



# Towards an Optimal Placement of Learning Resources in a Fog Computing Based E-Learning System: The Case of UVS

Serigne Mbacke Gueye<sup>1(✉)</sup>, Alassane Diop<sup>2</sup>, and Amadou Dahirou Gueye<sup>1</sup>

<sup>1</sup> University Alioune Diop de Bambey (UADB), Diourbel, Senegal  
{serignembacke.gueye,dahirou.gueye}@uadb.edu.sn

<sup>2</sup> Université Virtuelle du Senegal (UVS), Diamniadio, Senegal  
alassane.diop@uvs.edu.sn

**Abstract.** With the growing number of its students, Virtual University of Senegal is often faced with a lack of quality of service during synchronous learning activities. Thus, the University has set itself the challenge of reinforcing its technological device to make it even more robust in order to support the increasing number of students with a more powerful infrastructure. In this perspective, we have, in this paper, proposed an online teaching platform based on fog computing. Thus, we have proposed an optimal placement of educational resources at the fog layer nodes in order to relieve the backbone network and, consequently, reduce the response time of students' requests during synchronous activities, which are very sensitive to delay.

**Keywords:** Distance learning · Fog computing · Optimization · Placement of learning resources in fog nodes

## 1 Introduction

In recent years, many universities and training institutes have taken advantage of ICT (Information and Communication Technology) to improve their educational strategies in order to attract the maximum number of students [1]. In this perspective, e-learning is widely adopted to offer a quality and more flexible educational service. Accordingly, this form of learning is getting more and more popular worldwide and the number of learners in these online courses increasing exponentially. In this mode of education, all the educational resources are stored in the cloud to allow the various learners to access via the internet.

UVS, with more than 50,000 students in 15 Open Digital Spaces, has adopted this teaching model in order to offer a wide range of training to its students [2]. At this university, two types of learning activities are carried out: Synchronous activities, where learners and the teacher (or tutor) interact live via a dedicated platform (blackboard collaborate, bigbluebutton, Google Meet, etc.) and asynchronous activities i.e. that those students can do at any time. Thus, with

this large number of students who are supposed to connect at the same time to the university's cloud servers with their respective teacher (or tutor) in a synchronous session, the university is often confronted with problems of computing resources of its cloud servers (I/O, RAM, CPU, etc.). In this respect, the quality of service (QoS) during synchronous activities remains poor.

To address these kinds of QoS issues, particularly response time, fog computing has emerged. It is a recent paradigm that aims to extend cloud computing to the network edge to address the challenges associated with the exponential growth of connected objects at the network edge. In fact, fog computing relies on network resources available at the edge of the network, known as fog nodes, to provide services with reduced response times. These devices, although characterized by limited computing and storage capacity, benefit from a proximity to the end-users which allows the location of learners, saves backbone bandwidth and reduces transmission delays [3].

In this work, we study in the context of the UVS pedagogical model, the implementation of a distance learning platform deployed on geo distributed computing in the form of an optimization problem.

In this context, we propose a heuristic, based on the Greedy algorithm, capable of proposing a better placement of educational resources, at the level of fog nodes, in the form of containerized applications (for requests concerning synchronous learning activities) in order to minimize the response time (transmission delay).

Thus, we will :

- Define a relevant use case, in line with the UVS pedagogical model, for a distance learning platform based on fog computing;
- Give a mathematical formulation of the latency problem of our e-learning platform based on fog computing by an optimization problem whose objective is to maximize the number of containerized learning resources to be deployed to satisfy the requirements of delay of the requests of the learners, under several constraints which we will detail in Sect. 4;
- Proving such a problem is NP-hard;
- Propose a heuristic based on the greedy algorithm.

The rest of the paper will be organized as follows: In Sect. 2, we will review the literature about fog computing based e-learning platforms. Section 3 will be devoted to the study of our proposed platform, in which we describe its components and their functioning. In the last section, we will describe the model system, the mathematical formulation of the problem, and finally propose a heuristic for solving the problem.

## 2 Related Work

In this section, we will review the literature in order to identify all the existing works with the aim of detecting their limits in relation to our field of study.

In recent years, being aware of the contribution that fog computing can have on e-learning, many researchers have become interested in this new field of research.

