



HAL
open science

Resource constrained project scheduling problem (RCPSP) using artificial intelligence

Amir Golab

► **To cite this version:**

Amir Golab. Resource constrained project scheduling problem (RCPSP) using artificial intelligence. Artificial Intelligence [cs.AI]. Université de Bretagne occidentale - Brest, 2023. English. NNT : 2023BRES0046 . tel-04354306

HAL Id: tel-04354306

<https://theses.hal.science/tel-04354306>

Submitted on 19 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE DE DOCTORAT DE

L'UNIVERSITE
DE BRETAGNE OCCIDENTALE

ECOLE DOCTORALE N° 644
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication en Bretagne Océane*
Spécialité : *Informatique et Architectures numériques*

Par

Amir GOLAB

Automatisation de la planification dynamique de projet à ressource limitée par l'intelligence artificielle

Thèse présentée et soutenue à Pole numérique, UBO, Brest, le 18/09/2023
Unité de recherche : ISEN Yncréa ouest, Brest

Rapporteurs avant soutenance :

Rachid JENNANE Professeur, L'université de Orleans – Polytech 'Orléans IDP Laboratory
Alan LOUSSERT Enseignant chercheur, ISEN Toulon

Composition du Jury :

Président : Christian BROSSEAU Professeur, LABSTICC -Université Bretagne Occidentale
Examineurs : Khalifa DJEMAL Professeur, Université d'Evry-val d'Essonne - Université Paris-Saclay
Denis AHMAD Professeur émérite, LISIC-ULCO
Nesma SETTOUTI HDR, ISEN Yncréa ouest
Rachid JENNANE Professeur, L'université de Orleans – Polytech 'Orléans IDP Laboratory
Alan LOUSSERT Enseignant chercheur, ISEN Toulon

Dir. de thèse : Ayman AL-FALOU Professeur, ISEN Yncréa ouest

Invité(s)

Ehsan SEDGH GOOYA Enseignant Chercheur, ISEN Yncréa ouest
Mikael CABON Enseignant Chercheur, ISEN Yncréa ouest
Wissam KADDAH Enseignant Chercheur, ISEN Yncréa ouest

ACKNOWLEDGEMENT

I would like to thank the organisations and individuals for their help and support in completing this work.

I gratefully acknowledge the generous financial support provided by the College de France and the programme PAUSE. I am also very grateful to my supervisors, Dr. Ayman Al-falou, Dr. Ehsan Sedgh Gooya, and Dr. Mikael Cabon, who helped me in the preparation of this thesis. My special thanks to Dr. Sedgh Gooya who gave me great feedback, excellent encouragement and guidance.

To my family, I thank them for the tremendous support and hope they have given me. Without that hope, this thesis would have been much more difficult.

I would also like to thank the école doctoral de Mathstic and the University UBO (Univercité de Bretagne Occidentale) for all the support.

I appreciate Yncrea ISEN from Brest who was a nice host. And finally, I would like to thank my colleagues and some of the staff of Yncrea ISEN in Brest who provided a friendly atmosphere while working on this thesis.

RÉSUMÉ EN FRANÇAIS

Motivations

La mise en œuvre de divers types de projets dans différents domaines joue un rôle important dans le développement national et la croissance économique. En outre, tous les projets nécessitent une structure, un plan et un échéancier pour répondre aux exigences du projet et atteindre les objectifs fixés. Je me suis concentré sur le problème de resource-constrained project scheduling problem (RCPSP) car il fait partie de la gestion de la planification des projets.

Il existe divers algorithmes tels que les méthodes exactes, les heuristiques et les métaheuristiques pour traiter le problème susmentionné, mais ils ont des difficultés d'optimisation dans les projets réels. Après l'inefficacité de la méthode exacte pour résoudre les grands projets, les chercheurs ont utilisé des heuristiques et des métaheuristiques pour les grands problèmes, mais les métaheuristiques communes ne peuvent pas être assez efficaces étant donné les limitations de ces types d'algorithmes. Ces algorithmes tentent d'améliorer les solutions en générant des populations ou en recherchant les solutions locales pour les prochaines générations d'algorithmes, puisque ces algorithmes fonctionnent sur le principe de la recherche et de la répétition des générations. De plus, la solution optimale ou quasi-optimale que l'algorithme obtient dépend de la qualité des solutions initiales. De mon point de vue, la génération d'une séquence optimale ou quasi-optimale de tâches est la clé du bon algorithme.

Car il y a un manque d'étude du problème à l'aide de méthodes telles que les réseaux neuronaux et machine learning. Pour atteindre cet objectif, j'utilise des approches ayant la capacité d'apprendre, comme les réseaux neuronaux intégrés aux algorithmes. Par conséquent, dans ce travail, je me concentre sur les algorithmes qui bénéficient de deux types de réseaux neuronaux, appelés réseau neuronal multicouche (multi-layer neural network) et réseau neuronal convolutif (convolutional neural network). J'espère que ce travail pourra servir de base aux chercheurs qui souhaitent utiliser les réseaux neuronaux pour planifier des projets.

Objectifs

Cette thèse présente deux algorithmes qui bénéficient de méthodes adaptatives pour obtenir des résultats compétitifs. Par conséquent, dans ce travail, deux types différents de réseaux neuronaux, à savoir un réseau neuronal multicouche (multilayer neural network) et un réseau neuronal convolutif (convolutional neural network), sont intégrés dans deux algorithmes visant à minimiser la durée du projet en tenant compte des contraintes données.

Contributions

Le problème de planification de projet avec contraintes de ressources (the resource-constrained project scheduling problem (RCPSP)) comporte deux contraintes importantes, à savoir les contraintes de ressources et les relations de précédence des activités pendant la planification du projet. L'objectif du problème est d'optimiser et de minimiser la durée du projet. Dans ce travail, je développe deux approches différentes de réseaux neuronaux intégrées dans deux algorithmes pour résoudre le RCPSP standard. L'avantage de ces algorithmes par rapport aux méthodes conventionnelles telles que les métaheuristiques est qu'ils ne génèrent pas nécessairement de nombreuses solutions ou populations. Dans cette thèse, serial schedule generation scheme (SSGS) est utilisé pour planifier les activités du projet. Ici, les réseaux neuronaux évolués sont utilisés comme un outil pour sélectionner une règle de priorité appropriée pour filtrer une activité adéquate. Les réseaux neuronaux évolués apprennent en fonction des huit paramètres du projet, à savoir la complexité du réseau (network complexity), le facteur de ressources (resource factor), la force des ressources (resource strength), le travail moyen par activité (average work per activity), le pourcentage de travail restant (percentage of remaining work), le pourcentage d'activités non planifiées (percentage of unscheduled activities), le pourcentage de successeurs restants (percentage of remaining successors) et les unités moyennes par jour (average units per day). Les paramètres ci-dessus sont les entrées des réseaux et sont recalculés à chaque étape du planning du projet. En outre, onze règles de priorité (priority rules), EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, MSLK, GRPW et WRUP, sont les sorties des réseaux neuronaux développés. Ainsi, après les processus d'apprentissage, les réseaux neuronaux peuvent sélectionner automatiquement une règle de priorité pour filtrer une activité parmi les activités éligibles. Par conséquent, les algorithmes peuvent

planifier toutes les activités du projet en fonction des contraintes du projet.

Contenu du manuscrit

Comme nous l'avons mentionné précédemment, l'objectif du problème standard de planification de projet sous contrainte de ressources (the standard resource-constrained project scheduling problem (RCPSP)) est de minimiser la durée du projet donnée par l'équation 1. Le problème comporte deux contraintes principales, la contrainte de ressources indiquée dans l'équation 2 et la contrainte de précedence indiquée dans l'équation 4.

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

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

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

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

Pour réaliser cet objectif, j'utilise le serial schedule generation scheme (SSGS) pour planifier le projet. Une règle de priorité (priority rule) sélectionnée filtre une tâche éligible de l'ensemble des tâches éligibles à chaque étape de la planification du projet. Les tâches sont planifiées une par une selon le SSGS. La difficulté du problème devient apparente lorsqu'il y a plus d'une tâche éligible. De plus, il n'est possible de filtrer qu'une seule tâche à chaque étape de l'algorithme. Ainsi, s'il y a différentes règles de priorité (priority rule) et plus d'une tâche éligible, nous ne savons pas laquelle d'entre elles est la plus appropriée pour le planning.

Pour surmonter la difficulté mentionnée, je présente deux algorithmes différents qui tirent parti de deux approches de réseaux neuronaux différentes pour résoudre le prob-

lème standard de planification de projets sous contrainte de ressources (standard single-mode resource-constrained project scheduling problem (RCPSP)). Il s'agit de l'approche du réseau neuronal multicouche (MLNN) et de l'approche du réseau neuronal convolutif (CNN). La tâche de ces réseaux neuronaux est de sélectionner une règle de priorité (priority rule) en sortie à chaque étape de la planification du projet.

Les deux réseaux neuronaux développés sont alimentés par huit paramètres de projet : complexité du réseau (network Complexity (NC)), facteur de ressources (resource factor (RF)), force des ressources (resource strength (RS)), travail moyen par tâche (average work per activity), pourcentage de travail restant (percentage of remaining work), pourcentage de tâches non programmées (percentage of unscheduled activities), pourcentage de successeurs restants (percentage of remaining successors), et unités moyennes par jour (average units per day). Les huit paramètres sont recalculés pour caractériser la nouvelle étape ou le nouveau sous-projet.

Les sorties des réseaux neuronaux développés sont des règles, appelées règles de priorité (priority rules), qui sont utilisées pour sélectionner une activité éligible en fonction de ses critères de sélection pour la planification du projet. Ces règles de priorité sont les suivantes : heure de début précoce (early start time (EST)), heure de fin précoce (early finish time (EFT)), heure de début au plus tard (latest start time (LST)), heure de fin au plus tard (latest finish time (LFT)), temps de traitement le plus court (shortest processing time (SPT)), demande totale de ressources (total resource demand (TRD)), pénurie totale de ressources (total resource scarcity (TRS)), nombre total de successeurs (most total successors (MTS)), temps de latence (slack time (ST)), poids positionnel de rang le plus élevé (greatest rank positional weight (GRPW)) et rapport pondéré d'utilisation des ressources et de priorité (weighted resource utilization ration and precedence (WRUP)).

Deux réseaux neuronaux sont proposés pour faire face à la difficulté de sélectionner les tâches éligibles à chaque étape de la planification d'un projet. Ces réseaux neuronaux sont un réseau neuronal multicouche (multilayer feed-forward neural network (MLFNN)), illustré sur la figure 1, et un réseau neuronal convolutif 1D (convolutional neural network 1D), indiqué sur la figure 2. Les entrées de ces réseaux neuronaux sont les caractérisations paramétriques du projet (the project parametric characterizations) et les sorties sont des règles de priorité (priority rules). Les performances d'apprentissage du réseau MLFNN développé sont vérifiées avec une, deux et trois couches cachées dans 500, 1000 et 2000 époques, et avec trois, quatre, cinq, sept et onze règles de priorités comme sorties. Le réseau neuronal convolutif (CNN) développé est vérifié avec une, deux et trois couches

convolutives et trois couches entièrement connectées (fully connected) en 500, 1000 et 2000 époques, et avec trois, quatre, cinq, sept et onze règles de priorités en sortie.

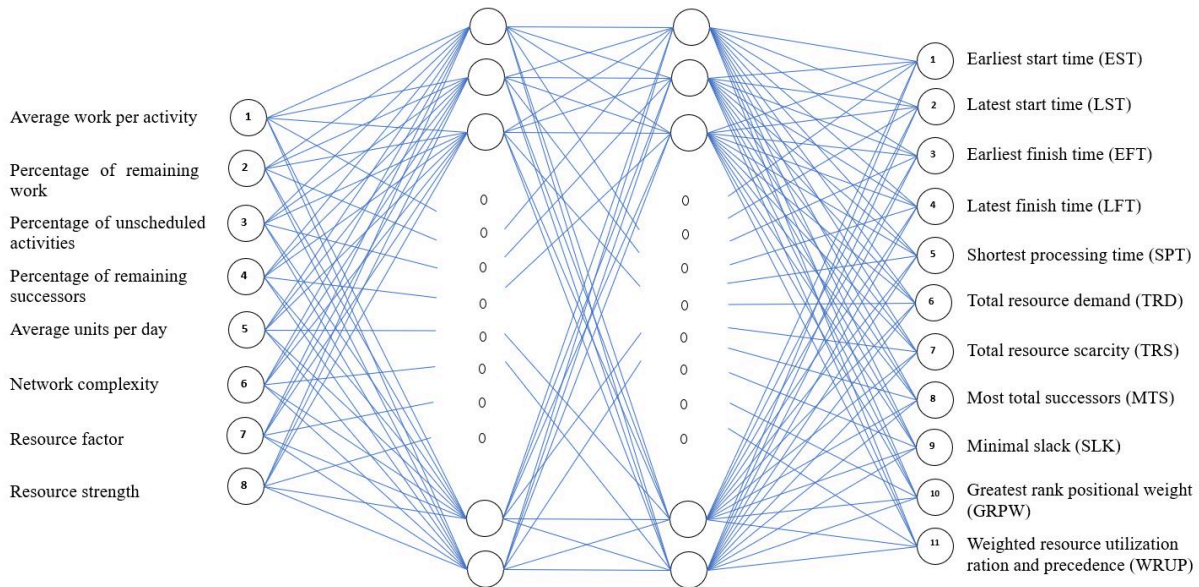


Figure 1 – Le réseau neuronal multicouche feed-forward (MLFNN) proposé se compose de huit entrées, de deux couches cachées et de onze sorties. Les sorties ou règles de priorité sont nommées EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, SLK, GRPW et WRUP.

Les résultats de la performance de l'apprentissage prouvent que les performances des réseaux neuronaux proposés augmentent lorsque le nombre de sorties est réduit.

Deux algorithmes développés sur la base du serial schedule generation scheme (SSGS) bénéficient séparément de deux réseaux neuronaux évolués. Le SSGS comprend n étapes et trois ensembles pendant la planification du projet, où n indique les étapes ou le nombre d'activités du projet, et les trois ensembles sont l'ensemble non planifié (unschedule set), l'ensemble éligible (eligible set) et l'ensemble planifié (scheduled set). La figure 3 présente le déroulement de l'algorithme.

Les instances standard comprenaient des projets avec quatre types de ressources renouvelables et 60 et 120 activités sélectionnées pour la planification. Les résultats sont présentés sous la forme d'un pourcentage moyen de déviations par rapport à la borne inférieure basée sur le chemin critique (average percentage of deviations from the lower bound based on the critical path) pour les instances de projet. Le tableau 1 présente les résultats compétitifs. Les résultats obtenus expliquent que le CNN développé est plus

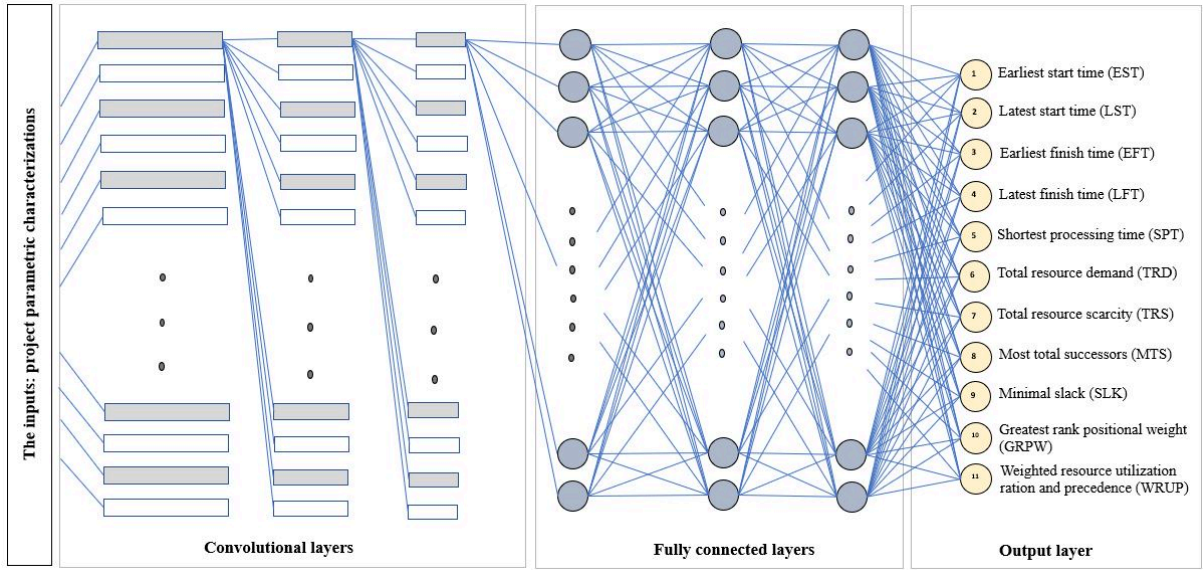


Figure 2 – Le réseau neuronal convolutif (CNN) proposé se compose de huit entrées comme couche d’entrée, de trois couches convolutives et de trois couches entièrement connectées, et de onze sorties. Les sorties ou règles de priorité sont nommées EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, SLK, GRPW et WRUP.

Règles de priorité utilisées pour la couche de sortie	Nombre de tâches			
	J60		J120	
	Approche			
	CNN Approche	MLFNN Approche	CNN Approche	MLFNN Approche
Trois règles de priorité sont utilisées en sorties: EST, LST, and EFT	16.57	15.97	38.39	37.77
Quatre règles de priorité sont utilisées en sorties: EST, LST, EFT and LFT	15.97	16.28	37.77	39.74
Cinq règles de priorité sont utilisées en sorties: EST, LST, EFT, LFT and MTS	16.19	47.04	39.48	89.61
Sept règles de priorité sont utilisées en sorties: EST, LST, EFT, LFT, MTS, TRD and PT	20.48	45.59	53.06	86.39
onze règles de priorité sont utilisées en sorties: EST, LST, EFT, LFT, MTS, TRD, PT, ST, TRS, GRPW, and WRUP	63.33	58.96	124.72	117.42

Table 1 – Pourcentage des déviations moyennes par rapport à la borne inférieure du chemin critique (Percentage of average deviations from critical path lower bound) pour le J60 et le J120. Les déviations moyennes obtenues par les algorithmes utilisant les approches CNN et MLFNN montrent que l’approche CNN est plus performante que l’approche MLFNN.

