Decision Science Letters 12 (2023) 225–238

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

A convolutional neural network for the resource-constrained project scheduling problem (RCPSP): A new approach

Amir Golab[a*], Ehsan Sedgh Gooya[a], Ayman Al Falou[a] and Mikael Cabon[a]

[a]*ISEN Yncréa Ouest, Brest, France*

C H R O N I C L E

Article history:
Received: November 28, 2022
Received in revised format: December 27, 2022
Accepted: January 31, 2023
Available online: February 1, 2023

Keywords:
Project scheduling
Scheduling
Project management
Artificial neural network
Convolutional neural network
RCPSP
Resource constraint

A B S T R A C T

All projects require a structure to meet project requirements and achieve established goals. This framework is called project management. Therefore, project management plays an important role in national development and economic growth. Project management includes various knowledge areas such as project integration management, project scope management, project schedule management, etc. The article focuses on the resource-constrained project scheduling known as problem so- called the resource-constrained project scheduling problem (RCPSP). The RCPSP is a part of schedule management. The standard RCPSP has two important constraints, resource constraints and precedence relationships of activities during project scheduling. The objective of the problem is to optimize and minimize the project duration, subject to the above constraints. In this paper, we develop a convolutional neural network approach to solve the standard single mode RCPSP. The advantage of this algorithm over conventional methods such as metaheuristics is that it does not need to generate many solutions or populations. In this paper, the serial schedule generation scheme (SSGS) is used to schedule the project activities using an evolved convolutional neural network (CNN) as a tool to select an appropriate priority rule to filter out a candidate activity. The evolved CNN learns according to the eight project parameters, namely network complexity, resource factor, resource strength, average work per activity, etc. The above parameters are the inputs of the network and are recalculated at each step of the project planning. Moreover, the developed network has priority rules which are the outputs of the developed neural network. Therefore, after the learning process, the network can automatically select an appropriate priority rule to filter an activity from the eligible activities. In this way, the algorithm is able to schedule all project activities according to the given project constraints. Finally, the performance of the Convolutional Neural Network (CNN) approach is investigated using standard benchmark problems from PSPLIB in comparison to the MLFNN approach and standard metaheuristics.

1. Introduction

This article deals with the resource-constrained project scheduling is known problem, so-called the resource-constrained project scheduling problem (RCPSP). Since the mentioned problem is defined in the context of project schedule, we should know about the project definition and its specifications. A project is defined as a temporary attempt or endeavor undertaken to create a unique product, service, or result. Projects are defined or undertaken to meet objectives and achieve results. Project objectives may result in one or more outcomes, e.g., a unique product, a unique result, or a unique combination of products, etc. Some examples of projects include building infrastructure, creating software, conducting research, developing a service, etc. It is true that a project is a time-limited endeavor, but its results can be a process or the processes after the project ends (GUIDE Sixth Edition, 2017; PMBOK GUIDE Seventh Edition, 2021).

* Corresponding author. Tel.: + 33 750014801
E-mail address: amir.golab@isen-ouest.yncrea.fr (A. Golab)

doi: 10.5267/dsl.2023.2.002

A project needs a structure to achieve the set objectives. This framework is called project management, i.e., project management is defined as the application of knowledge tools, skills, and techniques to project activities to meet project requirements. Project management consists of various knowledge areas, such as project integration management, project scope management, project schedule management, and others. As mentioned earlier, project schedule management is one of the areas of project management that includes the processes used to manage the timely completion of the project. Two of these processes are called sequence activities and develop schedules. The importance of the sequence activities process is that this process establishes the logical sequence of project activities in order to achieve the greatest efficiency, considering all the constraints of the project. The process of developing a schedule is the investigation of the sequence of activities, duration, resource requirements, and schedule constraints to obtain the schedule model for project execution (PMBOK GUIDE Sixth Edition, 2017). In this research, we concentrate on the project schedule management knowledge area.

A project is represented in science by two types of project schedule network diagrams, called activity-on-node (AoN) and activity-on-arc (AoA). This research benefits from the activity-on-node (AoN) diagram to drive calculations and explain the structure of a project. This type of network diagram represents the order in which activities should be scheduled according to the existing logical precedence between project activities. An activity-on-node diagram (AoN) usually consists of nodes representing activities and arrows showing the relationships between the activities (Golab, Gooya, Alfalou, & Cabon, 2022; Koulinas, Kotsikas, & Anagnostopoulos, 2014).

The resource-constrained project scheduling problem (RCPSP) is academically similar to the stated processes called sequence activities and developing schedules to obtain a project schedule. The RCPSP focuses on determining an order of activities to minimize and optimize project duration given project constraints, i.e., precedence relations between activities and resource constraints. The standard RCPSP is represented by the set $A = \{1,...,i\}$ of activities constrained by two types of constraints. The constraints are so-called precedence relations through activities and resource constraints, in particular renewable resources.

The standard RCPSP is mathematically defined by four equations, where Eq. (1) states the standard objective of the problem, which is to optimize or minimize the project duration considering the two main constraints. Eq. (2) ensures that all priority relationships between activities are satisfied during the project schedule. This equation explains that activity i cannot be begun until its immediate predecessors are fully completed. Eq. (3) states that the first and last tasks are dummy activities and milestones with zero duration. There is a set of renewable resources $R = \{1,...,r\}$ during project execution and all activities require u_{irt} units per time for execution. Eq. (4) represents the second respectable constraint of the RCPSP , which is to consider the available resource quantities period by period. Therefore, the feasibility of resources during the project schedule is achieved (Bouleimen & Lecocq, 2003; Kolisch & Hartmann, 2006; Koulinas, Kotsikas, & Anagnostopoulos, 2014; Roy & Sen, 2019; Golab, Sedgh Gooya, Alfalou, & Cabon, 2022). The defined elements are presented in Table 1.

$$\min f_n \tag{1}$$

subject to

$$f_i \leq f_j - d_j \qquad \forall \left(A_i, A_j\right) \in Pred \tag{2}$$

$$f_1 = 0\,, d_1 = 0\,, d_n = 0 \qquad \text{the activities 1 and n are dummy activities or milestones} \tag{3}$$

$$\sum_{i \in P_t} u_{irt} \leq U_r \qquad \forall\ t = 1, \dots, f_n \text{ and } \forall\ r \in R \tag{4}$$

Table 1
The definitions of employed elements in the standard RCPSP

Element	Definition
A	The set of project activities that consist of the activities with duration di and i = 1, 2, 3, ..., n. The activities 1 and n are dummy or milestones.
R	The set of $R = \{1,...,r\}$ represents the renewable resources required to project execution.
U_r	The available quantities of renewable resource r
f_i	The finish time of activity i
f_j	The finish time of activity j, which is the immediate successors of activity i
$Pred$	The set of $Pred$ consisting of ordered pairs (A_i, A_j) shows that A_j is an immediate successor of A_i
u_{irt}	The amount of renewable resource r consumed by activity i in the period t.