In [4], the authors presented a model based on fog computing for accessing educational content in a virtual learning environment. It presents an optimized use of bandwidth that significantly reduces latency, thus improving the quality of teaching/learning. Their results show that fog computing could make virtual learning environments more relevant to the needs of students and teachers.

The authors of [5] proposed an e-learning environment architecture enhanced by fog computing and big data streams. They discussed virtual learning environments and their limitations and then explained the paradigms of big data streams and fog computing to introduce an e-learning architecture integrating both to enhance the ability of virtual learning environments to approximate the needs of all stakeholders.

In [6], the authors have proposed a new secure e-learning scheme based on fog computing. Their solution extends educational content from the cloud to the edge of the network, closer to the learners and teachers. They provided precise access control to learning content by encrypting courses and exams using IBBE and CP-ABE techniques.

In [7], the authors present an IoT-Fog-Cloud framework to provide security factors in sharing E-exam which poses several security challenges, such as fine-grained access control and security preservation of E-exam. Further, there proposed framework supports bringing closer the services to the students. Besides, there paper improves the efficiency of E-exam data analysis, reduces the encryption burden in terms of computation cost on user's devices by offloading part of encryption cost to fog servers, and provides fine-grained access control to E-exam content by encrypting with different cryptographic techniques.

A recommendation system based on fog computing to improve e-learning environments is proposed in [8]. Their system is based on three modules: A class identification module that calculates the desired course category based on the user's query keyword by calculating the correlation between the queries and all domain classes. A subclass identification module that calculates the subclass of the desired subject by applying the association rule technique. A Matching module to retrieve the selected items (classes) and classify them according to their relevance to the user's query.

In [9], a fog-assisted cyber physical system is proposed that deals with the various aspects of the panic well-being of the student, including the virtual reality platform for remote learning. The proposed system utilizes the concepts of physical and cyberspace. The physical space facilitates real-time data acquisition, and cyberspace determines and predicts the panic well-being of the student. The performance assessment of the proposed model acknowledges the efficiency of the virtual learning system and panic well-being determination and prediction. The proposed system also discussed a virtual learning system that provides a virtual classroom environment to the students at remote sites and reduces the panic due to stressful times during the COVID-19 pandemic.

Through all these works, it can be noted that fog computing is widely used in the field of e-learning with a view to improving teaching/learning, even going as far as securing learning objects. However, to our knowledge, no proposal has

been made concerning the optimal placement of learning objects in the form of containers at the level of fog nodes for better support of synchronous activities in e-learning systems.

Thus, in the following part, we will, according to the pedagogical model of the UVS, try to propose an online teaching platform based on fog computing (PELBFC - UVS). We will also propose an optimal placement of pedagogical resources in the form of containerized applications according to an optimization problem. Finally, we will propose a heuristic for solving this problem.

### 3 PELBFC - UVS: Components and Functioning

The proposed system, illustrated in Fig. 1, is based on geo distributed computing (*fog computing*) and works according to the collaboration of three hierarchical layers: the cloud layer, the fog layer and the user layer (Learners, Teachers, Tutors, Administrative staff) according to the following architecture.

The user layer is mainly made up of all the students spread over the 15 Open Digital Spaces of the University across the country. Teachers, tutors and pedagogical administrative staff are also part of the users. In our scenario, we consider learners who wish to access a set of synchronous learning activities via requests sent to the orchestration module of the fog layer. This orchestration module communicates with all the fog controllers in order to determine the most appropriate fog node (the fog node offering the best quality of service) for the execution of the learner's request.

The fog layer, on the other hand, is made up of intelligent devices (routers, switches, gateways, access points, proxy servers, etc.) better known as *fog node*. A fog node may be mobile or static (fixed) and capable of processing, routing or storing data. They are usually grouped into domains which are placed under the authority of a fog controller responsible for assigning Learner's requests to the most appropriate fog nodes according to the required delay of the request, the resources available at the nodes and the position of the learner. However, it should be noted that the computational and storage resources of the fog nodes are limited compared to those of the cloud.

Thus, resource-intensive requests are offloaded to the cloud layer to take advantage of the large computer and storage capacities of its data centers with a higher delay. The orchestration module coordinates services between the fog controllers and the cloud layer data centers.

### 4 Mathematical Formulation

In this section, we will give a mathematical formulation of our e-learning platform based on fog computing with consideration of delay requirements for learner requests for synchronous learning activities.