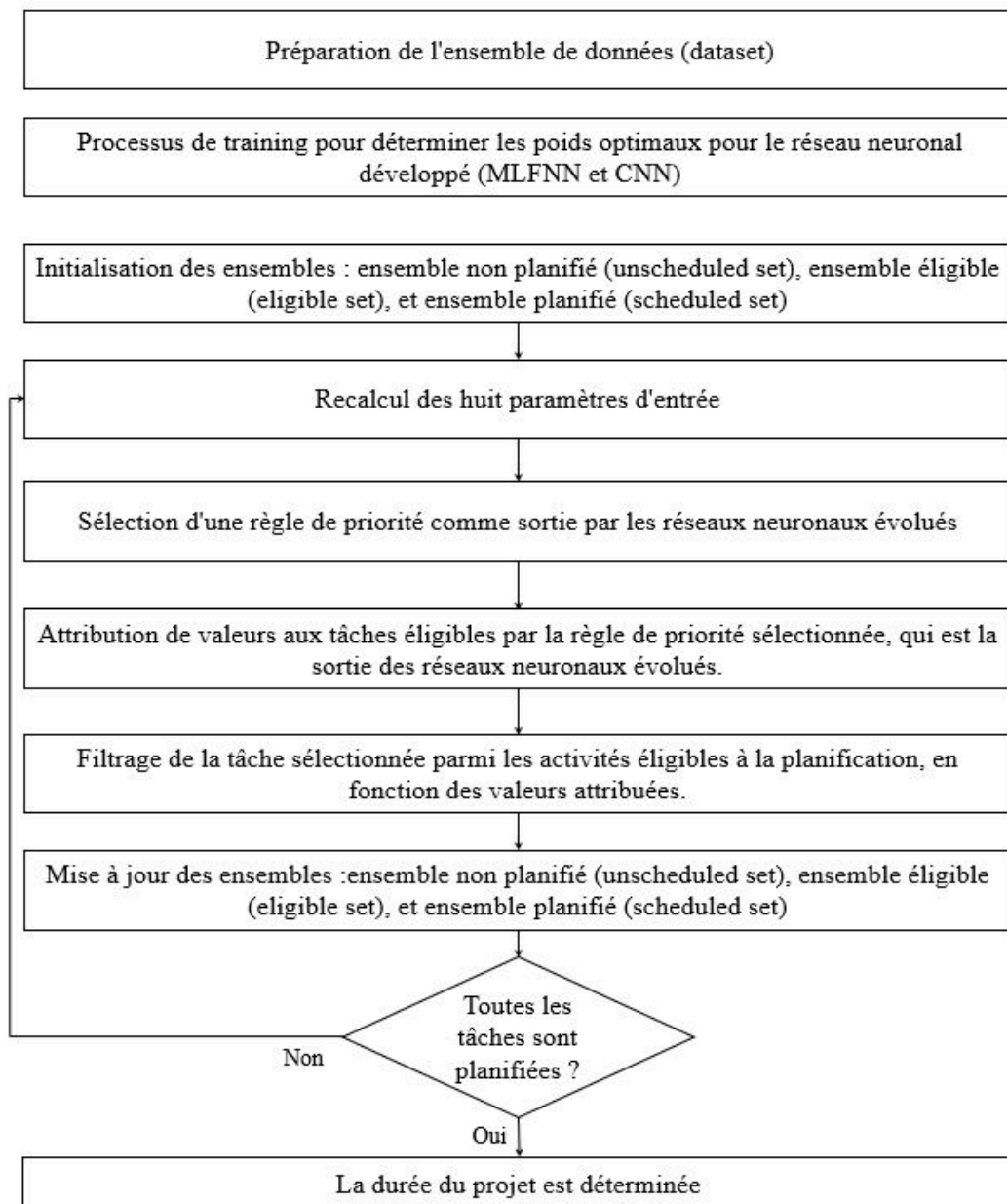


Figure 3 – La figure présente l'algorithme développé dans ce travail. Le processus commence par l'apprentissage sur la base de l'ensemble de données créé et se poursuit par la planification de toutes les tâches du projet afin de déterminer la durée finale du projet.

performant que le MLFNN développé. Les résultats résumés dans le tableau 1 expliquent également que les déviations moyennes de la borne inférieure du chemin critique sont meilleures lorsque les performances des réseaux neuronaux évolués augmentent. Il a été mentionné que les performances des réseaux neuronaux développés s'améliorent lorsque le nombre de sorties utilisées est réduit.

Les résultats obtenus avec les deux algorithmes proposés ne sont pas les meilleurs parmi ceux obtenus avec d'autres méthodes, mais les algorithmes proposés utilisent de nouvelles approches, et l'avantage important des algorithmes développés par rapport aux méthodes évolutionnaires ou méta-heuristiques est qu'il n'est pas nécessaire de générer des populations ou de répéter les itérations, au contraire, les algorithmes proposés génèrent une séquence d'activité en fonction du MLFNN et du CNN entraînés.

SUMMARY IN ENGLISH

Motivations

The implementation of various types of projects in different fields plays an important role in national development and economic growth. Moreover, all projects need a structure, plan and schedule to meet the project requirements and achieve the set goals. I focused on the topic of the resource-constrained project scheduling problem (RCPSP) because it is a part of project schedule management.

There are various algorithms such as exact methods, heuristics, and metaheuristics to deal with the aforementioned problem, but they have difficulties in optimization in real projects. After the ineffectiveness of the exact method to solve large RCPSP, researchers have used heuristics and metaheuristics for large problems, but the common metaheuristics cannot be efficient enough given the limitations of these types of algorithms. These algorithms try to improve the solutions by generating populations or searching the local solutions for the next generations of algorithms, since these algorithms work on the principle of searching and repeating generations. Moreover, the optimal or near-optimal solution that the algorithm achieves depends on the quality of the initial solutions. From my point of view, the generation of an optimal or near-optimal sequence of activities is the key to the right algorithm.

Since there is a lack of investigation of the problem using methods such as neural networks and machine learning. I employ approaches with the ability to learn, such as neural networks embedded in the algorithms, to achieve the objective. Therefore, in this work, I focus on the algorithms that benefit from two types of neural networks, called multilayer neural network and convolutional neural network. I hope that this work can be a basis for researchers who want to use neural networks to schedule projects.

Objectives

This thesis attempts to present two algorithms that benefit from adaptive methods to achieve competitive results. Therefore, in this work, two different types of neural networks,

namely a multilayer neural network and a convolutional neural network, are embedded in two algorithms to minimize the project duration considering the given constraints.

Contributions

The resource-constrained project scheduling problem (RCPSP) has two important constraints, namely resource constraints and precedence relationships of activities during project scheduling. The objective of the problem is to optimize and minimize the project duration. In this work, I develop two different neural network approaches embedded in two algorithms to solve the standard RCPSP. The advantage of these algorithms over conventional methods such as metaheuristics is that they do not necessarily generate many solutions or populations. In this thesis, the serial schedule generation scheme (SSGS) is used to schedule the project activities. Here, evolved neural networks are used as a tool to select an appropriate priority rule to filter out a suitable activity. The evolved neural networks learn according to the eight project parameters, namely network complexity, resource factor, resource strength, average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of remaining successors, and average units per day. The above parameters are the inputs of the networks and are recalculated at each step of the project schedule. In addition, eleven priority rules, EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, MSLK, GRPW and WRUP, are the outputs of the developed neural networks. Therefore, after the learning processes, the neural networks can automatically select a priority rule to filter an activity from the eligible activities. Consequently, the algorithms can schedule all project activities according to the given project constraints.

Content of the manuscript

As mentioned earlier, the objective of the standard resource-constrained project scheduling problem (RCPSP) is to minimize the project duration given in Equation 5. The problem has two main constraints, the resource constraint stated in Equation 6 and the precedence constraint stated in Equation 8.

$$\min f_n \tag{5}$$

$$f_i \leq f_j - d_j \quad \forall (A_i, A_j) \in Pred \quad (6)$$

$$f_1 = 0, d_1 = 0, d_n = 0 \quad \text{the activities 1 and n are dummies or milestones} \quad (7)$$

$$\sum_{i \in P_t} u_{irt} \leq U_r \quad \forall t = 1, \dots, f_n \quad \text{and} \quad \forall r \in R \quad (8)$$

To achieve the objective, I employ the serial schedule generation scheme (SSGS) to schedule the project. A selected priority rule filters out an eligible task from the eligible set at each step of the project schedule. The activities are scheduled one by one according to the SSGS. The difficulty of the problem becomes apparent when there is more than one eligible activity. Also, it is only possible to filter out one activity at each step of the algorithm. Thus, if there are different priority rules and more than one eligible activity, we do not know which of them is more suitable for scheduling.

To overcome the mentioned difficulty, I present two different algorithms that benefit from two different neural network approaches to solve the standard single-mode resource-constrained project scheduling problem (RCPSP). These are the multilayer feed-forward neural network (MLFNN) approach and the convolutional neural network (CNN) approach. The task of these neural networks is to select a priority rule as output at each step of project scheduling.

The two neural networks developed are fed with eight project parameters: network Complexity (NC), resource factor (RF), resource strength (RS), average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of remaining successors, and average units per day. The eight parameters are recalculated to characterize the new step or subproject.

The outputs of the developed neural networks are rules, called priority rules, which are used to select an eligible activity according to its selection criteria for project scheduling. These priority rules are called early start time (EST), early finish time (EFT), latest start

time (LST), latest finish time (LFT), shortest processing time (SPT), total resource demand (TRD), total resource scarcity (TRS), most total successors (MTS), slack time (ST), greatest rank positional weight (GRPW), and weighted resource utilization ration and precedence (WRUP).

Two neural networks are proposed to deal with the difficulty of selecting eligible activities at each step of project scheduling. The neural networks are a multilayer feed-forward neural network (MLFNN), shown in Figure 4, and a convolutional neural network 1D, shown in Figure 5. The inputs of these neural networks are the project parametric characterizations and the outputs are priority rules. The training performance of the developed MLFNN is verified with one, two, and three hidden layers in 500, 1000, and 2000 epochs, and with three, four, five, seven, and eleven priority rules as outputs. The developed convolutional neural network (CNN) was verified with one, two, and three convolutional layers and three fully connected layers in 500, 1000, and 2000 epochs, and with three, four, five, seven, and eleven priority rules as outputs.

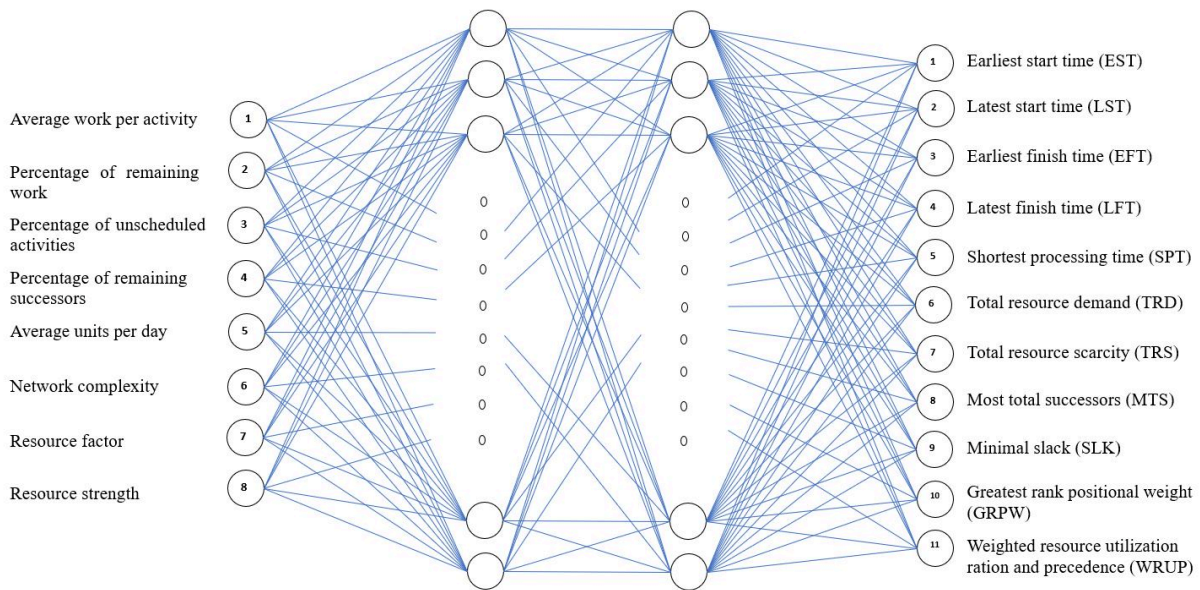


Figure 4 – The proposed multilayer feed-forward neural network (MLFNN) consists of eight inputs, two hidden layers, and eleven outputs. The outputs or priority rules are named EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, SLK, GRPW, and WRUP.

The training performance results prove that the performances of the proposed neural networks increase when the number of outputs is reduced.

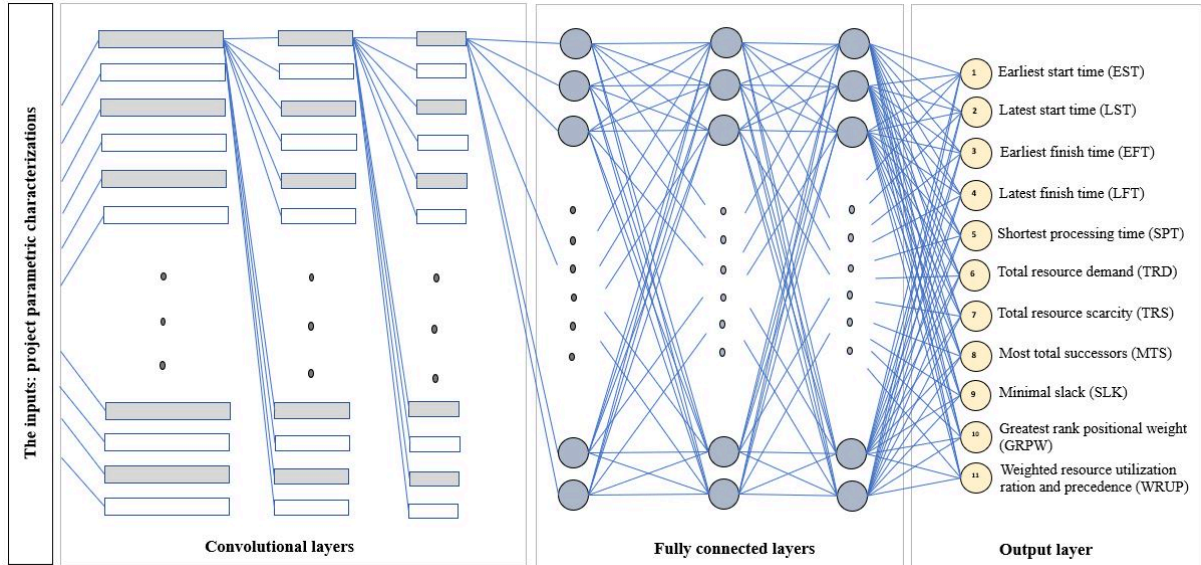


Figure 5 – The proposed convolutional neural network (CNN) consists of eight inputs as input layer, three convolutional layers and three fully connected layers, and eleven outputs. The outputs or priority rules are named EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, SLK, GRPW, and WRUP.

Priority rules used for output layer	Number of activities			
	J60		J120	
	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, ST, TRS, GRPW, and WRUP	63.33	58.96	124.72	117.42

Table 2 – Percentage of average deviations from critical path lower bound for the J60 and J120. The average deviations obtained by the algorithms using the CNN and MLFNN approaches show that the CNN approach performs better compared to the MLFNN approach.

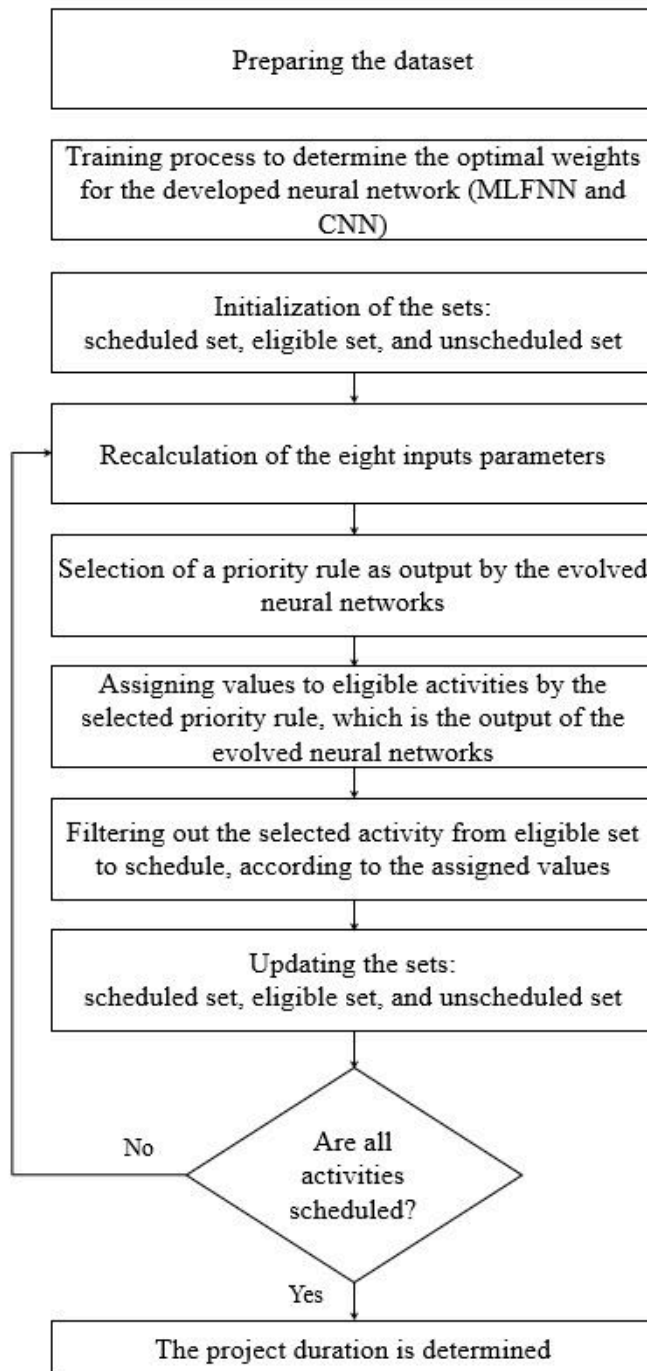


Figure 6 – The flowchart shows the algorithm developed in this work. The process starts with training based on the created dataset and continues with scheduling all project activities to determine the final project duration.

Two algorithms developed based on the serial schedule generation scheme (SSGS) benefit separately from two evolved neural networks. The SSGS includes n steps and three sets during the project schedule, where n indicates the steps or the number of project activities, and the three sets are schedule set, eligible set, and unscheduled set. Figure 6 shows the flow of the algorithm.

The standard instances included projects with four types of renewable resources and 60 and 120 activities selected for scheduling. 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. Table 2 presents the competitive results. The obtained results explain that the developed CNN performs better than the developed MLFNN. The results summarized in Table 2 also show that the average deviations from the critical path lower bound are better when the performances of the evolved neural networks increase. It was mentioned that the performance of the developed neural networks improves when the number of outputs used is reduced.

The results obtained with the two proposed algorithms are not the best among those obtained with other methods, but the proposed algorithms use new approaches, and the important advantage of the developed algorithms over evolutionary methods or meta-heuristics is that it is not necessary to generate populations or repeating the iterations, on the contrary, the proposed algorithms generate an activity sequence according to the trained MLFNN and CNN.