Several researchers have attempted to develop meta-heuristics to solve the problem. However, there is a lack of investigation of the RCPSP using new techniques such as neural networks (Golab, Gooya, Alfalou, & Cabon, 2022). In the following, we mention some related articles that employed the meta-heuristics to solve the RCPSP. One solution method, called the self-adaptive genetic algorithm, has been proposed, where the algorithm benefits from the well-known activity list representation and considers two different decoding methods to minimize the makespan of the project (Hartmann, 2002). A hybrid genetic algorithm (HGA) has been proposed for the resource constrained project scheduling problem (RCPSP). The proposed research introduces many changes in the genetic algorithm paradigm. The developed algorithm proposes a crossover operator specific to the RCPSP, a local improvement operator, a new parent selection method, and a two-phase strategy technique (Valls, Ballestín, & Quintanilla, 2008). The proposed genetic algorithm introduces modifications to the standard genetic algorithm, such as a new selection operator to filter out the parents to be recombined, a new two-point crossover operator with a specific crossover order, and a linearly decreasing probability- based on linear decreasing probability (Liu, Liu, Shi, & Li, 2020). For more information on meta-heuristic and related articles, we address the review article provided by (Golab, Gooya, Alfalou, & Cabon, 2022).

Some researchers have also proposed algorithms based on neural networks. In the following, we mention related articles. Augmented neural networks were developed to solve the task scheduling problem. The algorithm consists of a hybrid of heuristics and neural networks. The objective of the problem is defined as minimizing the project duration when scheduling n jobs on m machines (Agarwal, Pirkul b, & Jacob b, 2003). A neural network based heuristics was proposed to minimize the project duration. The algorithm is integrated with the scheduling scheme to automatically select the appropriate priority rules for project scheduling (Shou, 2005). A neurogenetic approach has been proposed in which the algorithm is composed of neural networks and genetic algorithms. In the proposed approach, the search process is performed by GA iterations to perform the global search, and NN iterations are employed for the local search. The GA search iterations and the NN are interleaved in such a way that the NN can select the best solution from the GA pool. In addition, the suitable solutions obtained by the NN search are included in the GA population to be used in the GA iterations (Agarwal, Colak, 2011). An artificial neural network has been used to plan 240 projects such as offices, schools, etc. that are designed under limited resources. Three priority rules labelled latest finish time, minimum slack time, and maximum remaining path length are used to determine the amount of resources for these projects (Özkan & Gulçlçek, 2015). The training performance of a feed-forward neural network was studied on the standard RCPSP, where the artificial neural network learns based on eight parameters computed as inputs and priority rules as outputs (Golab, Sedgh Gooya, Alfalou, & Cabon, 2022). A multilayer feed-forward neural network (MLFNN) was developed to deal with the standard single-mode RCPSP. The presented MLFNN learns based on eight factors, namely average work per activity, percentage of work remaining, etc. as inputs and identified priority rules as outputs. The proposed algorithm benefits from the serial schedule generation scheme as a schedule decoding function (Golab, Sedgh Gooya, Alfalou, & Cabon, 2022).

In this article, we employ a convolutional neural network (CNN) approach to solve the standard resource constrained project scheduling problem (RCPSP). For this purpose, the developed convolutional neural network (CNN) is fed with eight project parametric characterizations to select an appropriate priority rule as the output of the CNN. Therefore, at each scheduling step, the procedure can select an appropriate activity based on the selected priority rule among the eligible activities. The selected activity is added to the project schedule. The algorithm proceeds to schedule all the activities of the project considering the given project constraints. We also benefit from the serial schedule generation scheme as a function to the decoding function.

The main body of the article is organized as follows. In Section 2, the convolutional neural network is explained, including the developed convolutional neural network, the inputs called project parametric characterizations, and the priority rules representing the outputs of the developed CNN. In Section 3, the algorithm used is presented. The computational analysis is presented in section 4. Finally, a conclusion is drawn.

2. Convolutional neural network (CNN)

In this section, an overview of convolutional neural networks and their components is given. Following this part, the developed convolutional neural network with its inputs and outputs is explained.

2.1 Overview of convolutional neural network

Artificial neural networks (ANN) are flexible mathematical models and powerful machine learning methods that originally emerged from the functional structure of the human brain. These networks simulate the learning mechanism in the biological organism, where billions of interconnected neurons process data in parallel. Artificial neural networks consist of computational units called neurons. The neurons process data while interconnected by adoptable weight connections. Artificial neural networks are a popular and helpful model for classification, prediction, optimization, clustering, etc.(Svozil, Kvasnicka, & Pospichal, 1997; Wang , 2003; Choi, Coyner, Kalpathy-Cramer, Chiang, & Campbell, 2020).

Artificial neural networks are divided into two groups: feed forward neural networks and feed backward neural networks. In this context, we focus on feed forward neural networks. A feed-forward neural network (FFNN) is an algorithm that consists of ordered layers similar to the neural processing units of the human brain. In this type of neural network, each unit or neuron in one layer is connected to the other neurons in the other layers. The connections between neurons are not all the same, as each connection can have a different weight. The weights of the network connections determine the possible part of the knowledge of the network (Abiodun, et al., 2018; Choi, Coyner, Kalpathy-Cramer, Chiang, & Campbell, 2020).