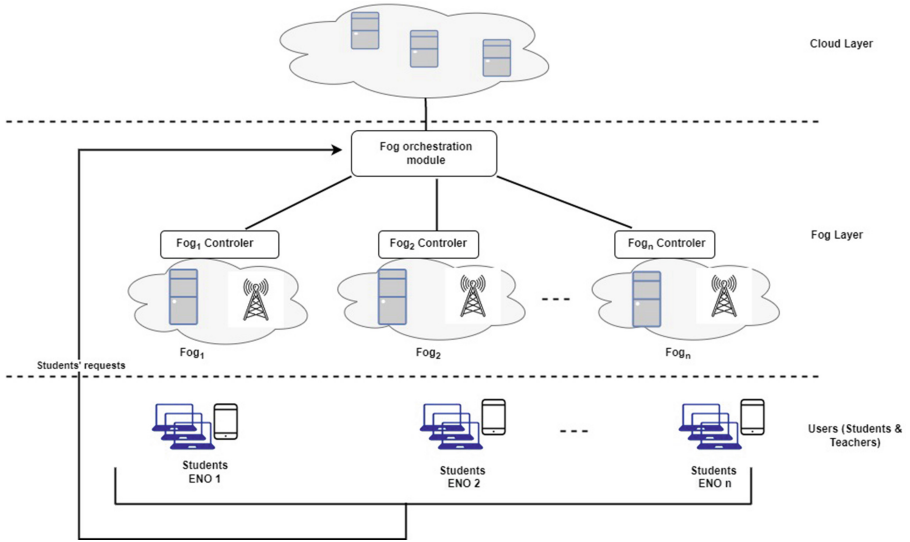


Fig. 1. Architecture PELBFC-UVS

### 4.1 Problem Description (Use Case)

Consider a set of synchronous learning activity teaching resources to be deployed on the fog nodes in order to allow students access in a reasonable time frame (real time).

A fog controller will be responsible for deploying a resource allocation strategy to maximize the number of learner requests that meet the delay requirements by taking into account the location of the learners, the fog nodes and their available resource capacity.

After setting up this allocation strategy, the learning resources, in the form of containerized applications, are deployed on the *fog nodes* with support for the delay requirements.

For reasons of resource optimization, a container will be able to run multiple compatible learning resources from multiple learners.

### 4.2 Variables Definition

Let  $\mathcal{K}$  be a set of fog domains. Each domain is managed by a fog controller  $k$ .  $\mathcal{F}_k$  represents a heterogeneous set of fog nodes in a given domain managed by a  $k$  controller. It is assumed that each fog node  $f \in \mathcal{F}_k$  integrates a containerization platform for better management of the containers assigned to it. Let  $\mathcal{F}$  be a set of *fog nodes*. Let us consider the following vectors :

The vector  $\mathbf{l} = [l^f]$  expresses the locations of the *fog nodes* and  $\mathbf{c} = [c^f]$  denotes their remaining carrying capacity. As the load supported by the dedicated functions varies over time, the amount of resources available on the fog

computing platform fluctuates as result from [10]. Thus, we express by the vector  $\mathbf{z} = [z^f]$ , the use of the *fog nodes*, proportional to the quantity of resources dedicated to the primary functions.  $\mathbf{b} = [b^f]$  is a binary vector representing whether or not a *fog node* is available to host containerized applications.

We define by  $\mathcal{A}$ , a set of learners in the system with  $\mathbf{p} = [p^a]_{|\mathcal{A}|}$ , the vector representing the location of learners. We consider  $\mathcal{Y}$  to be a set of synchronous pedagogical resource types in the form of a containerized application that are available in the University pedagogical model.

Let  $\mathbf{Q} = [Q^{a,y}]_{|\mathcal{A}| \times |\mathcal{Y}|}$  be the matrix defining the task load of the applications and  $\mathbf{T} = [T^{a,y}]_{|\mathcal{A}| \times |\mathcal{Y}|}$  the matrix defining the threshold delay of the applications.

Let the binary matrix  $\mathbf{W} = [w^{a,y,f}]_{|\mathcal{A}| \times |\mathcal{Y}| \times |\mathcal{F}|}$  indicate whether or not a pedagogical resource of type  $y$ , requested by a learner  $a$ , is placed on the *fog node*  $f$ . The table below gives a summary of all these variables we have just defined (Table 1).