TABLE OF CONTENTS

List of acronyms	23
list of figures	26
list of tables	27
Introduction	29
Basic contents	29
Resource-constrained project scheduling problem	33
A practical example of the problem	35
The conclusion and the objective of the thesis	49
1 literature review	51
1.1 Classification of published RCPSP articles	51
1.2 developed conventional Meta-heuristics to solve the RCPSP	52
1.2.1 RCPSP and genetic algorithms (GA)	52
1.2.2 RCPSP and particle swarm optimization (PSO)	58
1.2.3 RCPSP and ant colony optimization (ACO)	61
1.2.4 RCPSP and bees colony optimization (BCO)	63
1.2.5 RCPSP and simulated annealing (SA)	64
1.2.6 RCPSP and tabu search (TS)	65
1.2.7 RCPSP and teaching-learning-based optimization (TLBO)	66
1.2.8 RCPSP and evolutionary algorithms (EA)	67
1.2.9 RCPSP and hybrid algorithms (HA)	68
1.2.10 RCPSP and the proposed algorithms based on other metaheuristics	70
1.2.11 RCPSP and the proposed algorithms based on reinforcement learn- ing	71
1.2.12 RCPSP and the proposed algorithms based on neural networks . . .	72
1.3 Conclusion	73

TABLE OF CONTENTS

2	Overview of neural networks, inputs and outputs	75
2.1	Overview of neural networks	75
2.2	The inputs: project parametric characterizations	80
2.3	The outputs: Priority rules	82
2.4	Conclusion	85
3	A multilayer feed-forward neural network (MLFNN) to solve the RCPSP	87
3.1	Algorithm and the developed MLFNN	88
3.2	Computational analysis	95
3.2.1	MLFNN training performance results and discussion	95
3.2.2	Comparative results and discussion	97
3.3	Conclusion	102
4	A convolutional neural network (CNN) to solve the RCPSP	103
4.1	Algorithm and the developed CNN	104
4.2	Computational analysis	110
4.2.1	The convolutional neural network training performance results and discussion	110
4.2.2	Comparative results and discussion	113
4.3	Conclusion	117
	Conclusion	119
	The problem and related works	119
	Employing the two neural networks to project schedule	120
	Bibliography	123
	List of publications	137

LIST OF ACRONYMS

AoN	Activity-on-Node
AoA	Activity-on-Arc
ACO	Ant Colony Optimization
ABC	Artificial Bee Colony
AWA	Average Work per Activity
AUD	Average Units per Day
ANN	Artificial Neural Network
BCO	Bees Colony Optimization
CPM	Critical Path Method
CNN	Convolutional Neural Network
DEA	Distribution Estimation Algorithm
DL	Deep Learning
EST	Earliest Start Time
EFT	Earliest Finish Time
EA	Evolutionary algorithm
GA	Genetic Algorithm
GRPW	Greatest Rank Positional Weight
HA	Hybrid Algorithm
HAntCO	Hybrid ant Colony Optimization
LST	Latest Start Time
LFT	Latest Finish Time
MTS	Most Total Successors
MSLK	Minimal Slack
MLFNN	Multilayer Feed-Forward Neural Network
ML	Machine Learning
NN	Neural Networks
NC	Network Complexity

PSPLIB	Project Schedule Problems Library
PSGS	Parallel Schedule Generation Scheme
PSO	Particle Swarm Optimization
PRW	Percentage of Remaining Work
PUA	Percentage of Unscheduled Activities
PRS	Percentage of Remaining Successors
RCPSP	Resource-Constrained Project Scheduling Problem
RL	Reinforcement Learning
RF	Resource Factor
RS	Resource Strength
SPT	Shortest Processing Time
SGS	Schedule Generation Scheme
SSGS	Serial Schedule Generation Scheme
SA	Simulated Annealing
TRD	Total Resource Demand
TRS	Total Resource Scarcity
TS	Tabu Search
TLBO	Teaching-Learning-Based Optimization
WRUP	Weighted Resource Utilization Ration and Precedence

LIST OF FIGURES

1	The proposed MLFNN with 11 outputs	9
2	The proposed CNN with 11 outputs	10
3	The flowchart of the proposed algorithms	11
4	The proposed MLFNN with 11 outputs	16
5	The proposed CNN with 11 outputs	17
6	The flowchart of the developed algorithms	18
7	A project representing activity on node diagram	31
8	An practical example	37
9	Step 1 of the project scheduling	38
10	Step 2 of the project scheduling	39
11	Step 3 of the project scheduling	40
12	Step 4 of the project scheduling	41
13	Step 5 of the project scheduling	42
14	Step 6 of the project scheduling	43
15	Step 7 of the project scheduling	44
16	Step 8 of the project scheduling	45
17	Step 9 of the project scheduling	46
18	Step 10 of the project scheduling	47
19	Step 11 of the project scheduling	48
2.1	An example of multi-layer neural network	78
2.2	An example of convolutional neural network	79
3.1	The developed MLFNN with 11 outputs	90
3.2	The developed MLFNN with 7 outputs	91
3.3	The developed MLFNN with 5 outputs	92
3.4	The developed MLFNN with 4 outputs	93
3.5	The developed MLFNN with 3 outputs	93
3.6	The flowchart of the algorithm	94

LIST OF FIGURES

4.1	The flowchart of the algorithm	106
4.2	The developed CNN with 11 outputs	107
4.3	The developed CNN with 7 outputs	108
4.4	The developed CNN with 5 outputs	108
4.5	The developed CNN with 4 outputs	109
4.6	The developed CNN with 3 outputs	109

LIST OF TABLES

1	The comparative results obtained by two proposed algorithms	10
2	The comparative results obtained by two developed algorithms	17
2.1	The definitions of the employed items in the formulas' parameters	81
2.2	The priority rules and the selection criteria	85
3.1	Training performance of the developed MLFNN with one hidden layer . . .	96
3.2	Training performance of the developed MLFNN with two hidden layers . .	96
3.3	Training performance of the developed MLFNN with three hidden layers .	97
3.4	The results of average deviations from critical path lower bound for the J60 and J120	99
3.5	Percentage of average deviations from critical path lower bound for the J60	100
3.6	Percentage of average deviations from critical path lower bound for the J120	101
4.1	Training performance of the developed CNN with one convolutional layer . .	111
4.2	Training performance of the developed CNN with two convolutional layers .	112
4.3	Training performance of the developed CNN with three convolutional layers	112
4.4	The results of average deviations from critical path lower bound for the J60 and J120	114
4.5	Percentage of average deviations from critical path lower bound for the J60 using CNN	115
4.6	Percentage of average deviations from critical path lower bound for the J120 using CNN	116
4.7	The comparative results of average deviations from critical path lower bound for the J60 and J120	122

INTRODUCTION

The topic of the thesis is the automation of resource-constrained project scheduling by artificial intelligence. Resource-constrained project scheduling is a well-known problem, the resource-constrained project scheduling problem (RCPSP). The problem is a part of schedule management, which is one of the knowledge areas of project management.

Basic contents

Since the subject is related to project management. We should know some explanations, structures, and definitions about project management and the problems. **project definition :**

project definition

A project is defined as a temporary endeavor or attempt undertaken to create a unique product, service, or result.

The projects are carried out to achieve the goals and achieve results. And the implementation of the project objectives may produce one or more results, such as a unique product, service, outcome, or combination of products, services, or results.

Some examples of projects:

- Merging two or some organizations
- Constructing infrastructures
- Improving business process
- Producing or modifying software or hardware
- Developing market
- Conducting research
- Developing a tour guide service
- And ...

It was mentioned that a project is a temporary endeavor, which means that a project has a definite start and end. However, the results of the projects can be a process or the

processes after the end of the project. A project can also be terminated for a variety of reasons, such as resources being exhausted, the need for the project no longer exists, the objective cannot be achieved, and other reasons.^{1 2}.

A project can be represented academically by two project schedule network diagrams called activity-on-node (AoN) and activity-on-arc (AoA). We use the activity-on-node (AoN) diagram to explain the structure of a project. The network diagram shows the order in which activities should be scheduled to account for logical precedencies between them. It usually consists of nodes representing activities and arrows representing relationships between the activities^{3 4}.

A project needs a framework to achieve the defined objectives called project management.

project management definition:

Project management is defined as the application of knowledge tools, skills, and techniques to project activities to meet project requirements.

project management refers to managing the work of a project to achieve results.⁵. Project management plays a fundamental role in ensuring that projects achieve their planned strategic objectives, promote economic growth, grow businesses and create value, or determine infrastructure. Thus, project management plays an important role in national development and economic improvement^{6 7}.

project management enables organizations to execute projects efficiently and effectively. Lack of project management or ineptly managed projects can lead to cost over-

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

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

3. Amir Golab et al., « Review of conventional metaheuristic techniques for resource-constrained project scheduling problem », *in: Journal of Project Management 7.2* (2022), pp. 95–110.

4. Georgios Koulinas, Lazaros Kotsikas, and Konstantinos Anagnostopoulos, « A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem », *in: Information Sciences 277* (2014), pp. 680–693.

5. PMI, *op. cit.*

6. Amir Golab et al., « Investigating the performance of an artificial neural network for solving the resource constrained project scheduling problem (RCPSp) », *in: Pattern Recognition and Tracking XXXIII*, vol. 12101, SPIE, 2022, pp. 78–83.

7. Farhad Habibi, Farnaz Barzinpour, and Seyed Sadjadi, « Resource-constrained project scheduling problem: review of past and recent developments », *in: Journal of project management 3.2* (2018), pp. 55–88.

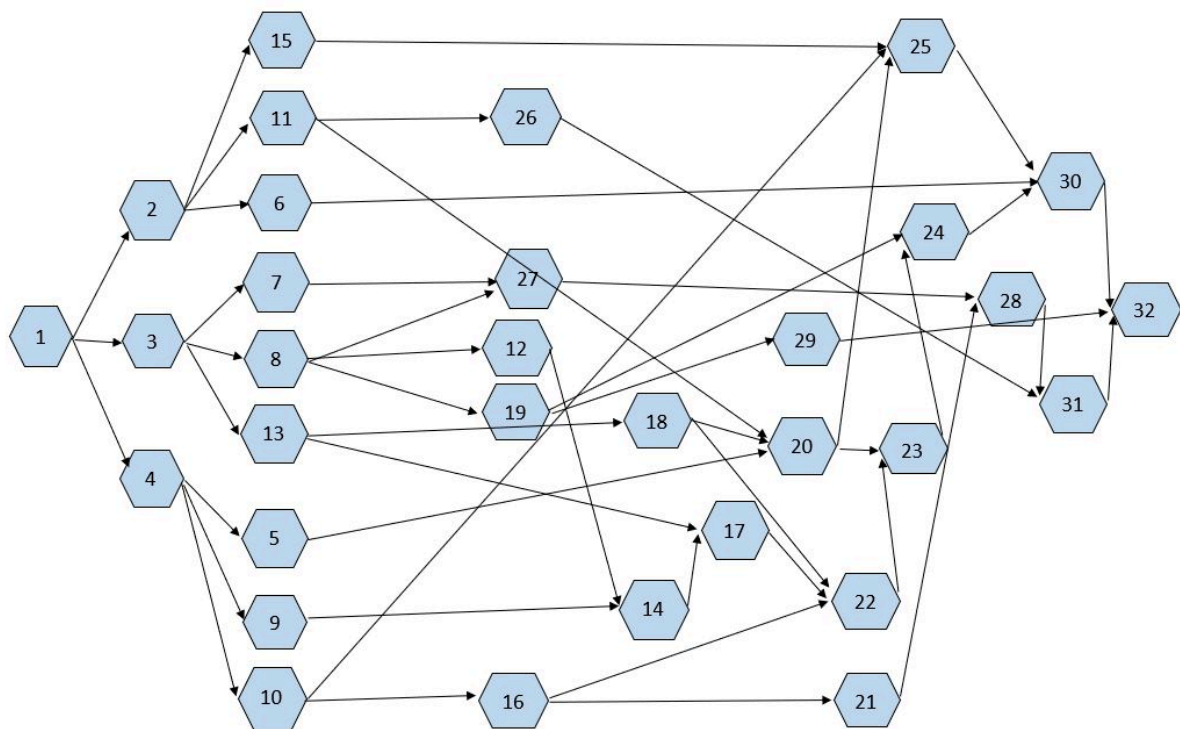


Figure 7 – An example showing an Activity-on-Arrow (AoA) network representing a project with 32 activities. Activities 1 and 32 are milestones or dummy activities with a duration of zero. The example project was selected from the PSPLIB.

runs, rework, uncontrolled expansion of project scope, dissatisfied stakeholders, missed deadlines, and other unpleasant consequences. On the other hand, effective project management helps organizations achieve pleasant outcomes such as achieving business objectives, better predictability, solving problems, and managing constraints such as resources and other pleasant outcomes. Management of constraints such as renewable resources is a focus of our research. Effective project management should be a weighted strategic competency within organizations that enables organizations to achieve business goals and remain competitive in the marketplace, as well as meet other pleasant expectations⁸.

Project management is composed of various knowledge areas, such as project integration management, project scope management, project schedule management, project risk management, project cost management, project quality management, project resource management, project communication management, project procurement management and project stakeholder management. In the thesis, we focus on the knowledge area of project schedule management.

As mentioned earlier, project schedule management is one of the knowledge 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 schedule. The sequence activities process is used to determine the relationships between project tasks. The importance of this process is that it defines or determines the logical sequence of work or activities to achieve the greatest possible efficiency, given all the constraints of the project. The sequence of project activities can be realised by software or by using manual or automated techniques. In this work, we have tried to define the sequence of activities using automated techniques.

The develop schedule process is the analysis of the sequence of activities, duration, resource requirements, and schedule constraints to create the project schedule for project execution. The goal of this process is to create a schedule model, or the identification and sequencing of activities with planned dates for the execution of project tasks or project work. The schedule model determines the planned start and finish dates for project tasks and milestones based on available information⁹.

8. PMI, *A guide to the project management body of knowledge, PMBOK GUIDE Sixth Edition*.

9. *Ibid*.

Resource-constrained project scheduling problem

The Resource-constrained Project Scheduling Problem (RCPSP), which is the problem under consideration in this thesis, is academically similar to the introduced processes called sequence activities and develop schedule to obtain a project plan. The RCPSP focuses on creating an activity sequence to optimize and minimize project duration while respecting project constraints, i.e., priority relationships between activities and resource constraints^{10 11}.

The final objective:

the objective of the RCPSP

The ultimate objective of the problem is to optimize and minimize the project duration considering the resource constraints.

The RCPSP is defined by the set $A = 1, \dots, i$ of activities constrained by two types of constraints, called precedence relations or relations through activities and resource constraints, in particular renewable resources.

Equation 9 states the objective or purpose of the problem, which is to optimize or minimize the project duration given the two main constraints. Equation 10 ensures that all precedence constraints or all relationships between activities are satisfied during the project schedule. It states that activity i cannot be started until its immediate predecessors are completed. Equation 11 illustrates that the first and last tasks or activities are dummy activities with zero duration, i.e., milestones. Moreover, during the execution of the project, there are a set of renewable resources $R = 1, \dots, r$, and each activity requires u_{irt} units per time for execution. Therefore, Equation 12 states that the second respectable constraint of the RCPSP is to consider the available resource quantities period by period

10. Golab et al., *op. cit.*

11. Golab et al., « Review of conventional metaheuristic techniques for resource-constrained project scheduling problem ».

to ensure the feasibility of the resources during the project schedule^{12 13 14 15 16 17 18 19}.

$$\min f_n \tag{9}$$

$$f_i \leq f_j - d_j \quad \forall (A_i, A_j) \in Pred \tag{10}$$

$$f_1 = 0, d_1 = 0, d_n = 0 \quad \text{the activities 1 and n are dummies or milestones} \tag{11}$$

$$\sum_{i \in P_t} u_{irt} \leq U_r \quad \forall t = 1, \dots, f_n \quad \text{and} \quad \forall r \in R \tag{12}$$

The followings lines present the defined elements:

- The set A consists of the project activities with duration d_i and $i = 1, 2, 3, \dots, n$. Activities 1 and n are dummy.
- The set of $R = 1, \dots, r$ represents the renewable resources.
- U_r represents the available quantities of renewable resource r
- f_i represents the finish time of activity i

12. KLEIN Bouleimen and HOUSNI Lecocq, « A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version », *in: European journal of operational research* 149.2 (2003), pp. 268–281.

13. Rainer Kolisch and Sönke Hartmann, « Experimental investigation of heuristics for resource-constrained project scheduling: An update », *in: European journal of operational research* 174.1 (2006), pp. 23–37.

14. Koulinas, Kotsikas, and Anagnostopoulos, *op. cit.*

15. Bidisha Roy and Asim Kumar Sen, « Meta-heuristic techniques to solve resource-constrained project scheduling problem », *in: International conference on innovative computing and communications*, Springer, 2019, pp. 93–99.

16. Golab et al., « Investigating the performance of an artificial neural network for solving the resource constrained project scheduling problem (RCPSP) ».

17. Golab et al., « Review of conventional metaheuristic techniques for resource-constrained project scheduling problem ».

18. A Golab et al., « A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP) », *in: Decision Science Letters* 11.4 (2022), pp. 407–418.

19. A Golab et al., « A convolutional neural network for the resource-constrained project scheduling problem (RCPSP): A new approach », *in: Decision Science Letters* 12.2 (2023), pp. 225–238.

-
- f_j represents the start time of activity j , which is the immediate successor of activity i .
 - 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} represents the amount of renewable resource r consumed by activity i in the period t .

A practical example of the problem

In this section the problem is explained by means of an example. In this way, the problem becomes clear. The elements of the example are shown in figure 8. The figure consists of the elements highlighted by the alphabet, which are presented below.

- A shows an activity-on-node (AoN) network diagram representing a project with 11 activities. Activities 1 and 11 are milestones or dummy activities with zero duration.
- B is the guide that represents a node with elements around which are ID, duration, work, and unit/day. They indicate the activity identifier, the duration required to perform the activity, the total work content required to perform the activity, and the renewable resources per day required to perform the activity.
- C presents the list of IDs of the project. and R mentions that for this example there is only one renewable resource, limited to 8h/day. So the resource cannot be allocated to activities more than 8 hours per day.
- D presents three priority rules and their selection criteria. These priority rules are selected for this example. They are shortest processing time (SPT), most total successors (MTS), and total resource demand (TRD).
- E shows the work content boxes associated with the activities. They are scheduled in figure G.
- F consists of three lists: the list of unscheduled activities, the list of eligible activities, which indicates activities that are eligible for scheduling, and the last list is the list of scheduled activities. All lists should be updated during project schedule.
- G is a time diagram. On the chart there are two axes, one horizontal and one vertical, representing the time axis and the unit axis.

For scheduling the example project, we use the serial schedule generation scheme (SSGS) method. This method will be explained in the next chapters. In summary, this method schedules the selected activities from the list of eligible activities and inserts them into the schedule step by step, according to the constraints. We schedule the project step by step to illustrate the difficulty of the problem. Moreover, the goal of scheduling is to minimise the project duration.

Step 1: We should respect the precedence relations between activities, which is one of the constraints of the problem, as I explained in the previous section. So, there is only activity 1 in the list of eligible activities, as shown in Figure 9. Then, activity 1 can be scheduled or included in the schedule according to the SSGS method²⁰. At the end of each step, the lists must be updated.