A crucial feature of neural networks is the selection of an appropriate network size for a given problem. The network size explains the number of layers in a network, the number of neurons per layer, and the number of connections between neurons (Aggarwal, 2018). The prediction accuracy of a neural network also depends on the type of activation function or transfer function employed in the NN. Activation functions are functions employed to calculate the weighted sum of inputs and biases. In other words, activation functions are used to control the outputs of neural networks in different domains. Activation functions assist in learning complicated mappings between inputs and corresponding outputs. These functions transfer the input signals to the output signals. Therefore, activation functions dynamically shape the network and give it the ability to extract complicated information from the data. Therefore, the network can apply the backpropagation optimization strategy to calculate the errors or losses related to the weights and optimize the weights using gradient descent or other optimization techniques to reduce the errors (Sagar, Sharma, & Athaiya, 2017; Nwankpa, Ijomah, Gachagan, & Marshall, 2018). In the developed convolutional neural network (CNN), we use the relu activation function, which is explained below.

In this research, we focus on convolutional neural networks (CNN), which is a kind of feedforward neural network to solve the problem. A convolutional neural network (CNN) is one of the remarkable networks in the field of Deep Learning, which is capable of extracting features from data with convolutional structures. In a CNN, each neuron benefits from local connections, i.e., each neuron is not connected to all neurons of the previous layer, but only to a small number of neurons, which helps to reduce parameters and speed up convergence. There is also weight sharing between a group of connections, which can lead to further reduction of parameters. In addition, there are pooling layers that are able to reduce the amount of data while preserving useful information by reducing the dimensions (down sampling). This phenomenon can reduce the number of parameters by removing trivial features (Li, Liu, Yang, Peng, & Zhou, 2021).

In general, in a convolutional neural network (CNN), data is input directly into the network, then layers or stages of convolution and pooling do the process. The processed data feeds one or more fully linked layers, as in a regular neural network. Finally, the last output of the fully connected layer is the desired output (Rawat & Wang, 2017).

The CNN architecture consists of three layers: Convolutional layer, pooling layer, and fully linked layer. The convolutional layer is the main design unit of a convolutional neural network. This layer controls the output of the associated inputs in the receptive field. This output is achieved by kernels that are convolved over the data by computing the dot product between the input and filter values, creating an activation map using this filter. In this way, the CNN can quickly learn the appropriate filters to activate when a particular type of feature is observed at a particular position on the input. The main task of the pooling layer is to reduce the spatial size of the representation in order to reduce the number of parameters and computations in the model. This not only increases the speed of the calculations, but also avoids the problem of overfitting. The most common form of pooling layer is called max pooling. As mentioned earlier, fully connected layers are the standard neural network, which attempts to make predictions or classifications. This layer obtains the full connections through each neuron in that part of the network (Rawat & Wang, 2017; Aloysius & Geetha, 2017; Dhillon & Verma, 2020; Li, Liu, Yang, Peng, & Zhou, 2021).

CNNs are basically learned in a supervised manner through the so-called backpropagation algorithm (BP), which involves two phases. The first phase is the forward phase, and the second phase is the backward phase. In the forward phase, the activation functions are propagated from the input layer to the output layer, and in the backward phase, the detected errors between the detected actual value and the target value are back propagated in the output layer to upgrade the weights and bias values (Aggarwal, 2018; Kiranyaz, et al., 2021).

2.2 Developed convolutional neural network

It has been shown that convolutional neural networks (CNNs) are formed in different types of layers. These layers consist of a series of interconnected neurons that benefit from activation functions. In addition, the neural network is fed with data through the input layer. As presented in Fig. 1, the developed CNN is fed with eight different data. The input layer includes eight different parameters, which are explained in detail in the next section. The parameters mentioned are average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of remaining successors, average units per day, network complexity, resource factor, and resource strength. The inputs of the CNN are recalculated at each stage of project scheduling to characterize the new subproject or new step; in this way, the conditions for selecting an eligible activity for project scheduling are prepared. For example, in a project with 62 activities, there are 62 scheduling phases or steps, so these parameters or the inputs are recalculated 62 times to characterize the subprojects.

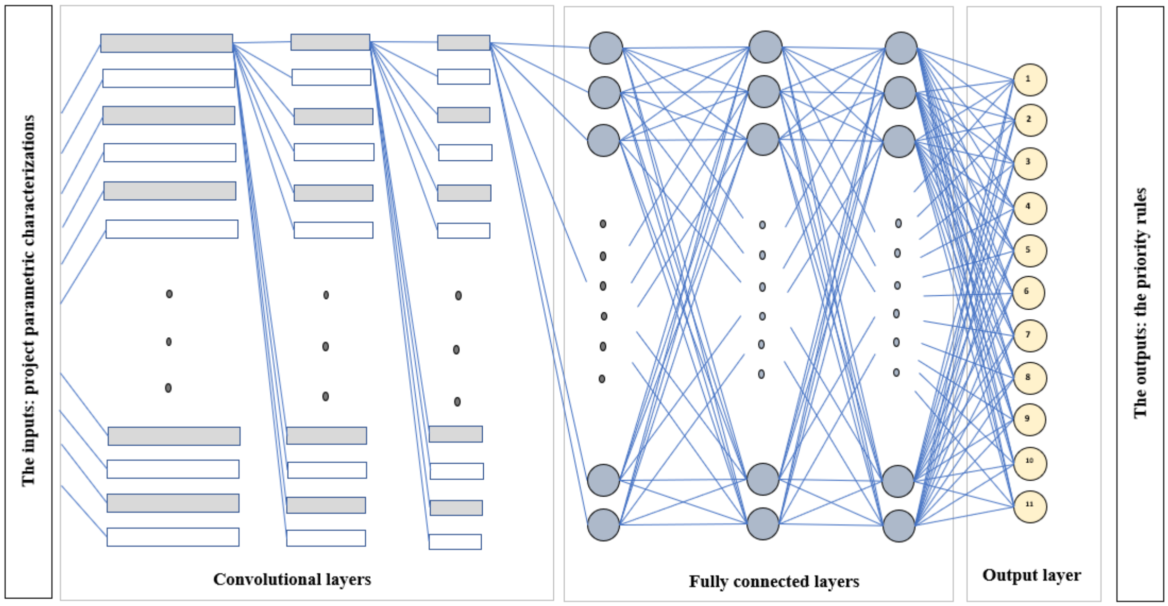

Fig. 1. The developed convolutional neural network(CNN) is composed of eight inputs as input layer, convolutional and fully connected layers, and the output layer.