### 4.3 Definition of the Delay

In the implementation of our fog computing based e-learning platform, we will consider the delay (response time) as a QoS parameter. In the following lines, we will give its expression that we have considered in the realization of this work.

**Table 1.** Tables of variables

|                 |   |
|-----------------|---|
| $\mathcal{K}$   | Set of fog domains  |
| $\mathcal{F}$   | Set of fog nodes  |
| $\mathcal{F}_k$ | Set of <i>fog nodes</i> in fog domaine $k \in \mathcal{K}$  |
| $\mathcal{A}$   | Set of Learner  |
| $\mathcal{Y}$   | All types of learning resources   |
| $\mathbf{l}$    | $ \mathcal{F} $ location vector of the fog nodes  |
| $\mathbf{c}$    | $ \mathcal{F} $ vector representing the remaining load capacity of the fog nodes  |
| $\mathbf{z}$    | $ \mathcal{F} $ Vector representing the use of <i>fog nodes</i>   |
| $\mathbf{b}$    | $ \mathcal{F} $ Vector where the element $b^f$ is a binary vector representing if $f \in \mathcal{F}$ is free to host containers or not   |
| $\mathbf{p}$    | $ \mathcal{A} $ Learner location vector   |
| $\mathbf{Q}$    | $ \mathcal{A}  \times  \mathcal{Y} $ matrix representing the load of the containerized applications   |
| $\mathbf{T}$    | $ \mathcal{A}  \times  \mathcal{Y} $ matrix indicating applications threshold latency requirements  |
| $\mathbf{W}$    | $ \mathcal{A}  \times  \mathcal{Y}  \times  \mathcal{F} $ binary matrix representing whether a learning resource of type $y$ , requested by learner $a$ , is deployed on <i>fog node</i> $f$ or not |

The delay  $d^{a,y}$  of the tasks of an application  $y$  for a learner  $a$  is composed of three delays: (1) the communication delay (round trip) between the fog node and the learner, (2) the task processing delay, and (3) the waiting delay, which is included in the processing delay, in this work. Thus, the latency of an application can be written as follows :

$$d^{a,y} = d_{cmp}^{a,y} + d_{com}^{a,y} \quad \forall a \in \mathcal{A}, y \in \mathcal{Y} \quad (1)$$

where  $d_{cmp}^{a,y}$  and  $d_{com}^{a,y}$  represent the processing time and the communication time respectively.

Since a learning resource can be available on several fog nodes, the processing time  $d_{cmp}^{a,y}$  of a learner's tasks  $a$  for an application  $y$  is equal to the minimum of the fog nodes' processing times  $d_{cmp}^{f,a,y}$ . Furthermore, if a node is not available, we assume that its processing time will take a very large value such that :

$$d_{cmp}^{a,y} = \min_{f \in \mathcal{F}} (d_{cmp}^{f,a,y} \cdot w^{a,y,f}), \quad \forall a \in \mathcal{A}, y \in \mathcal{Y} \quad \text{and} \quad w^{a,y,f} = 1. \quad (2)$$

For simplicity purpose, and to make our platform compatible with other threading models, we have derived  $d_{cmp}^{a,y}$  using an M/M/1 threading model. Thus,  $d_{cmp}^{f,a,y}$  is represented as follows :

$$d_{cmp}^{f,a,y} = \frac{1}{\rho^f - Q^{a,y,f}} \quad (3)$$

where  $\rho^f$  is the service rate of a node  $f$ .

In the same way as for the processing delay, the communication delay associated with a learner  $a$  for an application  $y$  is the minimum of the communication delays of the fog nodes which hosted this resource  $y$ . The communication delay is given by the following expression :

$$d_{com}^{a,y} = \min_{f \in \mathcal{F}} (d_{com}^{f,a,y} \cdot w^{a,y,f}) \quad \forall f \in \mathcal{F}, a \in \mathcal{A}, y \in \mathcal{Y} \quad \text{and} \quad w^{a,y,f} = 1. \quad (4)$$

The communication delay  $d_{com}^{f,a,y}$  of a resource  $y$ , hosted in the fog node  $f$  and requested by the learner  $a$  is broken down as follows :

$$d_{com}^{f,a,y} = d_{com,a,f}^{f,a,y} + d_{com,f,a}^{f,a,y} \quad (5)$$

where  $d_{com,a,f}^{f,a,y}$  represents the transmission delay of the wireless communication from learner  $a$  to node  $f$  and  $d_{com,f,a}^{f,a,y}$  vice versa. It is expressed as follows :

$$\frac{Q^{a,y,f}}{B \cdot \log_2(1 + SNR)} \quad (6)$$

where  $B$  is the bandwidth of the transmission medium,  $SNR$  is the signal-to-noise ratio and  $Q^{a,y,f}$  is the load on the resource  $y$ .