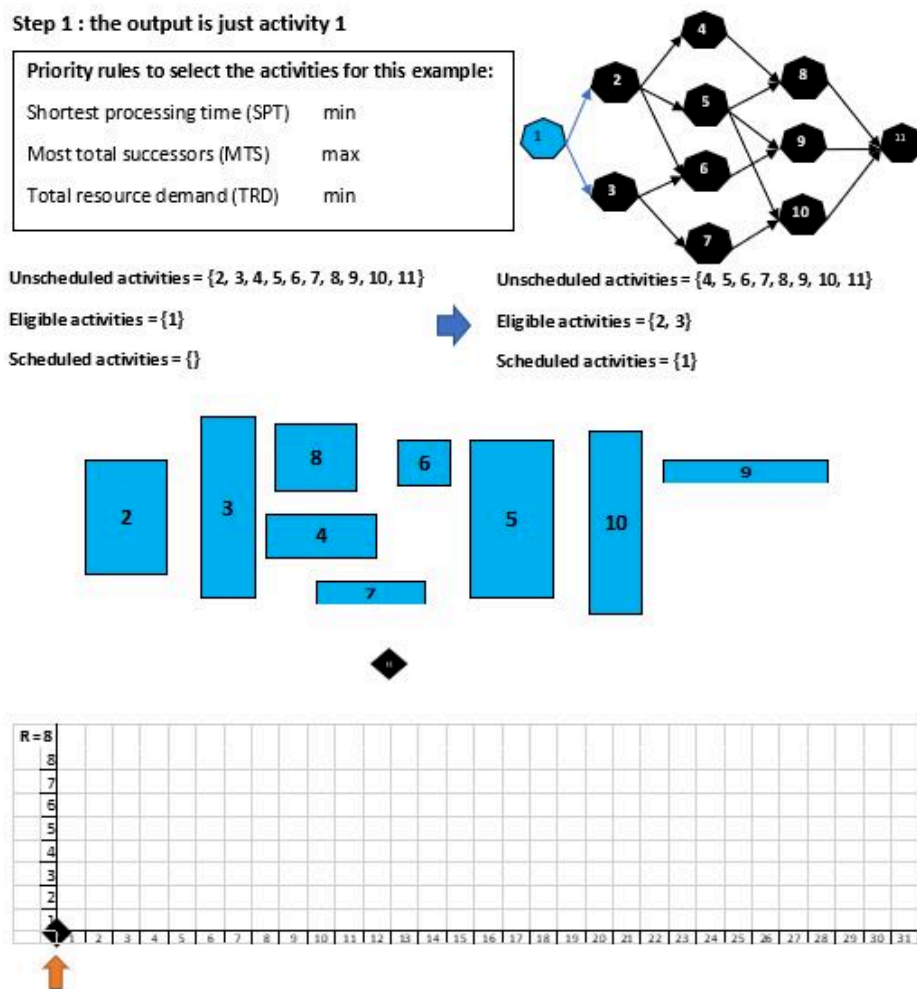


Figure 9 – Step 1 of project scheduling: activity 1 is selected for planning on the time chart, taking into account the precedence relationships between the activities.

20. Serial Schedule Generation Scheme

Step 2: Activity 1 was scheduled in the previous step. Considering the first constraint, the precedence relationship between activities, there are two activities 2 and 3 in the list of eligible activities, as shown in Figure 10. We should select one of them to schedule according to the resource constraint. Therefore, the priority rules are used (the priority rules are explained in detail in chapter 2) to select one of the eligible activities. There are 3 different priority rules, as shown in Figure 10. The difficulty of the problem is that we do not know which of the priority rules is more appropriate to select an activity from the list of eligible activities. If we choose the rule of shortest processing time (SPT), this rule filters out activity 3. If we choose the rule with the most total successors (MTS), activity 2 will be filtered out by the chosen priority rule. In this step we select the shortest processing time without any special logic.

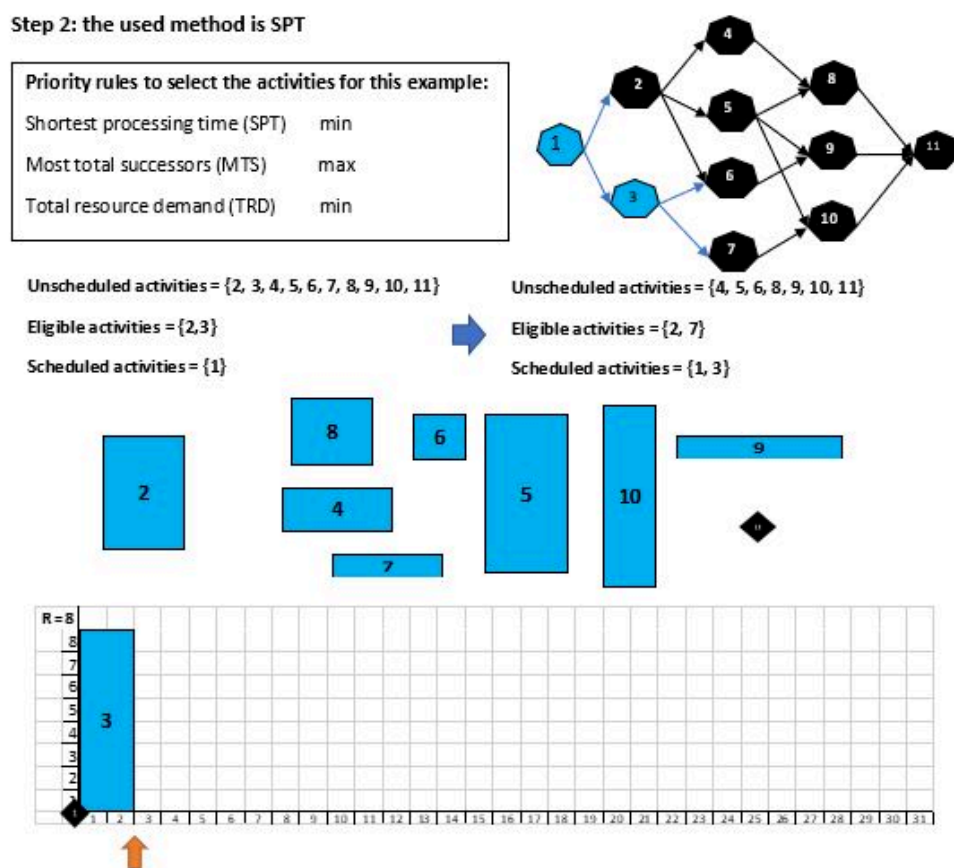


Figure 10 – Step 2 of the project scheduling: Task 3 is filtered out to schedule on the time chart according to shortest processing time (SPT).

Step 3: There are two activities 2 and 7 in the list of eligible activities, given the first constraint, which is the precedence relationships between the activities, as shown in Figure 11. We should select one of them to schedule it according to the resource constraint as in the previous step. Also, there are 3 different priority rules as shown in Figure 11. The difficulty in this step is that we do not know which of the priority rules is more appropriate to select an activity from the list of eligible activities. In this step, we choose the shortest processing time to select the activity from the list of eligible activities without by chance.

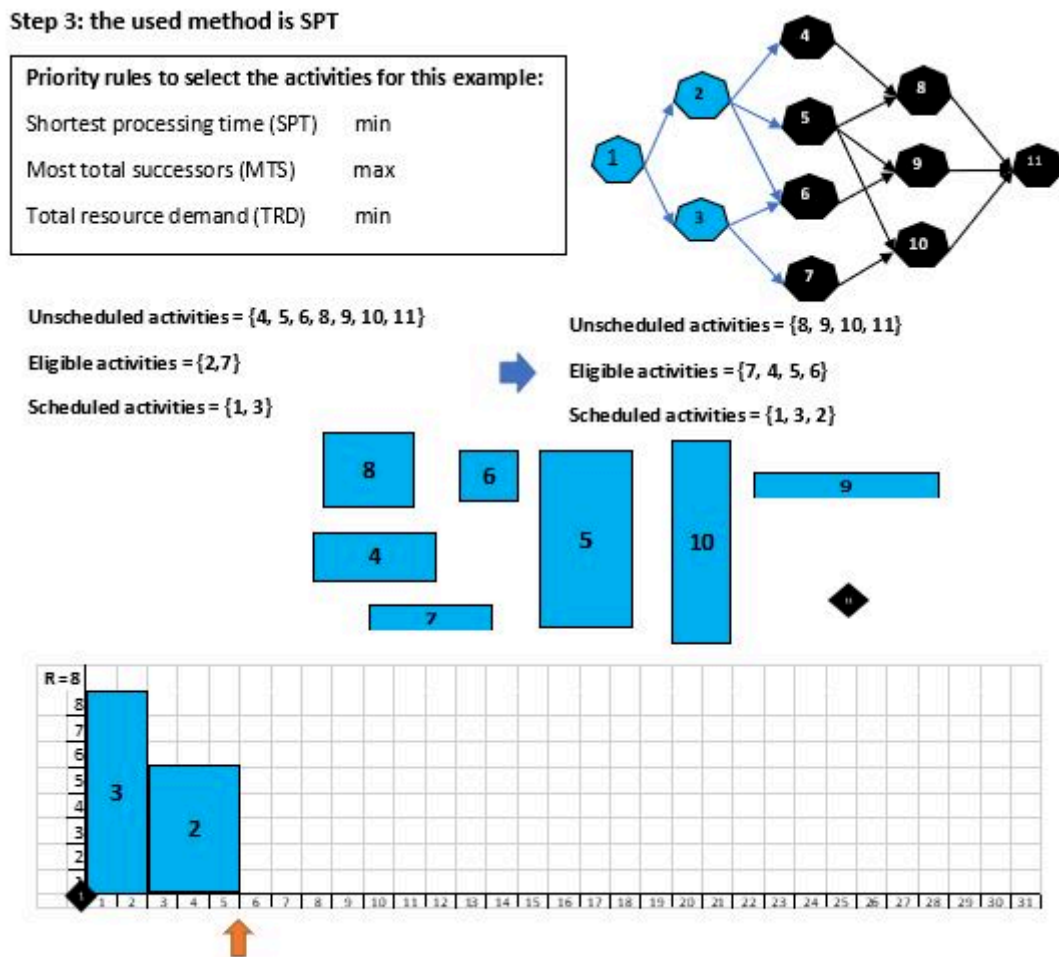


Figure 11 – Step 3 of project scheduling: task 2 is filtered out to schedule on the time chart according to the shortest processing time (SPT).

Step 4: After project scheduling, activities 7, 4, 5 and 6 are on the list of eligible list. According to the procedure, one of the activities should be selected by one of the 3 priority rules for planning. The difficulty in this step is that we do not know which of the priority rules is best for selecting an activity. In this step, we select most of the successors without any special logic.

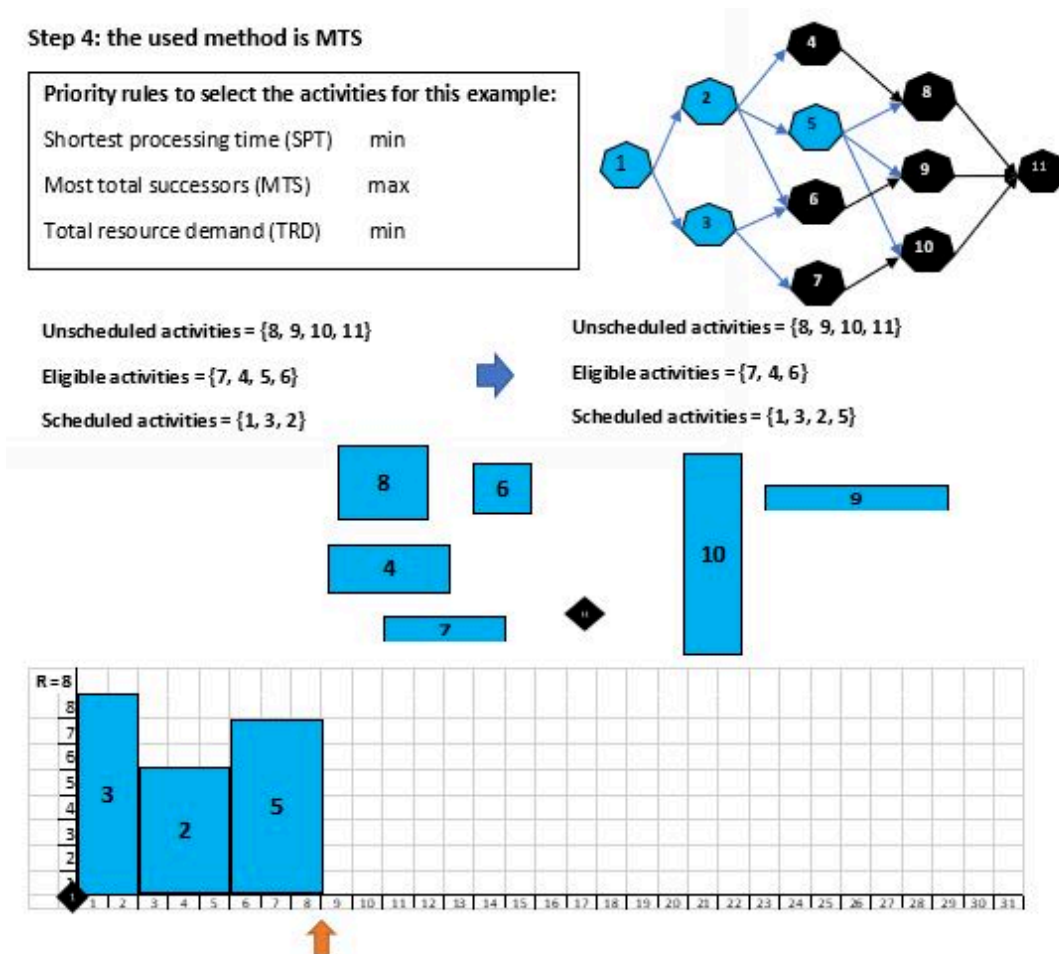


Figure 12 – Step 4 of project scheduling: task 5 is filtered out to schedule on the time chart by most total successors (MTS).

Step 5: Activity 5 was scheduled in the previous step. Thus, activities 7, 4, and 6 are eligible given the first constraint, which is the precedence relationships between activities, as shown in Figure 13. As in the previous steps, we need to select an appropriate priority rule that filters out an eligible activity. The difficulty in this step is choosing the appropriate priority rule. If we use most total successors (MTS), all three activities are eligible. If the shortest processing time (SPT) is chosen, activity 6 is filtered out, and if we choose total resource demand (TRD), activities 6 and 7 are filtered out. In this step, we select the shortest processing time without any special logic.

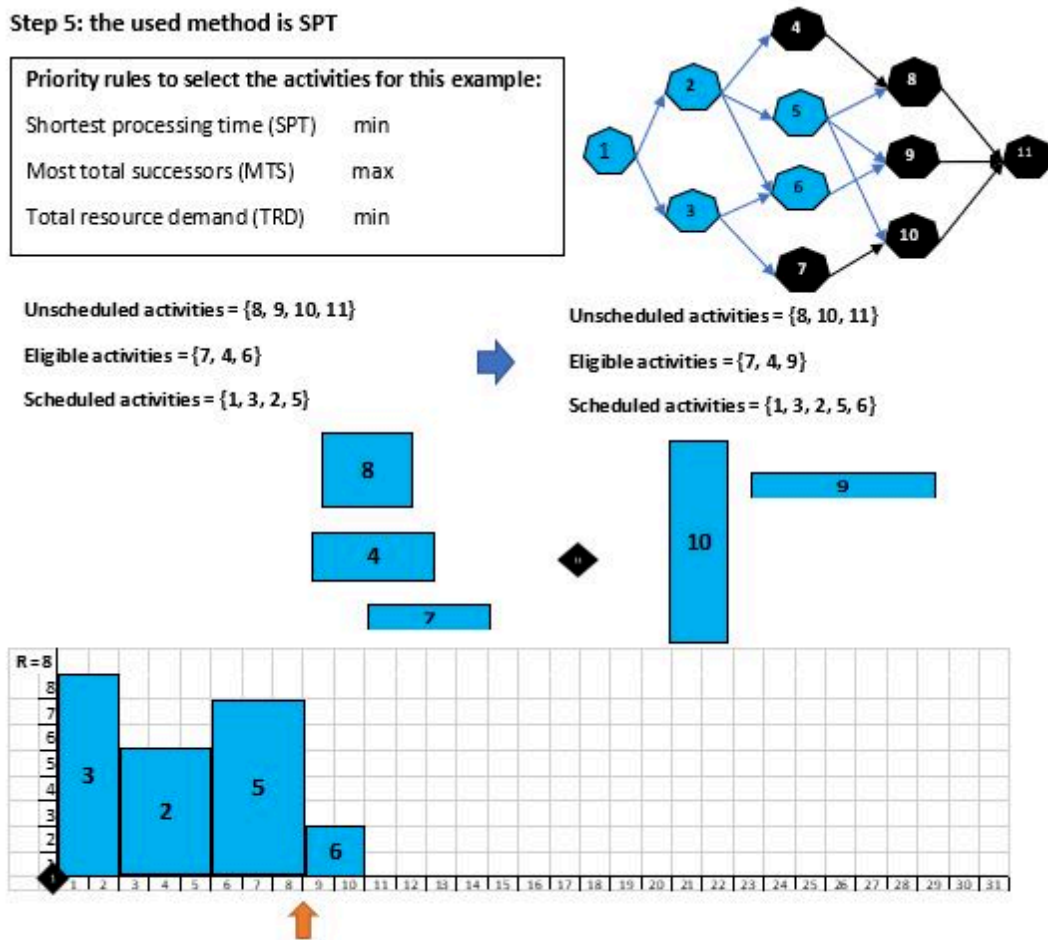


Figure 13 – Step 5 of the project scheduling: Task 6 is filtered out to schedule on the time chart according to shortest processing time (SPT).

Step 6: There are three activities 7, 4, and 9 in the list of eligible activities in this step that takes into account the first constraint, which is the precedence relationships between activities, as shown in Figure 14. We should select one of them to schedule it according to the resource constraint as in the previous steps. Also, there are 3 different priority rules as shown in Figure 14. The difficulty of the problem in this step is that we do not know which of the priority rules is the most appropriate to select an activity from the list of eligible activities. In this step, the total resource demand is selected to filter out the activity from the eligible activity list without any particular logic.

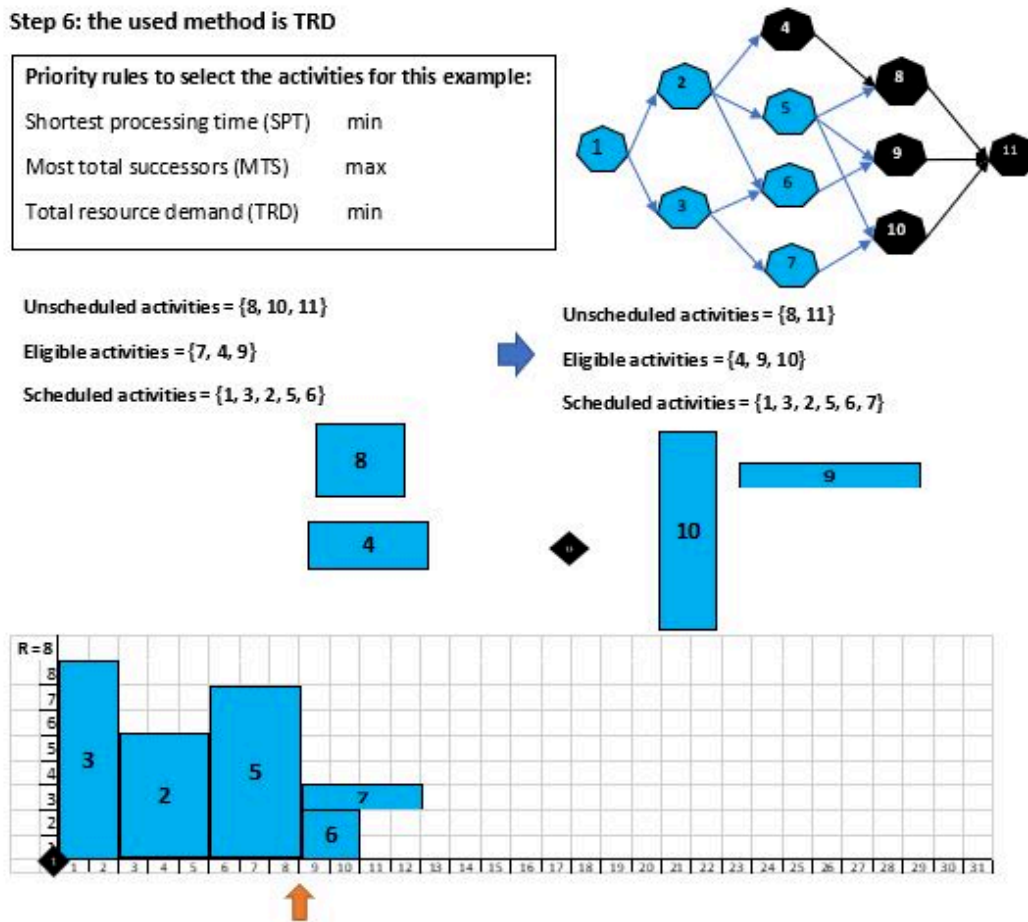


Figure 14 – Step 6 of project scheduling: activity 7 is filtered out to schedule on the time chart according to total resource demand (TRD).