The output layer of the developed convolutional neural network consists of priority rules, which are detailed in sections 2-4. First, eleven priority rules are set for the output layer. The task of the outputs or priority rules is to select an eligible activity from the list of eligible activities at each step of project scheduling by assigning a value to the eligible activities. For example, if the output of the network is the latest finish time (LFT), the activity with the minimum finish time is selected for scheduling. To check the performance of the developed CNN, a different number of neurons or priority rules is assigned to the output layer.

If no activation function is used in a convolutional neural network, the output would be just a simple linear function. Therefore, we need to apply the activation function to dynamically build the network and give it the ability to extract complex data. For this purpose, the relu activation function is embedded in the developed convolutional neural network. The so-called relu is a nonlinear activation. This function is capable of calculate mathematically as $f(x) = \max(0, x)$, and this function is widely used in neural networks. This activation function is used for the convolutional and fully connected layers of the developed CNN.

As mentioned earlier, the developed network is verified with eleven, seven, five, four and three priority rules as outputs of the developed convolutional neural network. The obtained training performance results are reported in Table 3, Table 4 and Table 5.

2.3 The inputs of the developed convolutional neural network: project parametric characterizations

The inputs of the developed convolutional neural network are eight project parametric characterizations. These parameters characterize the structure of the project network. They were named network complexity (NC), resource factor (RF), and resource strength (RS), and were re-explained by (Kolisch, Sprecher, & Drexl, 1995). These parameters explain the constraints of the average number of successors per unscheduled activities, the average proportion of resources requested per activity, and the measurement of the availability of resources during the project execution, receptively (Sprecher & Kolisch, 1997). In addition, we use other parameters introduced by (Golab, Sedgh Gooya, Alfalou, & Cabon, 2022), namely average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of remaining successors, and average units per day, which can be calculated using the project data. All parameters are used to calculate the factors of a complete project, but they can be modified to equip the scenarios of partial schedules, which means that in each stage of the project schedule the mentioned parameters are recalculated to characterize the new phase or sub-project. The inputs of the developed CNN are calculated according to the formulas below. Also, the definitions of the elements used are represented in Table 2.

Network complexity (NC) = $\sum_{i\in US} S_i \Big/ |US|$ (5)

Resource factor (RF) = $\frac{1}{|US|}\frac{1}{|R|}\sum_{i\in US}\sum_{r\in R}\begin{cases}1 & if\ u_{ir}>0\\ 0 & otherwise\end{cases}$ (6)

Resource strength (RS) = $\sum_{r\in R} U_r - U_r^{max} \Big/ U_r^{max} - U_r^{min}$ (7)

Average work per activity (AWA) = $\sum_{i\in US} w_i \Big/ |US|$ (8)

Percentage of remaining work (PRW) = $\sum_{i\in US} w_i \Big/ W$ (9)

Percentage of unscheduled activities (PUA) = $|US| \Big/ |A|$ (10)

Percentage of remaining successors (PRS) = $\sum_{i\in US} S_i \Big/ \sum_{i\in A} S_i$ (11)

Average units per day (AUD) = $\sum_{i\in US} w_i \Big/ \sum_{i\in US} d_i$ (12)

Table 2
The definitions of employed elements in the formulas' parameters

Element	Definition
A	Set of project activities
US	Set of unscheduled activities
U_r	Available quantities of the renewable resource r
U_r^{max}	Maximum quantity of the resource r
U_r^{min}	Minimum quantity of the resource r
W	Total work-content of the project
w_i	Work-content of activity i calculated by $w_i = d_i * u_{ir}$
d_i	Duration of activity i
S_i	Number of the immediate successors of activity i

2.4 The outputs of the developed convolutional neural network: priority rules

The outputs of the developed convolutional neural network are rules, called priority rules. These rules are employed to select an eligible activity among the eligible activities contained in the available decision set. The priority rules select activities according to their selection criteria for project scheduling. To select an eligible activity, a priority rule assigns values to the eligible activities available in the decision or eligible set. The assigned values are used to select an eligible activity according to the selection criteria, which can be the minimum or maximum value. Different priority rules will result in the selection of different activities and consequently different project duration results obtained depending on the project specifications, such as the number of resources. If there is more than one activity with the same assigned value through eligible activities, it is practical to randomly select the activity with the smallest activity label. Therefore, it is key to use an appropriate priority rule that selects an activity among the eligible activities during the project schedule (Ulusoy & Özdamar, 1989; Olaguíbel & Goerlich, 1989; Kolisch, 1996; Kolisch & Hartmann, 1999; Özkan & Gulçlçek, 2015). For more information about the priority rules, we refer you to the article written by (Golab, Sedgh Gooya, Alfalou, & Cabon, 2022).

As reported in Table 2, there are different priority rules classified by their selection criteria. We use these priority rules as outputs of the developed convolutional neural network.

Table 2
The priority rules and their selection criteria

Priority rules	selection criteria
Earliest start time (EST)	Min
Latest start time (LST)	Min
Earliest finish time (EFT)	Min
Latest finish time (LFT)	Min
Shortest processing time (SPT)	Min
Total resource demand (TRD)	Min
Total resource scarcity (TRS)	Min
Most total successors (MTS)	Max
Minimal slack (MSLK)	Min
Greatest rank positional weight (GRPW)	Max
Weighted resource utilization ration and precedence (WRUP)	Max

3. Algorithm

The algorithm uses a schedule generation scheme as a function to generate and decode a feasible schedule. In general, there are two particular schedule generation schemes (SGS) for generating feasible schedules, called serial schedule generation scheme (SSGS) and parallel schedule generation scheme (PSGS). For further explanation and better understanding of the differences between these two schemes, the reader is referred to the article by Kolisch and Hartmann (1999). The developed algorithm benefits from the serial schedule generation scheme (SSGS), which generates active schedules that contain at least a feasible or optimal solution. The SSGS can construct a sequence of activities that leads to one feasible or optimal schedule because the SSGS always generates feasible schedules (Koulinas, Kotsikas, & Anagnostopoulos, 2014; Golab, Sedgh Gooya, Alfalou, & Cabon, 2022).