Following the model, our objective is to determine the optimal placement of the containerized learning resources. We seek to maximize the number of satisfied learner's requests within the threshold time required for the learning resource under the constraints of the fog nodes capacity.

### 4.4 Problem Formulation

In this section, we will give the problem formulation of placing containerized learning resources in a fog computing infrastructure. We will start by describing the constraints that the system must satisfy before completing the objective function of the problem which will be the maximization of the number of containerized learning resources to be deployed that satisfies the required delay.

**Constraints Definition:** We start by defining the constraint to satisfy a threshold time for learners’ requests.

$$T^{a,y} - d^{a,y} \geq G \cdot (S^{a,y} - 1), \quad \forall a \in \mathcal{A}, y \in \mathcal{Y} \tag{7}$$

where  $G$  is a large positive number chosen arbitrarily and  $S^{a,y}$  is a binary variable indicating whether the delay  $d^{a,y}$  is less than the threshold delay  $T^{a,y}$  of the application.

The following constraint will ensure that the total load allocated to each fog node does not exceed its capacity ( $C^f$ ). It is represented as follows :

$$C^f \geq \sum_{a \in \mathcal{A}} \sum_{y \in \mathcal{Y}} Q^{a,y,f}, \quad \forall f \in \mathcal{F}. \tag{8}$$

**Objective Function:** As the latency of learners’ requests can vary depending on the location of the fog nodes and the learners, the objective of our problem is to maximize the number of deployed applications satisfying the specified delay requirements.

$$\begin{aligned} & \max_{W,Q} \sum_{a \in \mathcal{A}} \sum_{y \in \mathcal{Y}} (S^{a,y}) & \tag{P1} \\ \text{s.t.} & \quad (7) - (8). \end{aligned}$$

The problem (P1) presented above is NP-hard. Indeed, it consists in placing containerized applications on fog nodes with the objective of maximizing the number of applications to be deployed that satisfy the required deadline. Thus, the problem can be trivially modelled as the GAP (Generalized Assignment Problem) which is a well known problem in the literature [11].

The objective of the GAP problem is to find a mapping between  $m$  agents and  $n$  heterogeneous tasks in order to maximize the total profit. Each task is assigned, exactly, to one agent, given that it has sufficient capacity. Moreover, depending on the assigned agent, each task may have a different profit. By analogy to our problem, agents can be likened to fog nodes, containerized applications as tasks and the number of requests answered with latency satisfaction are profit. Given that it is well known that GAP is NP-hard, according to the previous explanation, our problem (P1) is also NP-hard.

#### 4.5 Strategy Greedy First Fit

According to [12], an NP-hard problem is synonymous with the impossibility of finding an optimal solution in reasonable time. Thus, we propose this heuristic based on Greedy First Fit in order to have a solution close to the optimal one in polynomial time.

The Greedy First Fit is an algorithmic strategy which, at each stage of solving a problem, considers the first best choice which satisfies the constraints as being the best optimal choice with the aim of finally leading to an optimal global solution, without taking into account the consequences that this choice may have on the global solution of the problem [13].

Our strategy for solving (P1) is to deploy, as much as possible, the containerized learning resources on fog nodes that satisfy the delay requirements of student requests during synchronous learning activities, taking into account the 7–8 constraints. The details are described in the algorithm below.