Step 7: Activity 7 was scheduled in the previous step. So, according to the first condition, priority relations, activities 4, 9, and 10 are eligible for scheduling, as shown in Figure 15. According to the procedure, we need an appropriate priority rule to select an eligible activity. As in the previous steps, the difficulty of this step lies in choosing the appropriate priority rule. If the shortest processing time (SPT) is chosen, activity 10 will be filtered out to schedule. If we use most total successors (MTS), all three activities are eligible, and activity 9 is selected if we use total resource demand (TRD). In this step, the shortest processing time is selected without any special logic, so activity 10 is filtered out for planning in the time chart.

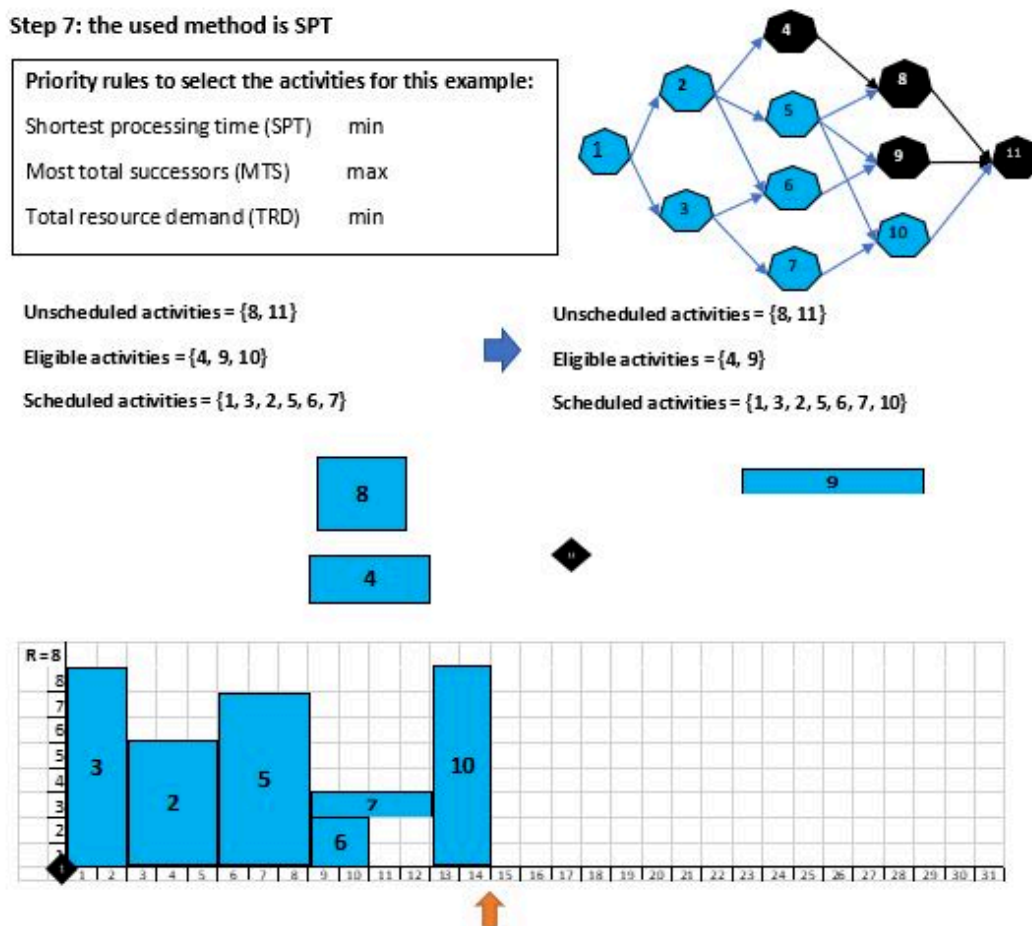


Figure 15 – Step 7 of the project scheduling: activity 10 is filtered out to schedule on the time chart according to the shortest processing time (SPT).

Step 8: In this scheduling step, there are two activities 4 and 9 in the list of eligible activities considering the first constraint, as shown in Figure 16. We should select one of the eligible activities to schedule it considering the resource constraint as in the previous steps. There are also 3 different priority rules as shown in Figure 16. The difficulty in this step is that we do not know which of the priority rules is the most appropriate to select an activity from the list of eligible activities. In this step, without any particular logic, the shortest processing time is selected to filter the activity from the list of eligible activities to be processed.

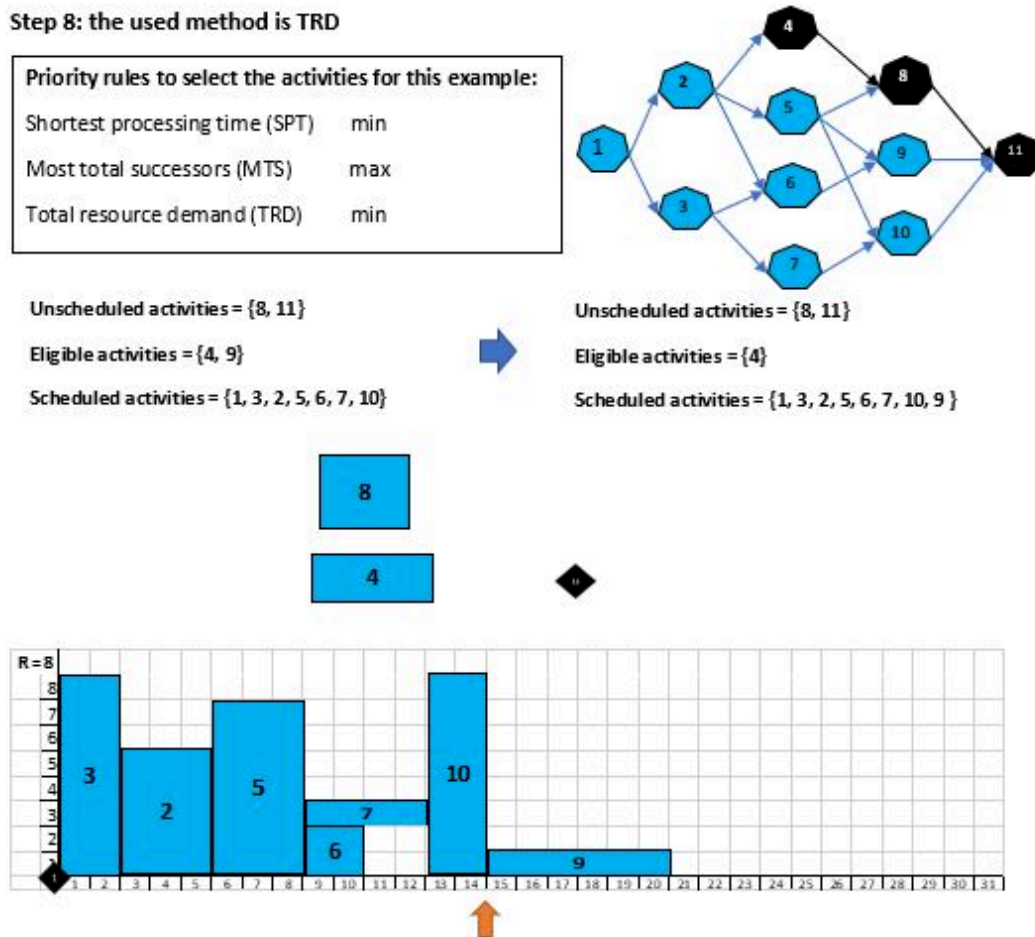


Figure 16 – Step 8 of project scheduling: activity 9 is filtered out to schedule on the time chart according to the shortest processing time (SPT).

Step 9: We should consider the precedence relations between the activities, which is the first constraint of the problem, in all steps of the project plan, as I explained in the previous steps. So, there is only activity 4 in the list of eligible activities, as shown in Figure 17. Then, activity 4 can be scheduled on the time chart according to the serial schedule generation scheme method. It is mandatory to update the lists at the end of each step.

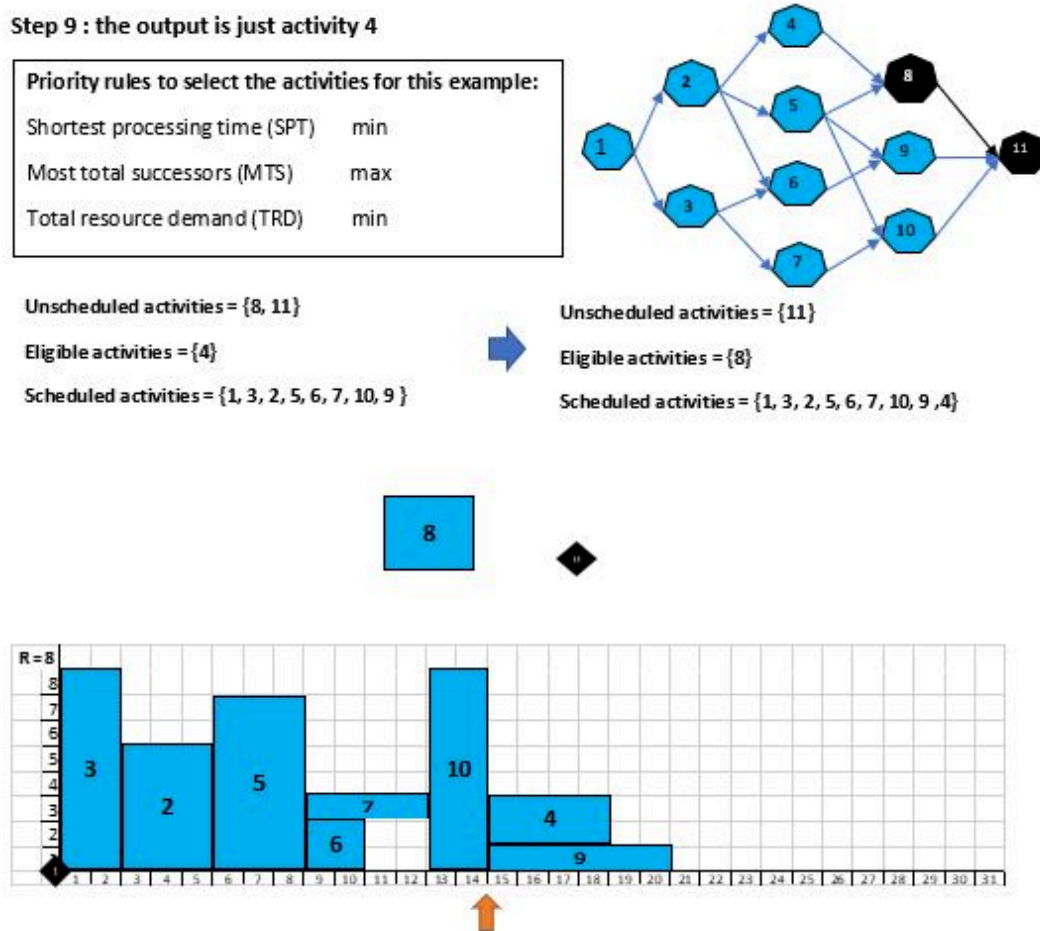
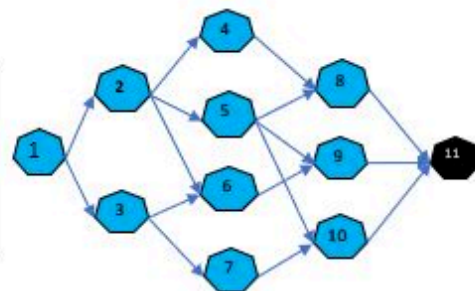


Figure 17 – Step 9 of project planning: activity 4 is selected for planning in the time diagram, taking into account the priority relationships between the activities.

Step 10: Updating the three lists is mandatory at the end of each step. After updating the list at the end of the previous step, it can be seen in Figure 18 that activity 8 is eligible for scheduling, taking into account the precedence relations. Then, activity 8 is selected for scheduling according to the serial schedule generation scheme method.

Step 10 : the output is just activity 8

Priority rules to select the activities for this example:	
Shortest processing time (SPT)	min
Most total successors (MTS)	max
Total resource demand (TRD)	min



Unscheduled activities = {11}
 Eligible activities = {8}
 Scheduled activities = {1, 3, 2, 5, 6, 7, 10, 9, 4}



Unscheduled activities = {}
 Eligible activities = {11}
 Scheduled activities = {1, 3, 2, 5, 6, 7, 10, 9, 4, 8}

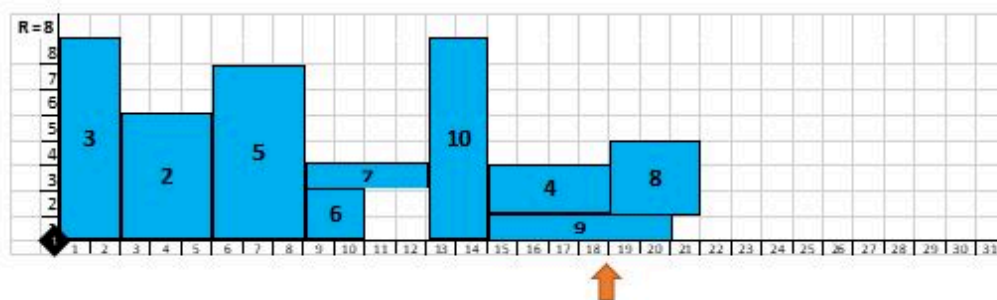


Figure 18 – Step 10 of project scheduling: activity 8 is selected for scheduling in the time diagram, taking into account the precedence relations between the activities.

Step 11: This step is the last step of scheduling the project with 11 activities. So it is obvious that a project with n activities requires n scheduling steps. After updating the lists at the end of step 10, it can be seen in Figure 19 that activity 11 can just be scheduled considering the precedence relationships. Then, activity 11, which is a milestone, is selected to be scheduled on the time chart according to the followed procedure. The project duration is 21 days, as shown in Figure 19.

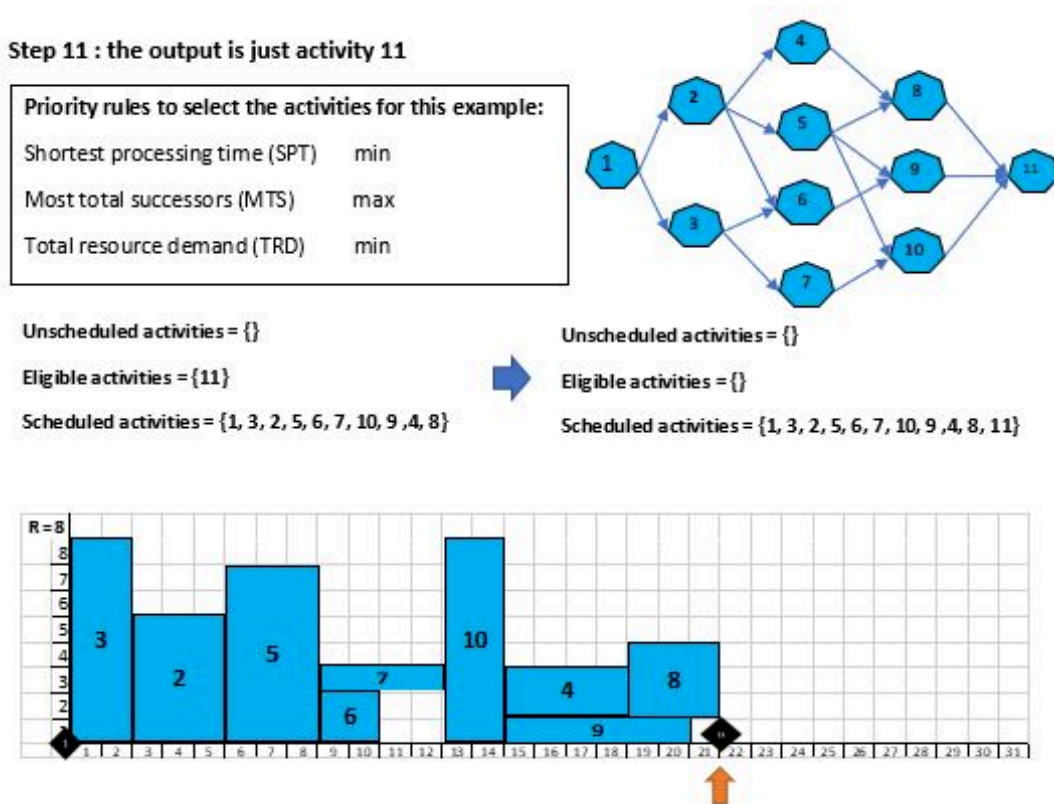


Figure 19 – Step 11 of project scheduling: activity 11 is selected to plan in the time chart considering the priority relationships between the activities. The project duration is 21 days according to the followed procedure.

The conclusion and the objective of the thesis

In the first and second sections, we explained the basic content and the resource-constrained project planning problem. Also, in the last section, I used a practical example to explain project scheduling and the difficulty of the problem.

It was explained that the resource-constrained project scheduling problem has two main constraints. The first constraint is the precedence relationships between project activities and the second constraint is the resource constraints. The second constraint means that it is not allowed to allocate more resources to the activities than are available.

The resource-constrained project scheduling problem (RCPSP), known as NP-complete problem²¹, is also classified in the category of nondeterministic polynomial problems. Therefore, exact solution techniques such as the branch-and-bound method are not suitable for solving large problems due to their complexity²². Therefore, researchers have developed meta-heuristics to solve this problem. The common meta-heuristics are explained in the next chapter.

Then, a practical example is used to explain the difficulty of the problem. As mentioned earlier, there are priority rules to assign a value to each activity under consideration. Then an activity is selected based on the value set and the selection criteria.

The difficulty of the problem occurs when there is more than one activity in the eligible set. Moreover, it is only possible to filter out one activity at each step of the serial schedule generation scheme (SSGS). Thus, if there are different priority rules and more than one activity in the eligible set, we do not know which of them is better for scheduling. As I explained in the previous section when scheduling the example project, the priority rules lead to different results depending on the project specifications and the set of constraints, such as the existing project conditions, etc. Therefore, it is necessary to choose an appropriate priority rule to select an activity from the set of eligible activities. This action may affect the duration of the project^{23 24}. It is also clear that the effectiveness

21. Jacek Blazewicz, Jan Karel Lenstra, and AHG Rinnooy Kan, « Scheduling subject to resource constraints: classification and complexity », *in: Discrete applied mathematics* 5.1 (1983), pp. 11–24.

22. JH Cho and Yeong-Dae Kim, « A simulated annealing algorithm for resource constrained project scheduling problems », *in: Journal of the Operational Research Society* 48.7 (1997), pp. 736–744.

23. Rainer Kolisch and Sanke Hartmann, « Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis », *in: Project scheduling*, Springer, 1999, pp. 147–178.

24. Ramon Alvarez-Valdes Olaguibel and Jose Manuel Tamarit Goerlich, « Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis », *in: Advances in project scheduling* (1989), pp. 113–134.

of the priority rules is different for different projects, since each project has a different character.²⁵. Therefore, the use of a single priority rule during project scheduling is not so effective because the project parameters differ from step to step.