The serial schedule generation scheme is composed of n steps during project schedule, where n refers to the steps or the number of project activities. For instance, in a project with 62 activities, there are 62 scheduling phases or steps. Therefore, these eight inputs or parameters are recalculated 62 times to characterize the new subprojects. The SSGS benefits from three sets are called the un-scheduled set US_n, the eligible set E_n and the scheduled set $S_n$which should be updated at each step or iteration of scheduling. Therefore, an unscheduled activity can be a member of the eligible set if its predecessors were scheduled. Also, an eligible activity can be a member of the scheduled activity if it has been scheduled (Kolisch & Hartmann, 1999; Golab, Sedgh Gooya, Alfalou, & Cabon, 2022).

The developed convolutional neural network is trained by the created dataset to determine the optimal weights before the project scheduling process begins. The weights are used for the computations of the network to select an appropriate output or priority rule at each step of scheduling.

The scheduling process is initialized with a partial schedule containing only the dummy start activity at time zero. Then, the inputs or eight parameters are recalculated, resulting in the selection of a priority rule at each iteration or step of the project schedule. In the proposed algorithm, the selection of a priority rule as output is the task of the developed convolutional neural network (CNN). The selected priority rule assigns values to the eligible activities, which leads to the selection of an eligible activity from the eligible set E_n according to the selection criteria. Then, the selected activity is added to the project schedule at the earliest possible time that is feasible in terms of both precedence and resource availability. The algorithm continues to schedule all the activities of the project. In this way, the project duration, which is the final objective of the problem, is determined at the end of the algorithm. For better understanding, the reader refers to Fig. 2, which explains the process of the developed algorithm.

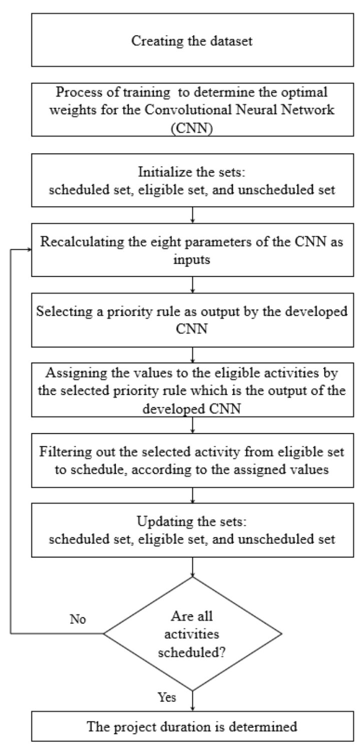

Fig. 2. The flowchart shows the explained algorithm. The process starts with training based on the created dataset and continues with scheduling all project activities to determine the final project duration.

4. Computational analysis

In this section, the obtained results and the related discussions about the research are presented. The obtained results include the training performance results of the developed convolutional neural network and the comparative results. Also, the strengths and weaknesses of the developed CNN and algorithm are mentioned below.

4.1 Convolutional neural network training performance results and discussion

The developed convolutional neural network is trained with the created dataset using the projects of PSPLIB. To create the dataset, all eight project parametric characterizations are calculated at each stage of the selected PSPLIB projects. The dataset feeds the developed CNN. To determine the output column of the dataset for each phase or subproject, the priority rule that gives a minimum project duration in that phase of scheduling is selected.

we need to equalize the initial dataset to achieve better performance of the developed convolutional neural network (CNN). This means that after balancing the initial dataset, the number of priority rules as outputs in the final dataset is the same.

The performance of the convolutional neural network is verified with eleven, seven, five, four, and three priority rules or neurons as the output layer. Since the size of the network also matters, the developed convolutional network was analyzed with one, two and three convolutional layers and three fully connected layers and different number of epochs of 500, 1000 and 2000. The developed CNN was run 10 times for each mode to determine the average training performance.

The training performance results are reported in Table 3, Table 4 and Table 5. The goal of the training process of the developed CNN is to find the optimal set of weights that will help to obtain the correct output. Therefore, the inputs, the outputs, the test sets and the required parameters must be initialized before training.

Table 3
Training performance of developed convolutional neural network (CNN) with one convolutional layer

Priority rules used for output layer	convolutional layers	Epoch	Min accuracy	Max accuracy	Average accuracies
Three priority rules are used as outputs: EST, LST and EFT	1	500	55%	60%	57%
	1	1000	58%	67%	62%
	1	2000	65%	72%	68%
Four priority rules are used as outputs: EST, LST, EFT and LFT	1	500	50%	55%	53%
	1	1000	57%	67%	61%
	1	2000	62%	70%	67%
Five priority rules are used as outputs: EST, LST, EFT, LFT and MTS	1	500	47%	53%	50%
	1	1000	56%	61%	58%
	1	2000	62%	68%	65%
Seven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	1	500	44%	56%	47%
	1	1000	46%	55%	51%
	1	2000	53%	64%	59%
Eleven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP and TRS	1	500	36%	50%	43%
	1	1000	41%	52%	44%
	1	2000	44%	58%	51%

Table 4
Training performance of developed convolutional neural network (CNN) with two convolutional layers