---

#### Algorithm 1: Strategy Greedy First Fit

---

**Input:**

- $\mathbf{F}$  : Set of fog nodes,
- $\mathbf{L}_{|\mathcal{F}|}$  : Location vector of the fog nodes,
- $\mathbf{C}_{|\mathcal{F}|}$  : vector representing the remaining load capacity of the fog nodes,
- $\mathbf{P}_{|\mathcal{A}|}$  : Learner location vector,
- $\mathcal{Y}$  : All types of learning resources,
- $\mathbf{Q}_{|\mathcal{A}| \times |\mathcal{Y}|}$  : Matrix representing the load of the containerized applications,
- $\mathbf{T}_{|\mathcal{A}| \times |\mathcal{Y}|}$  : Matrix indicating applications threshold latency requirements

**Output:**

- Deployment scheme :  $\mathbf{W}_{|\mathcal{A}| \times |\mathcal{Y}| \times |\mathcal{F}|}$
- 1 *Random initialization of the matrix  $\mathbf{W}_{|\mathcal{A}| \times |\mathcal{Y}| \times |\mathcal{F}|}$ ;*
  - 2 **for** each learning resource to be deployed : **do**
  - 3     Randomly select a fog node  $f_{al}$  with respect to the constraints 7 and 8;
  - 4     Updating the matrix  $\mathbf{W}_{|\mathcal{A}| \times |\mathcal{Y}| \times |\mathcal{F}|}$ ;
  - 5     **for** each  $f_j \in F \setminus f_{al}$  **do**
  - 6         **if**  $f_j$  is better than  $f_{al}$  **then**
  - 7             Replace  $f_{al}$  by  $f_j$  in  $\mathbf{W}_{|\mathcal{A}| \times |\mathcal{Y}| \times |\mathcal{F}|}$  ;
  - 8             break ;
  - 9         **end**
  - 10     **end**
  - 11     Update of  $\mathbf{C}_{|\mathcal{F}|}$ ;
  - 12 **end**
  - 13 **return**  $\mathbf{W}_{|\mathcal{A}| \times |\mathcal{Y}| \times |\mathcal{F}|}$
-

## 5 Conclusion

To address the deterioration of the quality of service/quality of experience of teaching/learning during synchronous activities, caused by the increase in the number of students, we proposed an online teaching/learning platform based on fog computing. We have subsequently optimized the placement of educational resources, in the form of containerized applications, at the fog layer nodes in order to relieve the bandwidth of the backbone network and reduce the response time of student requests during synchronous activities, which are very sensitive to the delay.

In perspective, we intend to improve our heuristic by reinforcement learning for the solution of our problem in order to have a better solution in polynomial time.

## References

1. Rana Lone, C.P., Ahmad, Z., Ajay.: Architecture du système d'apprentissage en ligne pour le cloud computing - un examen
2. Gueye, S.M., Diop, A., Gueye, A.D.: A machine learning model on virtual university of senegal's educational data based on lambda architecture. In: 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI), pp. 270–275. IEEE (2020)
3. OpenFog Consortium Architecture Working Group et al.: Openfog reference architecture for fog computing. OPFRA001 **20817** 162 (2017)
4. Pecori, R.: Augmenting quality of experience in distance learning using fog computing. *IEEE Internet Comput.* **23**(5), 49–58 (2019)
5. Pecori, R.: A virtual learning architecture enhanced by fog computing and big data streams. *Future Internet* **10**(1), 4 (2018)
6. Amor, A.B., Abid, M., Meddeb, A.: Secure fog-based e-learning scheme. *IEEE Access* **8**, 31920–31933 (2020)
7. Khairy, D., Amasha, M.A., Abougalala, R.A., Alkhalaf, S., Areed, M.F.: A security system for e-exams using an IoT and fog computing environment. *J. Theor. Appl. Inf. Technol.* **100**(2), (2022)
8. Ibrahim, T.S., Saleh, A.I., Elgaml, N., Abdelsalam, M.M.: A fog based recommendation system for promoting the performance of e-learning environments. *Comput. Electr. Eng.* **87**, 106791 (2020)
9. Sood, S.K., Rawat, K.S.: Fog-assisted virtual reality-based learning framework to control panic. *Expert Syst.* **39**(4), e12700 (2022)
10. Mseddi, A., Jaafar, W., Elbiaze, H., Ajib, W.: Joint container placement and task provisioning in dynamic fog computing. *IEEE Internet Things J.* **PP**(99), 1 (2019)
11. Ross, S., Terry, G., Richard, M.: A branch and bound algorithm for the generalized assignment problem. *Mathe. Program.* **8**(1), 91–103 (1975)
12. Hochba, D.S.: Approximation algorithms for NP-hard problems. *ACM Sigact News* **28**(2), 40–52 (1997)
13. Cormen, T.H, Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT press, Cambridge (2009)