The ultimate goal of the problem is to minimize the project duration subject to two constraints. To achieve this objective, I employ the serial schedule generation scheme (SSGS). According to the SSGS, activities are scheduled one by one or step by step. At each step of the project schedule, a selected priority rule filters out an eligible activity from the eligible set. In this thesis, I present two different neural network approaches to solve the standard resource-constrained project scheduling problem (RCPSP). They are the multilayer feed-forward neural network (MLFNN) approach and the convolutional neural network approach. They choose a priority rule as output at each step of project scheduling. For this purpose, the two evolved neural networks are fed with eight project parameters to select an appropriate priority rule as output. Therefore, at each scheduling step, the system can filter out a suitable eligible activity selected by a priority rule from the list of eligible activities and add it to the project schedule. The algorithm proceeds to schedule all the activities of the project considering the given project constraints.

the work is divided into six parts. The first part is the introduction, which explains the basic knowledge, the resource-constrained project scheduling problem and a practical example, and the last subsection explains the objective of the thesis. The literature review is the second part of this thesis. This chapter reviews published works that address the resource-constrained project scheduling problem. It also provides a classification of the published articles. Then, a summary of traditional metaheuristic techniques and related articles is given. In chapter two, the neural networks, inputs, and output set are presented. In chapter three, the proposed algorithm and the developed multilayer feed-forward neural network (MLFNN) are explained. In the second part of chapter three, I investigate the performance of the proposed approach using standard benchmark problems from the project schedule problems library (PSPLIB). In chapter four, the proposed algorithm and the developed convolutional neural network (CNN) are explained. I also investigate the performance of the proposed approach using the standard benchmark problems from the project schedule problems library (PSPLIB). Finally, the conclusion of the thesis is drawn.

25. Recep Kanit, Omer Ozkan, and Murat Gunduz, « Effects of project size and resource constraints on project duration through priority rule-based heuristics », *in: Artificial Intelligence Review* 32.1 (2009), pp. 115–123.

LITERATURE REVIEW

The resource-constrained project scheduling problem (RCPSP) and the proposed conventional meta-heuristic solution techniques have drawn the attention of many researchers to deal with this problem. Therefore, researchers have developed algorithms and methods to solve the problem. Researchers have developed various conventional meta-heuristic algorithms such as genetic algorithms, particle swarm optimization, ant colony optimization, bee colony optimization, simulated annealing, evolutionary algorithms, etc.¹.

Over the years, many researchers have used various solution techniques classified as metaheuristic techniques, exact methods, etc² to achieve the objective of the problem. Therefore, a considerable number of articles have been published to develop conventional metaheuristic approaches such as genetic algorithms (GA), ant colony optimization (ACO), simulated annealing (SA), etc., which are more practical than exact techniques to achieve an optimal or near-optimal objective, especially for project scheduling with a large number of activities^{3 4}.

This chapter reviews published articles and papers that address the problem of resource-constrained project planning. It also presents the classifications of the published articles, then explains a summary of traditional metaheuristic solution techniques and related articles.

1.1 Classification of published RCPSP articles

This section provides an overview of searched publications related to the RCPSP. These reference papers were collected from qualified databases such as scopus and include

-
1. Golab et al., *op. cit.*
 2. Nestor Raul Ortiz-Pimiento and Francisco Javier Diaz-Serna, « The project scheduling problem with non-deterministic activities duration: A literature review », *in: Journal of Industrial Engineering and Management (JIEM)* 11.1 (2018), pp. 116–134.
 3. Anurag Agarwal, Selcuk Colak, and Selcuk Erenguc, « Metaheuristic methods », *in: Handbook on Project Management and Scheduling Vol. 1*, Springer, 2015, pp. 57–74.
 4. Koulinas, Kotsikas, and Anagnostopoulos, *op. cit.*

journal articles and conference papers. The number of articles verifies that researchers are interested in developing metaheuristic techniques to achieve optimal or near-optimal results.

The frequency of algorithms developed to cope with the RCPSP presents interesting statistics. The statistics show that genetic algorithms are more popular than other metaheuristics. Hybrid algorithms are composed of two metaheuristics or they consist of one metaheuristic and another method⁵. However, other metaheuristic techniques such as particle swarm optimization (PSO), ant colony optimization (ACO), bee colony optimization (BCO), simulated annealing (SA), other evolutionary algorithms (EA), tabu search (TS), teaching-learning based optimization (TLBO), distribution estimation algorithm (DEA), etc. are also considered to address the objective of the problem.

1.2 developed conventional Meta-heuristics to solve the RCPSP

In this section, we present works that use conventional meta-heuristic techniques. We briefly explain the conventional meta-heuristics that have been developed to solve the resource-constrained project scheduling problem, and following each method we briefly describe the related published works.

1.2.1 RCPSP and genetic algorithms (GA)

Genetic algorithms originate from biology, where descendants want to inherit desirable traits. These algorithms also belong to the evolutionary algorithms. In a standard algorithm GA an initial population is generated, then a fitness function evaluates the individuals or children, then the algorithm applies the operators crossover and mutation to improve the solutions or individuals. The selected operator selects the parents to generate the solutions of the offspring for the next generations. The algorithm continues until it satisfies the specified stopping conditions, such as the specified number of generations^{6 7}. Below, we briefly review related work.

5. Önder Halis Bettemir and Rifat Sonmez, « Hybrid genetic algorithm with simulated annealing for resource-constrained project scheduling », *in: Journal of Management in Engineering* 31.5 (2015), p. 04014082.

6. Michel Gendreau, Jean-Yves Potvin, et al., *Handbook of metaheuristics*, vol. 2, Springer, 2010.

7. Jean-Yves Potvin and Michel Gendreau, *Handbook of Metaheuristics*, Springer, 2018.

A genetic algorithm approach proposed to solve the RCPSP with minimization of project duration as the objective. The proposed approach uses a permutation-based genetic encoding, then compares with the other two methods that benefit from a priority value-based and a priority rule-based representation, respectively⁸. A robust genetic algorithm has been proposed for the resource-constrained project scheduling problem. The algorithm generates the feasible activity list using a priority rule with a random key between 0 and 1. The method uses a crossover operator that randomly selects parents to pair for the next generation⁹. The proposed genetic algorithm uses six operators to generate offspring for the next generation and establish diversity in generations. Moreover, the objective of the problem is to minimize the project makespan¹⁰. A genetic algorithm with a fuzzy logic controller was proposed to deal with the RCPSP. The genetic operators were designed with the fuzzy logic controller. The child randomly takes some genes from one parent and then fills up the chromosome with the genes from the other parent by position-based crossover. The mutation operator randomly selects two positions of the chromosome to generate a child by exchange mutation¹¹. A genetic algorithm with two populations was developed for the RCPSP. This algorithm benefits from two separate populations, each containing left- and right-justified schedules. The serial schedule generation scheme is used to decode and generate the feasible schedules¹². A genetic algorithm has been proposed for the RCPSP with fuzzy activity duration and a fuzzy deadline to find a schedule that maximizes the schedule robustness. In the research, a genetic algorithm based on an activity list representation is proposed to solve the problem, and the performance of the proposed GA is compared with another GA based on the priority value representation¹³. A hybrid genetic algorithm was developed for the RCPSP that benefits from a new representation of the activity list. The method uses the two-point crossover

8. Sönke Hartmann, « A competitive genetic algorithm for resource-constrained project scheduling », in: *Naval Research Logistics (NRL)* 45.7 (1998), pp. 733–750.

9. Javier Alcaraz and Concepción Maroto, « A robust genetic algorithm for resource allocation in project scheduling », in: *Annals of operations Research* 102.1 (2001), pp. 83–109.

10. Khalil S. Hindi, Hongbo Yang, and Krzysztof Fleszar, « An evolutionary algorithm for resource-constrained project scheduling », in: *IEEE Transactions on evolutionary computation* 6.5 (2002), pp. 512–518.

11. Kwan Woo Kim, Mitsuo Gen, and Genji Yamazaki, « Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling », in: *Applied soft computing* 2.3 (2003), pp. 174–188.

12. Dieter Debels and Mario Vanhoucke, « A bi-population based genetic algorithm for the resource-constrained project scheduling problem », in: *International Conference on Computational Science and Its Applications*, Springer, 2005, pp. 378–387.

13. Hong Wang, Dan Lin, and Minqiang Li, « A genetic algorithm for solving fuzzy resource-constrained project scheduling », in: *International Conference on Natural Computation*, Springer, 2005, pp. 171–180.

and mutation operator that randomly selects genes for mutation¹⁴. The genetic algorithm applied is based on a priority list of activities to optimize the project makespan. The algorithm benefits from local and global search methods. In addition, bit string crossover and bit flipping mutation with a certain probability are used to achieve the goal¹⁵. A genetic algorithm capable of providing near-optimal heuristic solutions has been proposed. The presented method was extended with a decomposition-based genetic algorithm that iteratively solves subprojects of the project¹⁶. A permutation-based elitist GA was presented, the main aspect of which is called the elitist roulette selection operator. The method uses a one-point crossover and uniform mutation to generate children. Moreover, the feasible solutions are generated by a serial schedule generation method¹⁷. A genetic algorithm capable of outperforming a branch-and-bound method has been proposed for the problem. The objective of the study is subject to time constraints while minimizing the cost of resource availability. The investigated problem under study involves both time-independent fixed costs and time-dependent variable resource rental costs¹⁸. A GA approach was developed using an elitist strategy to select the best individuals for the next generation. A one-point crossover and uniform mutation operator were used to generate the offspring, and a schedule generation scheme is employed to generate feasible schedules¹⁹. A hybrid genetic algorithm was applied to solve the RCPSP. The proposed algorithm includes a peak crossover operator, a local improvement operator, and a parent selection method. The algorithm consists of two phases: The first phase is called general search and the second phase searches in the neighborhood of the best generated solutions. The serial schedule generation scheme is used to generate active schedules that are checked for re-

14. Javier Alcaraz and Concepción Maroto, « A hybrid genetic algorithm based on intelligent encoding for project scheduling », *in: Perspectives in modern project scheduling*, Springer, 2006, pp. 249–274.

15. Leonidas Sakalauskas and Gražvydas Felinskas, « Optimization of resource constrained project schedules by genetic algorithm based on the job priority list », *in: Information technology and control* 35.4 (2006).

16. Dieter Debels and Mario Vanhoucke, « A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem », *in: Operations Research* 55.3 (2007), pp. 457–469.

17. Jin-Lee Kim, « Permutation-based elitist genetic algorithm using serial scheme for large-sized resource-constrained project scheduling », *in: 2007 Winter Simulation Conference*, IEEE, 2007, pp. 2112–2118.

18. Francisco Ballestin, « A genetic algorithm for the resource renting problem with minimum and maximum time lags », *in: European Conference on Evolutionary Computation in Combinatorial Optimization*, Springer, 2007, pp. 25–35.

19. Jin-Lee Kim and Ralph D Ellis Jr, « Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling », *in: Journal of construction engineering and management* 134.11 (2008), pp. 904–913.

source availability and earliest precedence²⁰. A genetic algorithm was used to optimize the project makespan while resources are constrained. In addition, the method uses uniform crossover and swap mutation to generate the offspring of the next generations²¹. An improved elite genetic algorithm was proposed to optimize the project duration. The proposed method generates a random number for the initial solutions. It also benefits from one-point crossover and uniform mutation to generate new solutions for the next generations²². A genetic algorithm has been proposed in which chromosomes are represented based on random keys. The random keys help in obtaining feasible individuals generated by crossover. In the algorithm, each chromosome consists of two groups of genes. The first group represents the priorities and the second represents the delay time. To generate the next generation, the algorithm performs three actions. The best individuals are directly transferred to the next generation, while the other individuals are generated by one-point crossover and mutation. The algorithm uses the scheduling generation scheme to generate the schedules²³. To minimize project duration, a genetic algorithm based on an object-oriented model was developed. The method employs a one-point and a two-point crossover. There is also a mutation operator that mutates the genes of chromosomes according to a certain probability²⁴. A genetic algorithm with neighborhood search was proposed to handle the RCPSP, including the non-preemptive activities, and minimize the project duration. In the research, it was claimed that the neighborhood search operator can improve the feasible solution if the start times of some activities are fixed to search for other activities²⁵. A genetic algorithm was developed in which the procedure is equipped with a random key, a parameterized uniform crossover, an SSGS method, and a backward-forward improvement. The objective is defined as minimizing the project

20. Vicente Valls, Francisco Ballestin, and Sacramento Quintanilla, « A hybrid genetic algorithm for the resource-constrained project scheduling problem », *in: European journal of operational research* 185.2 (2008), pp. 495–508.

21. Toni Frankola, Marin Golub, and Domagoj Jakobovic, « Evolutionary algorithms for the resource-constrained scheduling problem », *in: ITI 2008-30th International Conference on Information Technology Interfaces*, IEEE, 2008, pp. 715–722.

22. Jin-Lee Kim, « Improved genetic algorithm for resource-constrained scheduling of large projects », *in: Canadian journal of civil engineering* 36.6 (2009), pp. 1016–1027.

23. Jorge JM Mendes, José Fernando Gonçalves, and Mauricio GC Resende, « A random key based genetic algorithm for the resource constrained project scheduling problem », *in: Computers & operations research* 36.1 (2009), pp. 92–109.

24. Jairo R Montoya-Torres, Edgar Gutierrez-Franco, and Carolina Pirachicán-Mayorga, « Project scheduling with limited resources using a genetic algorithm », *in: International Journal of Project Management* 28.6 (2010), pp. 619–628.

25. Sepehr Proon and Mingzhou Jin, « A genetic algorithm with neighborhood search for the resource-constrained project scheduling problem », *in: Naval Research Logistics (NRL)* 58.2 (2011), pp. 73–82.

duration²⁶. A GA was introduced that uses a standardized random key, a local search method, and an elite selection method to deal with the RCPSP. The crossover operator used selects one of the parents from above and another randomly. Also, the method uses two different mutation operators²⁷. A genetic algorithm was developed in which the initial feasible solutions are randomly generated. The proposed algorithm benefits from three different types of crossover generations, namely one-point crossover, two-point crossover, and priority-reserve crossover. In addition, two mutation operators are used to generate the offspring of the next generations²⁸. A genetic-based hyperheuristic algorithm was introduced to control some low-level heuristics that work in the solution domain. Moreover, the chromosomes specify the order in which the algorithm applies the low-level heuristics²⁹. A genetic algorithm is used to deal with RCPSP. In the algorithm, a scheme for encoding priority values is predicted. The method uses two crossover operators, a one-point operator and a uniform operator, and a classical mutation operator³⁰. GA was used, where the fitness function reports the value of the project duration back to the algorithm by evaluating the solutions. There are two operators, the crossover and mutation operators, which act on the current population, excluding the identified fitted individuals³¹. Five different genetic algorithms have been proposed for the RCPSP and a comparison between them is being investigated. The objective of the research is to schedule the project activities to minimize the project duration. The initial population is generated using a priority rule based algorithm³². The proposed genetic algorithm benefits from a random initial generation, a one-point crossover operator, and random mutation. In addition, two selection methods are used in the algorithm. In the study, the completion phase of construction

26. José Fernando Gonçalves, Mauricio GC Resende, and Jorge JM Mendes, « A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem », in: *Journal of Heuristics* 17.5 (2011), pp. 467–486.

27. Hong Wang, Tongling Li, and Dan Lin, « Efficient genetic algorithm for resource-constrained project scheduling problem », in: *Transactions of Tianjin University* 16.5 (2010), pp. 376–382.

28. Marcin Klimek, « A genetic algorithm for the project scheduling with the resource constraints », in: *Annales Universitatis Mariae Curie-Skłodowska. Sectio AI, Informatica* 10.1 (2010).

29. Konstantinos P Anagnostopoulos and Georgios K Koulinas, « A genetic hyperheuristic algorithm for the resource constrained project scheduling problem », in: *IEEE Congress on Evolutionary Computation*, IEEE, 2010, pp. 1–6.

30. Yan Hua Ren, De Cai Kong, and Wu Liang Peng, « A genetic algorithm based solution with schedule mode for RCPSP », in: *Advanced Materials Research*, vol. 268, Trans Tech Publ, 2011, pp. 1802–1805.

31. Jie Zhu, Xiaoping Li, and Weiming Shen, « Effective genetic algorithm for resource-constrained project scheduling with limited preemptions », in: *International Journal of Machine Learning and Cybernetics* 2.2 (2011), pp. 55–65.

32. F Gargiulo and D Quagliarella, « Genetic algorithms for the resource constrained project scheduling problem », in: *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*, IEEE, 2012, pp. 39–47.

projects was considered³³. A genetic algorithm was introduced to solve the RCPSP, and the innovation of the algorithm is the use of a magnet-based crossover operator. This operator can obtain up to two contiguous parts of the receiver and one contiguous part of the donator genotype³⁴. A genetic algorithm was developed for the RCPSP in which the fitness function has a feedback loop with the makepan value. The binary string chromosomes also represent the individuals, and there is a one-point crossover that produces the offspring for the next generations³⁵. The genetic algorithm with two subpopulations was proposed for a bi-objective problem, where the defined objectives are the minimization of the project and the NPV of the project³⁶. A local search approach based on a genetic algorithm was proposed to minimize the project duration. For this purpose, the neighborhood operator acts on a selected individual in the current population³⁷. The article presents a GA aimed at minimizing project duration. To achieve the objective, the method uses a one-point crossover operator and a mutation operator. The mutation operator exchanges the two positions of two genes. Moreover, the SSGS method is used to decode the individuals³⁸. A genetic algorithm was introduced to deal with the RCPSP. The algorithm randomly generates the initial individuals. The procedure benefits from a classical one-point crossover and a classical mutation operator, which exchange the position of the gene to achieve the objective of the problem, which was defined as minimizing the project duration³⁹. A genetic algorithm benefiting from a priority-based crossover was developed to generate new solutions. Also, a local search operator was used to improve the solutions. The objective of the research is to minimize the project duration⁴⁰. A GA

33. Ning Dong et al., « A genetic algorithm-based method for look-ahead scheduling in the finishing phase of construction projects », in: *Advanced Engineering Informatics* 26.4 (2012), pp. 737–748.

34. Reza Zamani, « A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem », in: *European journal of operational research* 229.2 (2013), pp. 552–559.

35. S Diana, L Ganapathy, and Ashok K Pundir, « An improved genetic algorithm for resource constrained project scheduling problem », in: *International Journal of Computer Applications* 78.9 (2013).

36. Somayeh Khalili, Amir Abbas Najafi, and Seyed Taghi Akhavan Niaki, « Bi-objective resource constrained project scheduling problem with makespan and net present value criteria: two meta-heuristic algorithms », in: *The International Journal of Advanced Manufacturing Technology* 69.1 (2013), pp. 617–626.

37. Olfa Dridi, Saoussen Krichen, and Adel Guitouni, « Solving resource-constrained project scheduling problem by a genetic local search approach », in: *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, IEEE, 2013, pp. 1–5.

38. Sachin U Kadam and Narendra S Kadam, « Solving resource-constrained project scheduling problem by genetic algorithm », in: *2014 2nd International Conference on Business and Information Management (ICBIM)*, IEEE, 2014, pp. 159–164.

39. Ismail M Ali et al., « Memetic algorithm for solving resource constrained project scheduling problems », in: *2015 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2015, pp. 2761–2767.

40. Sachin Uttam Kadam and Sandip U Mane, « A genetic-local search algorithm approach for resource

was proposed where the method creates a random list of activities to generate individuals and a local search operator tries to improve feasible schedules. In addition, the SGS is used to decode chromosomes to minimize the project makespan⁴¹. A genetic algorithm is developed in which the list of feasible activities includes a chromosome with two binary codes at the end of each individual. The first code represents the type of SGS method (serial or parallel), and the second code indicates the scheduling direction (forward or backward). In addition, the initial population is created by randomly selecting three priority rules: minimum latest finish time, minimum latest start time, and minimum total slack. The method benefits from a two-point crossover operator called modified magnet-based crossover, and a mutation operator is used to achieve diversity across all individuals in the population⁴². A genetic algorithm was developed to obtain an optimized schedule. To achieve the objective, the elitist strategy is used to find the fitted individuals. Then a crossover operator is used to generate an individual from two parents. The non-fitted individuals of the current population are replaced by the children of the next generation. Also, a mutation operator selects some individuals to increase the diversity using a local search approach⁴³.