Priority rules used for output layer	convolutional layers	Epoch	Min accuracy	Max accuracy	Average accuracies
Three priority rules are used as outputs: EST, LST and EFT	2	500	60%	69%	64%
	2	1000	67%	71%	69%
	2	2000	68%	74%	70%
Four priority rules are used as outputs: EST, LST, EFT and LFT	2	500	61%	65%	63%
	2	1000	66%	70%	67%
	2	2000	66%	71%	68%
Five priority rules are used as outputs: EST, LST, EFT, LFT and MTS	2	500	55%	64%	60%
	2	1000	61%	68%	64%
	2	2000	62%	69%	65%
Seven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	2	500	52%	60%	56%
	2	1000	59%	67%	63%
	2	2000	62%	74%	65%
Eleven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP and TRS	2	500	46%	57%	51%
	2	1000	47%	62%	56%
	2	2000	44%	63%	55%

Table 5
Training performance of developed convolutional neural network (CNN) with three convolutional layers

Priority rules used for output layer	convolutional layers	Epoch	Min accuracy	Max accuracy	Average accuracies
Three priority rules are used as outputs: EST, LST and EFT	3	500	65%	72%	68%
	3	1000	67%	73%	70%
	3	2000	69%	73%	71%
Four priority rules are used as outputs: EST, LST, EFT and LFT	3	500	66%	70%	68%
	3	1000	66%	69%	67%
	3	2000	66%	70%	68%
Five priority rules are used as outputs: EST, LST, EFT, LFT and MTS	3	500	65%	68%	66%
	3	1000	64%	69%	66%
	3	2000	66%	71%	68%
Seven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	3	500	58%	65%	61%
	3	1000	62%	65%	64%
	3	2000	62%	69%	66%
Eleven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP and TRS	3	500	49%	61%	55%
	3	1000	54%	66%	58%
	3	2000	53%	61%	56%

To obtain the optimal weights, 80% of the data from the balanced dataset were used for training the developed CNN, and 20% of the data, which is the remaining data, was assigned to the testing process. In addition, the learning rate and batch size parameters were set to 0.0007 and 64, respectively.

We confirm the priority rules with probability greater than 0.5 at each stage of scheduling as the final output of the developed CNN. According to the obtained training performance results, it is obvious that the performance of the developed convolutional neural network (CNN) increases when the fixed number of outputs is reduced. For instance, the probability of selecting the correct priority rule is higher when three neurons are used as outputs than when eleven neurons were used.

4.2 Comparative results and discussion

The goal of this research is to employ a convolutional neural network (CNN) embedded in an algorithm to solve the resource constrained project scheduling problem (RCPCP). The superiority of the proposed algorithm over evolutionary methods or metaheuristics is that it is not necessary to generate many populations or solutions. The developed CNN trains the weights only once, and the obtained weights are then used to schedule all project instances.

The developed CNN and training performance were illustrated in the previous section. The algorithm is applied to schedule the standard instances after training the developed convolutional neural network. The standard problem instances included projects with four types of renewable resources and 60 and 120 activities selected from the PSPLIB.

The obtained results are presented in the form of an average percentage of deviations from the lower bound based on the critical path for the project instances with 60 activities and 120 activities. The critical path lower bound just addresses the sequencing of all the activities of a project according to their precedence relations. This means that if a project is scheduled only according to the precedence relations, which is the first constraint of the problem, the obtained project duration corresponds to the critical path lower bound.

Table 6 represents the competitive results for 181 instances of J60 and 160 instances of J120. In general, the competitive results confirm that the CNN developed in this paper performs better than the MLFNN developed in another article. The results summarized in Table 6 also show that the average deviations from the critical path lower bound are better when the performance of the developed convolutional neural network (CNN) increases. In section 4-1, it was mentioned that the performance of the developed CNN improves when the number of outputs used is reduced.

Table 7 summarizes comparative results for J60 standard instances. These results confirm that the developed CNN embedded in the algorithm achieves average deviations of 16.57%, 15.97%, and 16.19% when the developed CNN benefits from three, four, and five neurons as outputs for J60, respectively. These are not the best results, but they can be quite competitive. Table 8 shows the comparative results for J120 standard instances. The obtained results confirm that our algorithm achieves better performance in handling larger projects. The average deviations are 38.39%, 37.77% and 39.48% when the developed CNN employs three, four and five neurons as outputs for J120 instances, respectively. The comparative results in Table 8 present that the selection of the priority rules by the developed CNN leads to sufficiently competitive results.

The results obtained are not the best among other results obtained by other researchers, but the advantage of the proposed algorithm over evolutionary methods or metaheuristics is that it is not necessary to generate many solutions or populations, on the contrary, the project can be scheduled only by generating a sequence activity.

The results obtained confirm that the performance of the proposed procedure and the solutions obtained can be improved. We suggest that the results can be improved by selecting appropriate priority rules as outputs and developing more neural networks. Another suggestion is to use the proposed algorithm for scheduling the specialized projects. This means that the results can be more competitive if the training dataset is specialized.

Table 6
Percentage of average deviations from critical path lower bound for the J60 and J120. The average deviations obtained by our algorithm using the CNN approach, represent that the CNN approach performs better compared to the MLFNN approach developed in another article.

Priority rules used for output layer	Number of activities			
	60		**120**	
	Approach			
	CNN approach	MLFNN approach	CNN approach	MLFNN approach
Three priority rules are used as outputs: EST, LST and EFT	16.57	15.97	38.39	37.77
Four priority rules are used as outputs: EST, LST, EFT and LFT	15.97	16.28	37.77	39.74
Five priority rules are used as outputs: EST, LST, EFT, LFT and MTS	16.19	47.04	39.48	89.61
Seven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	20.48	45.59	53.06	86.39
Eleven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP and TRS	63.33	58.96	124.72	117.42

Table 7
The percentage of average deviations from critical path lower bound for the J60 obtained by our algorithm represents that they are not the best but can be competitive.

