to engage in the implementation of a particular type of discipline in a certain form. It is worth noting that the links must support such properties as the level of competence and the degree of willingness of the teacher to engage in the implementation of a particular discipline, as well as the number of hours assigned to the teacher.

Further, the algorithm involves cyclic execution of load assignment to teachers with intermediate finding of the maximum pairings, adjusting the availability of using links in the following iterations and recalculating the sum of allocated hours by teacher rates and allocated hours of disciplines. The cycle is completed if all the hours on the curriculum are distributed among the teachers or if there is no teacher who has not passed the threshold of the minimum allowable number of hours in the load and has no remaining possible links with the disciplines of the curriculum. The first case of cycle completion characterizes ideal conditions. The second one requires making additional managerial decisions on expanding the staff in the unit by vacant places for certain disciplines, or organizing a meeting with more unloaded employees of the unit, to find out the possibility of expanding their preferences and competencies.

To reduce the load on the computing resources of the computer in the execution of the algorithm, hence reducing its complexity, it may be worth thinking about the gradual elimination of the vertices and edges of the graph at each iteration of the algorithm, in case all hours for the discipline will be allocated in the previous steps, or if the teacher will have reached the maximum number of allowable load hours, for example.

IV. Conclusion

The computational power of modern computers helps to automate routine operations performed by people in everyday life. Graphs, in their turn, allow to reproduce the structure and meaning of interrelations of real-world objects in a computer representation, without the need to create artificial data structures. At the same time, there are many efficient and fast algorithms on graphs for solving problems similar to resource allocation. A comparative review of some algorithms suitable for use in solving the problem of teacher workload allocation in higher education institutions is presented in the article.

In this paper, an algorithm for the distribution of teaching load hours among faculty members in a unit, accompanied by a graphical representation similar to a transport network, is proposed.

The distribution algorithm is planned for further development as part of the implementation of a decision support system for the teaching load distribution in the department, so changes and additions to the described key aspects are possible when inaccuracies or collisions are detected.

References

[1] Kalyugniy N. V. *Analiz processa raspredeleniya uchebnoj nagruzki professorsko-prepodavatel`skogo sostava na kafedrax* [Analysis of the distribution of teaching load of the teaching staff in the departments] / N. V. Kalyugniy . – Text: digital // *Science Time*. – 2015. – No. 6 (18). – P. 199-202. – Mode of access: open. – URL: https://cyberleninka.ru/article/n/analiz-protsessa-raspredeleniya-uchebnoy-nagruzki-professorsko-prepodavatelskogo-sostava-na-kafedrah (accessed: 14.06.2023), in Russian.

[2] Markvirer V. D. *Trebovaniya k sisteme podderzhki prinyatiya reshenij o raspredelenii prepodavatel`skoj nagruzki* [Teaching Distribution Load Decision Support System Requirements] / V. D. Markvirer, E. A. Karnaukhova // *Nauka nastoyashhego i budushhego*. – St. Petersburg, 2023. – 5 pp., in press.

[3] Yashin A. A. *Normirovanie i raspredelenie uchebnoj nagruzki: vzglyad praktika* [Work quota setting of academic load: practical review] / M. N. Strukova, A. A. Yashin . – Text: digital // *Universitetskoe upravlenie: praktika i analiz*. – 2015. – No. 6 (100). – P. 100-108. – Mode of access: open. – URL: https://cyberleninka.ru/article/n/normirovanie-i-raspredelenie-uchebnoy-nagruzki-vzglyad-praktika (accessed: 14.06.2023), in Russian.

[4] Kasyanov V. N. *Instrumenty` podderzhki primeneniya grafov i grammovy`x algoritmov* [Support tools for application of graphs and graph algorithms] / V. N. Kasyanov, S. N. Kasyanova // *Materialy` konferencii MIT*. – Belgrade, 2013. – P. 322–328. – Mode of access: open. – URL: http://conf.nsc.ru/files/conferences/MIT-2013/224768/zbornik-2013.pdf (accessed: 14.06.2023), in Russian.

[5] Markvirer V. D. *Prikladnoe primenenie teorii parosochetanij v grafax* [Applications of Vaporization Theory in Graphs] / V. D. Markvirer // *Sosedi po nauke*. – Perm, 2023. – P. 38-50. – Mode of access: open. – URL: https://perm.hse.ru/editorial_publishing/neighbors_in_research2 (accessed: 14.06.2023), in Russian.

[6] *10 animirovanny`x algoritmov na grafax* [10 animated algorithms on graphs]. – Mode of access: open. – URL: https://proglib-io.turbopages.org/proglib.io/s/p/10-animirovannyh-algoritmov-na-grafah-2020-09-09 (accessed: 14.06.2023), in Russian.