1.2.2 RCPSP and particle swarm optimization (PSO)

Particle swarm optimization was developed following the social collective behavior of living organisms, e.g., the collective movement of birds or fishes. Normally, a PSO consists of a swarm of particles moving in an n-dimensional space. Each particle is identified at each instant by its position, velocity vectors, and its own best position. The position and velocity of the particles are initialized randomly and optimized from time to time during the algorithm. Moreover, the quality of the solutions is evaluated and compared using a fitness function. In particle swarm optimization, the best local particle is introduced as the global best solution. The algorithm continues until the termination conditions such

constrained project scheduling problem », *in: 2015 International Conference on Computing Communication Control and Automation*, IEEE, 2015, pp. 841–846.

41. Evgenii N Goncharov and Valentin V Leonov, « Genetic algorithm for the resource-constrained project scheduling problem », *in: Automation and Remote Control* 78.6 (2017), pp. 1101–1114.

42. Roubila Lilia Kadri and Fayez F Boctor, « An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case », *in: European Journal of Operational Research* 265.2 (2018), pp. 454–462.

43. Jia Liu et al., « Solving resource-constrained project scheduling problem via genetic algorithm », *in: Journal of Computing in Civil Engineering* 34.2 (2020), p. 04019055.

as the maximum number of iterations or the computation time are satisfied^{44 45}. Related work is mentioned below.

A PSO was developed for RCPSP to minimize project duration, considering priority-based representation and permutation-based representation, respectively⁴⁶. A hybrid particle method and mapped crossover were proposed to optimize the project duration. In addition, the sequence represented by particles is converted into a schedule by the SSGS method into feasible schedules⁴⁷. A PSO was developed to minimize the project duration. To this end, the algorithm uses a new permutation of a priority-based encoding scheme. In addition, the procedure uses an adopted mechanism to update the velocity and position⁴⁸. A particle swarm optimization approach was introduced to solve the resource-constrained scheduling problem with multiple processors. In this algorithm, two new rules are proposed, namely, the anti-inertia solution generation rule and the bidirectional search rule⁴⁹. A particle swarm optimization was presented to minimize the project duration. In this method, the precedence constraints between activities are processed by the repair strategy, and the resource constraints of the project are indirectly processed by calculating the value of the particle⁵⁰. An improved particle swarm optimization was proposed to solve the RCPSP. In the algorithm, a mapping between the feasible schedule and the position of the particles is established. Then, the method starts exploring the global best and the local best until the stopping conditions are satisfied⁵¹. A PSO employed a delay local search rule and a bidirectional scheduling rule. They prevent staying in the local

44. Russell Eberhart and James Kennedy, « Particle swarm optimization », *in: Proceedings of the IEEE international conference on neural networks*, vol. 4, Citeseer, 1995, pp. 1942–1948.

45. Sylvain Kemmoe Tchomte and Michel Gourgand, « Particle swarm optimization: A study of particle displacement for solving continuous and combinatorial optimization problems », *in: International Journal of Production Economics* 121.1 (2009), pp. 57–67.

46. Hong Zhang et al., « Particle swarm optimization-based schemes for resource-constrained project scheduling », *in: Automation in construction* 14.3 (2005), pp. 393–404.

47. Hong Zhang, Heng Li, and CM Tam, « Permutation-based particle swarm optimization for resource-constrained project scheduling », *in: Journal of computing in civil engineering* 20.2 (2006), pp. 141–149.

48. Wuliang Peng and Yonghe Wei, « PSO for solving RCPSP », *in: 2008 Chinese control and decision conference*, IEEE, 2008, pp. 818–822.

49. Shih-Tang Lo et al., « Using particle swarm optimization to solve resource-constrained scheduling problems », *in: 2008 IEEE Conference on Soft Computing in Industrial Applications*, IEEE, 2008, pp. 38–43.

50. Kai Zhang, Guorong Zhao, and Jing Jiang, « Particle swarm optimization method for resource-constrained project scheduling problem », *in: 2009 9th International Conference on Electronic Measurement & Instruments*, IEEE, 2009, pp. 4–792.

51. Qiang Wang and Jianxun Qi, « Improved particle swarm optimization for RCP scheduling problem », *in: The Sixth International Symposium on Neural Networks (ISNN 2009)*, Springer, 2009, pp. 49–57.

search and evolving the local search to reach a global solution minimum⁵². A developed particle swarm optimization (PSO) method based on two different particle representations was proposed to solve the RCPSP. This algorithm uses the methods of SSGS and forward-backward improvement. The goal is also to minimize the project duration⁵³. A pseudo-particle swarm optimization (PSO) algorithm was proposed to solve the RCPSP. The presented algorithm uses the path relinking procedure as a way for the particles to move to local and global best positions⁵⁴. An improved particle swarm optimization (PSO) algorithm was proposed that included two operators, namely, greedy random local search and double justification. In addition, the algorithm used the SSGS method to decode the schedules⁵⁵. The radius particle swarm optimization was proposed to solve the RCPSP. The presented PSO was developed by regrouping the agent particles within the corresponding radius of the circle. This method initializes the group of particles which calculates the fitness function, and finds the best particle in the group⁵⁶. A particle swarm optimization based hyperheuristic was developed for the classical RCPSP to obtain the feasible minimized project duration. In the proposed study, each particle consists of eight integer numbers and each swarm consists of twenty particles. Initially, the global best solution is zero, then, it is equal to the local best solution. The procedure controls eight low-level heuristics that are randomly applied to the particles. The method uses the serial scheduling generation scheme to decode the particles and forward-backward improvement to improve the solutions⁵⁷. An improved method for determining particle position and velocities was proposed. The standard PSO was modified at two points: in updating the position and velocity of the particles. The objective of the method is to minimize the project duration⁵⁸. A hybrid particle swarm optimization was developed for the RCPSP,

52. Ruey-Maw Chen et al., « Using novel particle swarm optimization scheme to solve resource-constrained scheduling problem in PSPLIB », in: *Expert systems with applications* 37.3 (2010), pp. 1899–1910.

53. Fei Li, Changtao Lai, and Yongyi Shou, « Particle swarm optimization for preemptive project scheduling with resource constraints », in: *2011 IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE, 2011, pp. 869–873.

54. Mohammad Mahdi Nasiri, « A pseudo particle swarm optimization for the RCPSP », in: *The International Journal of Advanced Manufacturing Technology* 65.5 (2013), pp. 909–918.

55. Qiong Jia and Yoonho Seo, « An improved particle swarm optimization for the resource-constrained project scheduling problem », in: *The International Journal of Advanced Manufacturing Technology* 67.9 (2013), pp. 2627–2638.

56. Mana Anantathanvit and Mud-Armeen Munlin, « Radius particle swarm optimization for resource constrained project scheduling problem », in: *16th Int'l Conf. Computer and Information Technology*, IEEE, 2014, pp. 24–29.

57. Koulinas, Kotsikas, and Anagnostopoulos, *op. cit.*

58. Neetesh Kumar and Deo Prakash Vidyarthi, « A model for resource-constrained project scheduling

which benefits from three types of particle solution representation and two vector decoding methods⁵⁹. A PSO was introduced for the RCPSP, while the available quantities of resources are variable and the duration of the activities are fixed⁶⁰. The proposed PSO benefits from adaptive mutation and a forward-backward method to obtain a minimum project duration⁶¹. A particle swarm optimization based method was developed to deal with the RCPSP. In this studied problem, a resource pool was defined between different sites. Also, two different types of resources, namely fixed and mobile resources, were adopted⁶².

1.2.3 RCPSP and ant colony optimization (ACO)

Ant colony optimization is a metaheuristic for solving difficult combinatorial optimization problems. This optimization method is modeled after the collective social behavior of ant colonies and can be categorized as swarm intelligence. A colony of ants uses pheromone trails as indirect links between ants, allowing them to find suitable and short routes between their nest and food sources. The artificial ants are employed to maintain a non-systematized structure and make probabilistic decisions depending on the pheromone trails. There are different types of ant colony optimization, but in a standard type, each solution is made by probabilistic decisions. The solution found leaves a certain amount of pheromones on the path of the search space. The next generation solutions follow the nearby found suitable solution in the solution space or the marked direction. Thus, the feasible solutions in the neighborhood can be generated and then evaluated to obtain the shortest solution path or solutions with better quality. The procedure continues until the

using adaptive PSO », *in: Soft computing* 20.4 (2016), pp. 1565–1580.

59. Yongyi Shou, Ying Li, and Changtao Lai, « Hybrid particle swarm optimization for preemptive resource-constrained project scheduling », *in: Neurocomputing* 148 (2015), pp. 122–128.

60. Jiby Joy, Srijith Rajeev, and Vishnu Narayanan, « Particle swarm optimization for resource constrained-project scheduling problem with varying resource levels », *in: Procedia Technology* 25 (2016), pp. 948–954.

61. Mudarmeen Munlin, « Solving resource-constrained project scheduling problem using metaheuristic algorithm », *in: 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, IEEE, 2018, pp. 344–349.

62. Cyrine Stiti and Olfa Belkahla Driss, « A new approach for the multi-site resource-constrained project scheduling problem », *in: Procedia Computer Science* 164 (2019), pp. 478–484.

termination conditions are satisfied^{63 64 65}. The associated work is listed below.

An ant colony optimization (ACO) algorithm was presented for the resource constrained project scheduling problem (RCPSP). The algorithm uses a combination of two pheromone evaluation methods by the ants to find new solutions⁶⁶. Another ant colony optimization was presented to minimize the project duration. It also employs two operators, namely shift and backshift operators, to obtain the solutions of the neighbors⁶⁷. A modified ant colony system has been proposed to solve the resource-constrained scheduling problems. A two-dimensional matrix for scheduling activities with time is proposed, which allows a parallel scheme for solving project scheduling problems. A rule for generating a delay solution to escape the local optimal solution is proposed⁶⁸. An ant algorithm with dual ant colonies was developed to improve the effective allocation of project resources. One ant colony uses the forward planning technique, while another ant colony uses the backward scheduling technique. The pheromone information of the two ant colonies is exchanged from period to period to avoid early local convergence⁶⁹. An improved ACO was developed to deal with the RCPSP. The introduced method uses a local search method called PC -2opt, which guarantees precedence constraints between activities⁷⁰. A hybrid ant colony optimization approach was proposed to improve schedule quality and minimize project duration. The method is associated with an extended dual justification, in which the activity splitting is applied to predict whether the schedule can be improved⁷¹.

63. Marco Dorigo and Gianni Di Caro, « Ant colony optimization: a new meta-heuristic », *in: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 2, IEEE, 1999, pp. 1470–1477.

64. Daniel Merkle, Martin Middendorf, and Hartmut Schmeck, « Ant colony optimization for resource-constrained project scheduling », *in: IEEE transactions on evolutionary computation* 6.4 (2002), pp. 333–346.

65. Potvin and Gendreau, *op. cit.*

66. Merkle, Middendorf, and Schmeck, *op. cit.*

67. Shipeng Luo, Cheng Wang, and Jinwen Wang, « Ant colony optimization for resource-constrained project scheduling with generalized precedence relations », *in: Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, 2003, pp. 284–289.

68. Ruey-Maw Chen and Shih-Tang Lo, « Using an enhanced ant colony system to solve resource-constrained project scheduling problem », *in: Int. J. Comput. Sci. Netw. Secur* 6 (2006), pp. 75–84.

69. Yongyi Shou, « A Bi-directional Ant colony algorithm for resource constrained project scheduling », *in: 2007 IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE, 2007, pp. 1027–1031.

70. Yumiao Zhou, Qingshun Guo, and Rongwei Gan, « Improved ACO algorithm for resource-constrained project scheduling problem », *in: 2009 international conference on artificial intelligence and computational intelligence*, vol. 3, IEEE, 2009, pp. 358–365.

71. Linyi Deng, Van Lin, and Ming Chen, « Hybrid ant colony optimization for the resource-constrained project scheduling problem », *in: Journal of Systems Engineering and Electronics* 21.1 (2010), pp. 67–71.

1.2.4 RCPSP and bees colony optimization (BCO)

Bee colony optimization is another metaheuristic for solving optimization problems. Like ACO, BCO is inspired by natural collective behavior and belongs to the category of swarm-based optimization algorithms. In it, honeybees try to find flower patches. The search for flower patches starts with some bees randomly searching for food sources and exploring the space. Then the bees return to the hive to announce the location of the flower patches by doing waggle dance, which initiates communication between the bees. Waggle dance helps the bee colony by communicating three pieces of information about the food source: Direction, distance and quality of the flower patches. There are different types of NCO, but a standard type is based on a random solution and a neighborhood solution. In this algorithm, an initial population is generated, which is then evaluated using a fitness function. Additional bees are used to generate the neighborhood solutions for selected parts of the search space. The procedure continues until the given stopping conditions are satisfied⁷².

The standard algorithm ABC was developed by incorporating a cooperative approach with another algorithm called Split ABC. S-ABC aims to improve the performance of the standard algorithm ABC by taking advantage of cooperation as social behavior. In addition, the method uses SSGS to generate feasible schedules⁷³. An artificial bee colony algorithm with a random key was proposed for real-time resource-constrained project scheduling to minimize the project duration. The problem representation is based on a random key and a heuristic priority rule to assign activities⁷⁴. A bee algorithm was introduced with a new formula to evaluate the quality of the solutions found in the search space⁷⁵. Against the stochastic RCPSP, an artificial bee colony was developed in which the activity duration is variable with a certain probability to minimize the project duration⁷⁶. Three types of bee algorithms have been studied in the research. In addition,

72. Duc Truong Pham et al., « The bees algorithm—a novel tool for complex optimisation problems », in: *Intelligent production machines and systems*, Elsevier, 2006, pp. 454–459.

73. Reza Akbari, Vahid Zeighami, and Koorush Ziarati, « A Cooperative Artificial Bee Colony Optimizer », in: *7th International Industrial Engineering Conference*, 2010, pp. 1–10.

74. Yan-jun Shi et al., « An artificial bee colony with random key for resource-constrained project scheduling », in: *Life system modeling and intelligent computing*, Springer, 2010, pp. 148–157.

75. Amir Sadeghi et al., « Using bees algorithm to solve the resource constrained project scheduling problem in PSPLIB », in: *International Conference on Theoretical and Mathematical Foundations of Computer Science*, Springer, 2011, pp. 486–494.

76. Amin Tahooneh and Koorush Ziarati, « Using artificial bee colony to solve stochastic resource constrained project scheduling problem », in: *International Conference in Swarm Intelligence*, Springer, 2011, pp. 293–302.

a method is used to convert infeasible schedules into feasible schedules. The method benefits from local search, where priority values are exchanged to create a neighboring solution⁷⁷. The permutation-based artificial bee colony algorithm was developed to solve the resource-constrained project scheduling problem (RCPSP). The method uses a one-point crossover operator to generate neighbor solutions⁷⁸. To deal with the problem, an artificial bee colony was proposed in which three swap-based operators are randomly selected to generate neighbor solutions⁷⁹.

1.2.5 RCPSP and simulated annealing (SA)

Simulated annealing is referred to as SA and is a local search algorithm (metaheuristic) capable of escaping from local optima. SA is a random search technique and one of the reasons for the popularity of this method is its simplicity. This method was inspired by the process of physical annealing of solids. When a crystalline solid is heated and then slowly formed into a solid, a qualified solid is formed with minimal energy input. The simulated annealing algorithm combines this part of thermodynamics with local search to obtain an optimal solution. In this algorithm, an initial solution is generated at the beginning, and there is always a current solution. Moreover, the neighbors of the current solution can also be the current solution if the neighboring solution is better than the current solution. However, an impractical solution can also be a current solution in certain cases to prevent a local optimum. In addition, the SA algorithm incorporates a temperature parameter that has a large value initially and then slowly decreases to obtain a better solution. The solutions are also evaluated with a fitness function^{80 81}. Related work is listed below.

A simulated annealing algorithm was developed for the RCPSP with the objective of minimizing project duration. In the proposed algorithm, a solution is introduced with a priority list, a vector of numbers, each of which denotes the priority of each activity. A pri-

77. Reza Akbari, Vahid Zeighami, and Koorush Ziarati, « Artificial bee colony for resource constrained project scheduling problem », *in: International Journal of Industrial Engineering Computations* 2.1 (2011), pp. 45–60.

78. Qiong Jia and Yoonho Seo, « Solving resource-constrained project scheduling problems: conceptual validation of FLP formulation and efficient permutation-based ABC computation », *in: Computers & Operations Research* 40.8 (2013), pp. 2037–2050.

79. Broderick Crawford et al., « An artificial bee colony algorithm for the resource constrained project scheduling problem », *in: International Conference on Human-Computer Interaction*, Springer, 2015, pp. 582–586.

80. Agarwal, Colak, and Erenguc, *op. cit.*

81. Potvin and Gendreau, *op. cit.*

ority scheduling method is employed to generate a schedule from a given priority list⁸². An adaptation of the simulated annealing algorithm was proposed for solving non-preemptive RCPSP where resources are limited and renewable from period to period. The algorithm was proposed to deal with single-mode and multi-mode problems to optimize different objectives⁸³. Simulated annealing (SA) was proposed to handle the RCPSP to minimize the project duration. The initial solution in the method is generated by the heuristic priority rule called shortest processing time method (SPT). Moreover, the proposed algorithm uses a local search method that generates a set of neighbors. The neighbors are generated by randomly exchanging the activities at the new positions considering the priority constraints. Also, a schedule generation scheme is used to decode the schedules⁸⁴. The proposed algorithm is based on encoding the schedule by the priority list of activities. The optimal priority list of activities is found in algorithms of local and global search, which are random search and simulated annealing methods with the variable neighborhood⁸⁵. Three SA variants are studied and applied to RCPSP. The applied algorithms use a tabu list to search for a neighbor solution⁸⁶. An improved simulated annealing was proposed to optimize the duration of the project execution, while another objective of the problem is to study the resource consumption among the same obtained solutions⁸⁷.

1.2.6 RCPSP and tabu search (TS)

Tabu search is a heuristic method based on a local search technique formed on the basis of displacement strategies and neighborhood solution search. This method provides solutions that are close to the optimal solutions. The SA starts with an initial solution that can be feasible, then a neighborhood solution is generated by moving in the search space. The selected operator is transferred to the tabu list for a specified number of iterations to prevent reaching a local minimum. In some cases, the tabu action can be selected if it leads to a better solution according to the specified criteria. There is a fitness function to

82. Cho and Kim, *op. cit.*

83. Fayez F Boctor, « Resource-constrained project scheduling by simulated annealing », *in: International Journal of Production Research* 34.8 (1996), pp. 2335–2351.

84. Bouleimen and Lecocq, *op. cit.*

85. Sakalauskas and Felinskas, *op. cit.*

86. Partha Pratim Das and Sriyankar Acharyya, « Simulated annealing variants for solving resource constrained project scheduling problem: A comparative study », *in: 14th International Conference on Computer and Information Technology (ICCIT 2011)*, IEEE, 2011, pp. 469–474.

87. Nai-Hsin Pan and Yung-Yu Lin, « Using hybrid simulated annealing algorithm in resource constrained project scheduling problem », *in: Journal of Statistics and Management Systems* 14.3 (2011), pp. 555–582.

evaluate the solutions. The process continues until the termination conditions such as the number of iterations^{88 89}. The related work is listed below.