Reference	Algorithm	Deviation
Our algorithm	CNN approach (Three neurons used as outputs)	16.57
Our algorithm	CNN approach (Four neurons used as outputs)	15.97
Our algorithm	CNN approach (Five neurons used as outputs)	16.19
(Golab, S. Gooya, Alfalou, & Cabon, 2022)	MLFNN approach (Three neurons used as outputs)	15.97
(Golab, S. Gooya, Alfalou, & Cabon, 2022)	MLFNN approach (Four neurons used as outputs)	16.28
(Valls, Ballestin, & Quintanilla, 2008)	Hybrid GA	11.56
(Alcaraz & Concepción, 2001)	Genetic Algorithm	NA
(Hartmann, 2002)	Genetic Algorithm	12.21
(Nonobe & Ibaraki, 2002)	Tabu Search	12.97
(Koulinas, Kotsikas, & Anagnostopoulos, 2014)	Particle Swarm Optimization-HH	11.74
(Bouleimen & Lecocq, 2003)	Simulated Annealing	12.75
(Mendes, Gonçalves, & Resende, 2009)	Genetic Algorithm	11.72
(Chen, Shi, Teng, Lan, & Hu, 2010)	Hybrid (ACO and SS)	11.75
(Agarwal, Colak, & Erenguc, 2011)	Neurogenetic	11.51
(Mobini, Mobini, & Rabbani, 2011)	Artificial Immune Algorithm	11.17
(Gonçalves, Resende, & Mendes, 2011)	Genetic Algorithm	11.56
(Wang & Fang, 2012)	Hybrid Estimation of Distribution Algorithm	11.44
(Chen R.-M. , 2011)	Particle swarm optimization	12.03
(Ziarati & Akbari, 2011)	Bee Algorithms	12.55
(Proon & Jin, 2011)	Genetic Algorithm with Neighborhood Search	11.35
(Liu, Liu, Shi, & Li, 2020)	Genetic Algorithm	11.74
(Zamani, 2017)	Genetic Algorithm	11.61
(Lim, Ma, Rodrigues, & Tan, 2013)	Hybrid Genetic Algorithm	11.73

Table 8
The percentage of average deviations from critical path lower bound for the J120 obtained by our algorithm shows that they are not the best, but sufficiently competitive.

Reference	Algorithm	Deviation
Our algorithm	CNN approach (Three neuron used as outputs)	38.39
Our algorithm	CNN approach (Four neuron used as outputs)	37.77
Our algorithm	CNN approach (Five neurons used as outputs)	39.48
(Golab, S. Gooya, Alfalou, & Cabon, 2022)	MLFNN approach (Three neuron used as outputs)	37.77
(Golab, S. Gooya, Alfalou, & Cabon, 2022)	MLFNN approach (Four neuron used as outputs)	39.74
(Valls, Ballestin, & Quintanilla, 2008)	Hybrid GA	34.07
(Alcaraz & Concepción, 2001)	Genetic Algorithm	39.36
(Hartmann, 2002)	Genetic Algorithm	37.19
(Nonobe & Ibaraki, 2002)	Tabu Search	40.86
(Koulinas, Kotsikas, & Anagnostopoulos, 2014)	Particle Swarm Optimization-HH	35.20
(Bouleimen & Lecocq, 2003)	Simulated Annealing	40.82
(Mendes, Gonçalves, & Resende, 2009)	Genetic Algorithm	35.87
(Chen, Shi, Teng, Lan, & Hu, 2010)	Hybrid (ACO and SS)	35.19
(Agarwal, Colak, & Erenguc, 2011)	Neurogenetic	34.65
(Mobini, Mobini, & Rabbani, 2011)	Artificial Immune Algorithm	30.04
(Gonçalves, Resende, & Mendes, 2011)	Genetic Algorithm	35.94
(Wang & Fang, 2012)	Hybrid Estimation of Distribution Algorithm	34.83
(Chen R.-M. , 2011)	Particle swarm optimization	35.71
(Ziarati & Akbari, 2011)	Bee Algorithms	37.72
(Proon & Jin, 2011)	Genetic Algorithm with Neighborhood Search	33.45
(Liu, Liu, Shi, & Li, 2020)	Genetic Algorithm	34.88
(Zamani, 2017)	Genetic Algorithm	34.59
(Lim, Ma, Rodrigues, & Tan, 2013)	Hybrid Genetic Algorithm	34.95

5. Conclusion

In this paper, an algorithm equipped with a convolutional neural network was proposed to deal with the resource-constrained project scheduling problem (RCPSP). As mentioned earlier, the superiority of the proposed algorithm over evolutionary methods or meta-heuristics is that it does not require generating populations or numerous solutions. On the contrary, the proposed algorithm generates a solution based on the trained CNN. The proposed convolutional neural network benefits from the eight inputs, called project parametric characterizations and a different number of priority rules as outputs. It is obvious that the project specifications change during the project schedule. Therefore, the performance of the priority rules in selecting an eligible activity depends on the project specifications, such as the existing project constraints. Therefore, different priority rules or activity selection methods are suitable for different types of subprojects or projects. Therefore, in this article, we developed a convolutional neural network with a set of priority rules as outputs to select an appropriate eligible activity for the characterized sub-project.

The PSPLIB projects were used to create the dataset for training and testing the performance of the developed CNN. The training performance of the CNN was tested using the balanced data set after the parameters of the developed CNN were established. It was found that the performance of the CNN increased when the number of outputs was reduced. The duty of the convolutional neural network is to select a priority rule as an output after training the network. In this way, based on the selected priority rule, a suitable activity can be filtered from the list of eligible activities to be added to the project schedule. In the following, we applied the algorithm for scheduling the standard PSPLIB projects after training the CNN. The comparison results confirm that the results are competitive.

Although the results obtained are not yet the best, we believe that the developed convolutional neural network has adequate performance to deal with the RCPSP. This study can encourage researchers to exploit potential improvements to use this kind of algorithm to schedule specialized projects. In addition, researchers can develop neural networks to deal with project scheduling problems.

References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon, 4*(11).

Agarwal, A., Colak, S., & Erenguc, S. (2011). A Neurogenetic approach for the resource-constrained project scheduling problem. *Computers & Operations Research, 38*(1), 44–50.

Agarwal, A., Pirkul b, H., & Jacob b, V. S. (2003). Augmented neural networks for task scheduling. *European Journal of Operational Research, 151*(3), 481–502.

Aggarwal, C. C. (2018). *Neural networks and deep learning.* springer.

Alcaraz, J., & Concepción, M. (2001). A robust genetic algorithm for resource allocation in project scheduling. *Annals of operations Research, 102*(1), 83-109.

Aloysius, N., & Geetha, M. (2017). A review on deep convolutional neural networks. *2017 international conference on communication and signal processing (ICCSP)* (pp. 0588-0592). IEEE.

Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European journal of operational research, 149*(2), 268-281.

Chen, R.-M. (2011). Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications, 38*(6), 7102-7111.

Chen, W., Shi, Y.-j., Teng, H.-f., Lan, X.-p., & Hu, L.-c. (2010). An efficient hybrid algorithm for resource-constrained project scheduling. *Information Sciences, 180*(6), 1031-1039.

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020). introduction to machine learning, neural networks, and deep learning. *Translational Vision Science & Technology, 9*(2), 14 - 14.

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence, 9*(2), 85-112.

Golab, A., Gooya, S. E., Alfalou, A., & Cabon, M. (2022). Rview of conventional metaheuristic techniques for resource-constrained project scheduling problem. *Journal of Project Management, 7*(2), 95-110. doi:10.5267/j.jpm.2021.10.002

Golab, A., Sedgh Gooya, E., Alfalou, A., & Cabon, M. (2022). A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP). *Decision Science Letters, 11*(4), 407--418. doi:10.5267/j.dsl.2022.7.004

Golab, A., Sedgh Gooya, E., Alfalou, A., & Cabon, M. (2022). Investigating the performance of an artificial neural network for solving the resource constrained project scheduling problem (RCPSP). *In Pattern Recognition and Tracking XXXIII. 12101*, pp. 78-83. Florida: SPIE. doi:https://doi.org/10.1117/12.2618499

Gonçalves, J. F., Resende, M. G., & Mendes, J. J. (2011). A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem. *Journal of Heuristics, 17*(5), 467-486.

Hartmann, S. (2002). A self-adapting genetic algorithm for project scheduling under resource constraints. *Naval Research Logistics (NRL), 49*(5), 433-448.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical systems and signal processing, 151*.

Kolisch, R. (1996). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management, 14*(3), 179-192.

Kolisch, R., & Hartmann, S. (1999). heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computational analysis. *Project scheduling* (pp. 147 - 178). Boston: Springer.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research, 174*(1), 23-37.

Kolisch, R., Sprecher, A., & Drexl, A. (1995). characterisation and generation of a general class of resource-constrained project scheduling problem. *management science, 41*(10), 1693 - 1703.

Koulinas, G., Kotsikas, L., & Anagnostopoulos, K. (2014). A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. *Information Sciences, 277*, 680-693.

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 1-21.

Lim, A., Ma, H., Rodrigues, B., & Tan, S. T. (2013). New meta-heuristics for the resource-constrained project scheduling problem. *Flexible Services and Manufacturing Journal, 25*(1), 48-73.

Liu, J., Liu, Y., Shi, Y., & Li, J. (2020). Solving Resource-Constrained Project Scheduling Problem via Genetic Algorithm. *Journal of Computing in Civil Engineering, 34*(2).

Mendes, J. J., Gonçalves, J. F., & Resende, M. G. (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers & Operations Research, 36*(1), 92-109.

Mobini, M., Mobini, Z., & Rabbani, M. (2011). An Artificial Immune Algorithm for the project scheduling problem under resource constraints. *Applied Soft Computing, 11*(2).

Nonobe, K., & Ibaraki, T. (2002). Formulation and tabu search algorithm for the resource constrained project scheduling problem. *In Essays and surveys in metaheuristics* (pp. 557-588). Boston: Springer.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Olaguíbel, R. A., & Goerlich, J. M. (1989). Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis. *Advances in project scheduling*, 113-134.

Özkan, Ö., & Gulçlçek, Ü. (2015). A neural network for resource constrained project scheduling programming. *Civil Engineering and Management, 21*(2), 193 - 200.

PMI. (2017). *A guide to the project management body of knowledge, PMBOK GUIDE Sixth Edition.* Chicago: Project management institute.

PMI. (2021). *A guide to the project management body of knowledge, PMBOK GUIDE Seventh Edition.* Chicago: Project management institute, .

Proon, S., & Jin, M. (2011). A genetic algorithm with neighborhood search for the resource-constrained project scheduling problem. *Naval Research Logistics (NRL), 58*(2), 73-82.

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation, 29*(9), 2352-2449.

Roy, B., & Sen, A. K. (2019). Meta-heuristic Techniques to Solve Resource-Constrained Project Scheduling Problem. In *International conference on innovative computing and communications* (pp. 93-99). Springer.

Sagar, S., Sharma, S., & Athaiya, A. (2017). Activation fuctions in neurral networks. *towards data science, 6*(12), 310-316.

Shou, Y. (2005). A neural network based heuristic for resource-constrained project scheduling. *International Symposium on Neural Networks.* Berlin, Heidelberg.

Sprecher, A., & Kolisch, R. (1997). PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program. *European Journal of Operational Research, 6*(1), 205-216.

Svozil, D., Kvasnicka, v., & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems, 39*(1), 43-62.

Ulusoy, G., & Özdamar, L. (1989). Heuristic performance and network/resource characteristics in resource-constrained project scheduling. *Journal of the operational research society, 40*(12), 1145 - 1152.

Valls, V., Ballestín, F., & Quintanilla, S. (2008). A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research, 185*(2), 495-508.

Wang, L., & Fang, C. (2012). A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem. *Journal of Expert Systems with Applications, 39*, 2451-2460.

Wang, S.-C. (2003). Artificial neural network. In *Interdisciplinary computing in java programming* (pp. 81-100). Boston: springer.

Zamani, R. (2017). An evolutionary implicit enumeration procedure for solving the resource-constrained project scheduling problem. *International Transactions in Operational Research, 24*(6), 1525-1547.

Ziarati, K., Akbari, R., & Zeighami, V. (2011). On the performance of bee algorithms for resource-constrained project scheduling problem. *Applied Soft Computing, 11*(4), 3720–3733.