A tabu search method for minimizing the time span of a project was presented. In the proposed algorithm, the neighbor solution is obtained by a single swap or insert operation, and the tabu state is also repeatedly updated⁹⁰. An improved tabu search has been proposed, changing the way an initial solution is found instead of the classical tabu search. The objective is to solve the problems to minimize the project duration⁹¹. The proposed tabu search algorithm uses prioritization of activities to obtain the initial solution according to the slack time, while anticipation of activities and partial allocation of resources are not allowed⁹². For tabu search, two non-identical approaches were used to generate neighborhoods. The first is based on sharing resources assigned to a pair of tasks, the second on assigning any resource that could perform an identified task⁹³. An improved TS with four neighborhood operators has been proposed. The operators are named swap operation, insertion operation, exchange operation, and shift operation. In addition, two mutation operators are used in the procedure⁹⁴.

1.2.7 RCPSP and teaching–learning-based optimization (TLBO)

Teaching-learning-based optimization (TLBO) is a meta-heuristic technique inspired by the teaching-learning phenomenon that the best student can be a teacher to others. There are different types of TLBO, but a standard method consists of two phases. The first is the teacher phase and the second is the student phase. The algorithm starts with a population of solutions that are randomly initialized according to the given parameters. Then the best solution takes the role of teacher for the other solutions. Each student

88. Paul R Thomas and Said Salhi, « A tabu search approach for the resource constrained project scheduling problem », *in: Journal of heuristics* 4.2 (1998), pp. 123–139.

89. Potvin and Gendreau, *op. cit.*

90. Thomas and Salhi, *op. cit.*

91. Nai-Hsin Pan, Po-Wen Hsaio, and Kuei-Yen Chen, « A study of project scheduling optimization using Tabu Search algorithm », *in: Engineering Applications of Artificial Intelligence* 21.7 (2008), pp. 1101–1112.

92. Omer Atli, « Tabu search and an exact algorithm for the solutions of resource-constrained project scheduling problems », *in: International Journal of Computational Intelligence Systems* 4.2 (2011), pp. 255–267.

93. Marek E Skowronski et al., « Tabu search approach for multi-skill resource-constrained project scheduling problem », *in: 2013 Federated Conference on Computer Science and Information Systems*, IEEE, 2013, pp. 153–158.

94. Lirong Han et al., « Immunomodulatory activity of a water-soluble polysaccharide obtained from highland barley on immunosuppressive mice models », *in: Food & function* 10.1 (2019), pp. 304–314.

interacts with the teacher to generate new solutions. The defined fitness function evaluates the solutions to find a new teacher. The process runs until the defined termination condition, which can be the number of generations^{95 96}. Below we give a brief overview of related work.

A coevolutionary TLBO algorithm has been proposed to solve the stochastic resource-constrained project scheduling problem. In the method, the activity list is used for encoding and resource-based policies are used for decoding. Also, a new method is developed to select the best solution for detecting the teacher of each class. In the proposed algorithm, there are two initialized classes at the beginning of the process⁹⁷. For RCPSP, a TLBO algorithm with ordinal interval numbers was developed to minimize the project duration. In the process, a parallel schedule generation scheme based on ordinal intervals is used to generate feasible schedules. Two phases of self-study and testing are embedded in the algorithm, which improves the teaching-learning process⁹⁸. A (TLBO) was developed to solve the multi-skill resource-constrained project scheduling problem (MS-RCPSP). A task-resource list-based encoding scheme is proposed to generate feasible scheduled. Also, a reinforcement phase is incorporated into the TLBO algorithm to minimize the project duration⁹⁹. The objective of the proposed teaching-learning based optimization was to minimize the total project duration. In the proposed algorithm, there are two phases of self-study and testing to enhance the performance of TLBO¹⁰⁰.

1.2.8 RCPSP and evolutionary algorithms (EA)

Evolutionary algorithms are population-based metaheuristics that often first randomly generate solutions, such as genetic algorithms. Then, the solutions are evaluated against

95. R Venkata Rao, Vimal J Savsani, and DP Vakharia, « Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems », *in: Computer-aided design* 43.3 (2011), pp. 303–315.

96. Dheeraj Joshi et al., « An effective teaching-learning-based optimization algorithm for the multi-skill resource-constrained project scheduling problem », *in: Journal of Modelling in Management* (2019).

97. Huan-yu Zheng, Ling Wang, and Sheng-yao Wang, « A co-evolutionary teaching-learning-based optimization algorithm for stochastic RCPSP », *in: 2014 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2014, pp. 587–594.

98. Huan-yu Zheng and Ling Wang, « An effective teaching-learning-based optimisation algorithm for RCPSP with ordinal interval numbers », *in: International Journal of Production Research* 53.6 (2015), pp. 1777–1790.

99. Huan-yu Zheng, Ling Wang, and Xiao-long Zheng, « Teaching-learning-based optimization algorithm for multi-skill resource constrained project scheduling problem », *in: Soft Computing* 21.6 (2017), pp. 1537–1548.

100. Joshi et al., *op. cit.*

a specified fitness function to select the appropriate solutions for the next generation. There are also improvement methods that attempt to improve the solutions during the algorithm¹⁰¹. The corresponding works are listed below.

An evolutionary multi-agent algorithm has been proposed to deal with the RCPSP in which three operators-competition, crossover, and self-learning-are used to achieve the objective¹⁰². An evolutionary algorithm has been proposed that benefits from conglomerate-based crossover. This type of crossover combines the good parts of the solutions¹⁰³. The presented algorithm is also based on a differential evolutionary algorithm, with the addition of local search to improve the performance of the algorithm. The goal of the method is to minimize the duration and total cost of the project¹⁰⁴. A differential evolution algorithm for multi-skill RCPSP was proposed using a reassignment function to improve the solution quality at the end of each iteration¹⁰⁵.

1.2.9 RCPSP and hybrid algorithms (HA)

A large number of algorithms have been developed that do not purely follow the concepts of a single traditional metaheuristic, but represent a combination of different algorithmic ideas. The idea behind hybridizing different algorithms is to achieve better performance on complex problems. Unfortunately, developing an effective hybrid approach is generally a challenging task¹⁰⁶.

A hybrid algorithm, which is a combination of ant colony optimization (ACO), genetic algorithm (GA), and local search method, was used for the resource-constrained project scheduling problem¹⁰⁷. A hybrid of ACO and PSO algorithms was developed for

101. Gendreau, Potvin, et al., *op. cit.*

102. Xiaoying Pan and Hao Chen, « A multi-agent social evolutionary algorithm for resource-constrained project scheduling », *in: 2010 International Conference on Computational Intelligence and Security*, IEEE, 2010, pp. 209–213.

103. Francisco Ballestin, Agustin Barrios, and Vicente Valls, « An evolutionary algorithm for the resource-constrained project scheduling problem with minimum and maximum time lags », *in: Journal of scheduling* 14.4 (2011), pp. 391–406.

104. Arian Eshraghi, « A new approach for solving resource constrained project scheduling problems using differential evolution algorithm », *in: International journal of industrial engineering computations* 7.2 (2016), pp. 205–216.

105. Huu Dang Quoc et al., « New effective differential evolution algorithm for the project scheduling problem », *in: 2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*, IEEE, 2020, pp. 150–157.

106. Potvin and Gendreau, *op. cit.*

107. Lin-Yu Tseng and Shih-Chieh Chen, « A hybrid metaheuristic for the resource-constrained project scheduling problem », *in: European Journal of Operational Research* 175.2 (2006), pp. 707–721.

the RCPSP to optimize the project duration with minimum lag times¹⁰⁸. A hybrid of genetic algorithm and simulated annealing was developed for the RCPSP to improve the performance of the procedure. In the procedure, GA generates the temporary population and SA tries to improve the responses¹⁰⁹. A hybrid algorithm based on particle swarm optimization and a genetic algorithm was developed to deal with the RCPSP and minimize the project duration¹¹⁰. A hybrid of ant colony optimization and scatter search was used for the problem. In the algorithm, the ACO explores the solution space and generates an activity list, and then the SS algorithm tries to improve the answers¹¹¹. A neurogenetic approach was developed, which is a combination of a genetic algorithm and a neural network. The GA performs the process of global search and the NN performs the process of local search¹¹². A hybrid algorithm that is a combination of simulated annealing, tabu search, and genetic algorithm was developed to minimize project duration¹¹³. A hybrid algorithm was proposed combining artificial bee colony (ABC) with genetic algorithm (GA), whose objective is to minimize project duration¹¹⁴. A hybrid strategy based on genetic algorithms and simulated annealing was developed to deal with the RCPSP. The strategy aims to combine the parallel search capacity of genetic algorithms with the finetuning capabilities of simulated annealing to create an algorithm for the RCPSP¹¹⁵. A hybrid ant colony optimization (HAntCO) approach for solving multi-skill RCPSP was presented. The proposed approach combines classical heuristic priority rules for project scheduling with ant colony optimization (ACO). In the method, an approach for updat-

108. Miyuan Shan, Juan Wu, and Danni Peng, « Particle swarm and ant colony algorithms hybridized for multi-mode resource-constrained project scheduling problem with minimum time lag », *in: 2007 International Conference on Wireless Communications, Networking and Mobile Computing*, IEEE, 2007, pp. 5898–5902.

109. Xiaoguang Yu et al., « A novel genetic simulated annealing algorithm for the resource-constrained project scheduling problem », *in: 2009 International Workshop on Intelligent Systems and Applications*, IEEE, 2009, pp. 1–4.

110. Ming Li et al., « A particle swarm optimization algorithm with crossover for resource constrained project scheduling problem », *in: 2009 IITA International Conference on Services Science, Management and Engineering*, IEEE, 2009, pp. 69–72.

111. Wang Chen et al., « An efficient hybrid algorithm for resource-constrained project scheduling », *in: Information Sciences* 180.6 (2010), pp. 1031–1039.

112. Anurag Agarwal, Selcuk Colak, and Selcuk Erenguc, « A neurogenetic approach for the resource-constrained project scheduling problem », *in: Computers & operations research* 38.1 (2011), pp. 44–50.

113. Arit Thammano and Ajchara Phu-Ang, « A hybrid evolutionary algorithm for the resource-constrained project scheduling problem », *in: Artificial Life and Robotics* 17.2 (2012), pp. 312–316.

114. Vahid Zeighami et al., « An ABC-Genetic method to solve resource constrained project scheduling problem », *in: Artificial Intelligence Research (AIR) Journal, SCIEDU, Canada* 1.2 (2012), pp. 185–197.

115. Bettemir and Sonmez, *op. cit.*

ing pheromone values based on both best and worst solutions¹¹⁶ was proposed. A hybrid algorithm is proposed that is a combination of a greedy search and a genetic algorithm to minimize the project duration¹¹⁷. A hybrid genetic algorithm was developed using the SA algorithm as the GA operator to maximize the NPV of the project¹¹⁸. A PSO-based algorithm was proposed to work with the mutation operator and the algorithm benefits from forward-backward improvement method and local search process¹¹⁹. A hybrid approach, which is a combination of tabu search and simulated annealing algorithm, was developed to minimize the project duration¹²⁰. The proposed hybrid algorithm is a combination of differential evolution algorithm and cuckoo search algorithm. Also, a local forward-backward improvement is used to improve the new solutions¹²¹. A hyper method, called self-adaptive differential evolution, has been proposed for fussy stochastic RCPSP. In the proposed approach, the activity durations and project duration are fuzzily estimated and randomization is also used to generate new solutions through the operators of mutation and crossover¹²².

1.2.10 RCPSP and the proposed algorithms based on other metaheuristics

A social evolutionary multi-agent algorithm has been proposed in which agents behave in three ways: competition, crossover, and self-learning. The method was proposed to optimize the project duration¹²³. A distribution estimation algorithm was proposed to deal

116. Paweł B Myszkowski et al., « Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem », *in: Soft Computing* 19.12 (2015), pp. 3599–3619.

117. Aidin Delgoshaei et al., « Minimizing makespan of a resource-constrained scheduling problem: A hybrid greedy and genetic algorithms », *in: International Journal of Industrial Engineering Computations* 6.4 (2015), pp. 503–520.

118. Fatemeh Fathollahi and AA Naja, « A hybrid genetic algorithm to maximize net present value of project cash flows in resource-constrained project scheduling problem with fuzzy parameters », *in: Scientia Iranica. Transaction E, Industrial Engineering* 23.4 (2016), p. 1893.

119. M Munlin and M Anantathanavit, « Hybrid radius particle swarm optimization », *in: 2016 IEEE Region 10 Conference (TENCON)*, IEEE, 2016, pp. 2180–2184.

120. Behrouz Afshar-Nadjafi, Mehdi Yazdani, and Mahyar Majlesi, « A hybrid of tabu search and simulated annealing algorithms for preemptive project scheduling problem », *in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, 2017, pp. 102–111.

121. Karam M Sallam, Ripon K Chakraborty, and Michael J Ryan, « A hybrid differential evolution with cuckoo search for solving resource constrained project scheduling problems », *in: 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, IEEE, 2019, pp. 1344–1348.

122. Yagub Alipouri et al., « Solving the FS-RCPSP with hyper-heuristics: A policy-driven approach », *in: Journal of the Operational Research Society* 70.3 (2019), pp. 403–419.

123. Pan and Chen, *op. cit.*

with the RCPSP. Here, the solutions are generated by the latest finish time (LFT) priority rule and the random method, and then decoded using the serial schedule generation scheme¹²⁴. An artificial immune algorithm was used to solve the RCPSP to optimize the project duration. The method uses two types of mutation operations to generate new generations or solutions¹²⁵. A distribution estimation algorithm was proposed to optimize the project duration. This algorithm uses a local search operator and a forward-backward iteration method to improve the solutions¹²⁶. A firefly algorithm was employed to minimize the project duration¹²⁷ and also a distribution estimation algorithm (DEA) with a binary random variable matrix was adopted to deal with the RCPSP¹²⁸.

1.2.11 RCPSP and the proposed algorithms based on reinforcement learning

The strategies for the A-team with reinforcement learning were proposed to solve the RCPSP. An A-team is defined as a set of objects consisting of multiple agents and a shared memory that generate solutions to optimization problems through interactions. The interactions that lead to solutions are usually managed by a static strategy. In research, dynamic learning strategies based on reinforcement learning are proposed to monitor the interactions between the optimization agents and the common memory¹²⁹. A deep reinforcement learning algorithm has been proposed to formulate resource allocation as a Markov decision process and find the best resource allocation policy. The proposed research addresses the resource allocation for a project where there are iterations between the activities of the project and also, there is a method to shorten the duration of an

124. Chen Fang, Ling Wang, and Ye Xu, « An estimation of distribution algorithm for resource-constrained project scheduling problem », *in: 2010 Chinese Control and Decision Conference*, IEEE, 2010, pp. 265–270.

125. Mahdi Mobini, Zahra Mobini, and Masoud Rabbani, « An Artificial Immune Algorithm for the project scheduling problem under resource constraints », *in: Applied soft computing* 11.2 (2011), pp. 1975–1982.

126. Ling Wang and Chen Fang, « A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem », *in: Expert Systems with Applications* 39.3 (2012), pp. 2451–2460.

127. Pejman Sanaei et al., « Using firefly algorithm to solve resource constrained project scheduling problem », *in: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Springer, 2013, pp. 417–428.

128. Chen Fang et al., « An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem », *in: Flexible Services and Manufacturing Journal* 27.4 (2015), pp. 585–605.

129. Piotr Jędrzejowicz and Ewa Ratajczak-Ropel, « Reinforcement learning strategies for A-team solving the resource-constrained project scheduling problem », *in: Neurocomputing* 146 (2014), pp. 301–307.

activity by incorporating additional resources which called crashing¹³⁰. A reinforcement learning-based meta-heuristic switching approach has been introduced to solve these extended RCPSPs that leverages the capabilities of both multi-operator differential evolution (MODE) and discrete cuckoo search (DCS) algorithms in a single algorithm. In research, reinforcement learning (RL) is proposed as a method to choose either MODE or DCS based on the diversity of the population and the quality of the solutions¹³¹.

1.2.12 RCPSP and the proposed algorithms based on neural networks

Augmented neural networks was introduced to solve the task scheduling problem. The proposed algorithm is a hybrid of the heuristics and neural networks approach. The objective of the problem is to minimize the project duration when scheduling n jobs or activities on m machines, where the activities follow a predefined priority relationship and task pre-emption is not allowed¹³². A neurogenetic approach has been developed, which is a hybrid of neural networks and genetic algorithms. In the proposed approach, the search operation depends on GA iterations for global search and NN iterations for local search. The NN and GA search iterations are interleaved in such a way that the NN can select the best solution from the GA pool. Also, good solutions obtained by the NN search are included in the GA population to use the GA iterations¹³³. An artificial neural network proposed to schedule 240 projects such as residential, office, school, etc. is designed and programmed considering limited resources. Three priority rules named latest finish time, minimum slack time and maximum remaining path length are used to determine the resource sets of these projects that have the highest performance according to the literature, in the sets of 2, 4, 6 and 8¹³⁴.

130. Inkyung Sung, Bongjun Choi, and Peter Nielsen, « Reinforcement Learning for Resource Constrained Project Scheduling Problem with Activity Iterations and Crashing », *in: IFAC-PapersOnLine* 53.2 (2020), pp. 10493–10497.

131. Karam M Sallam, Ripon K Chakraborty, and Michael J Ryan, « A reinforcement learning based multi-method approach for stochastic resource constrained project scheduling problems », *in: Expert Systems with Applications* 169 (2021), p. 114479.

132. Anurag Agarwal, Hasan Pirkul, and Varghese S Jacob, « Augmented neural networks for task scheduling », *in: European Journal of Operational Research* 151.3 (2003), pp. 481–502.

133. Agarwal, Colak, and Erenguc, *op. cit.*

134. Ömer Özkan and Ümit Gülçiçek, « A neural network for resource constrained project scheduling programming », *in: Journal of Civil Engineering and Management* 21.2 (2015), pp. 193–200.

1.3 Conclusion

Heuristics are techniques that begin with an initially empty schedule in which no activities have yet been scheduled. The empty schedule is populated with activities at each step based on defined priority rules and schedule creation schemes. Activities are ordered and scheduled based on their assigned values using priority rules. Metaheuristic methods typically start with an initial solution set composed of the first generations. The algorithms continuously apply a series of operations to improve and evolve the solutions¹³⁵.

There are many heuristics and metaheuristics to deal with the resource-constrained project scheduling problem (RCPSP), but they have difficulty optimizing in real projects¹³⁶. As mentioned in the previous sections, after the ineffectiveness of the exact method to solve large RCPSP, researchers have used heuristics and metaheuristics for large problems. But the common metaheuristics, such as the explained approaches, may not be efficient enough given the limitations of these types of algorithms. Therefore, researchers are developing more effective algorithms. These methods start with one solution or a set of solutions as a population and then try to generate more suitable solutions during the process. Thus, these methods try to improve the solutions by creating a group of solutions or neighborhoods for the next generations of algorithms, because these methods work based on search and repeated generations. Moreover, the optimal or near-optimal solution at the end of the algorithm depends on the quality of the initial solutions^{137 138}.

As I explained in the introduction, there are two main constraints: precedence constraints and resource constraints. Thus, it is important to obtain a feasible activity list or solution that takes the constraints into account, because without taking the constraints into account, the solution is wrong. Therefore, these methods use rules such as priority rules to obtain feasible solutions¹³⁹. There are algorithms that generate the solutions and then check their feasibility. In this way, they have to change the placement of the activities in the activity list to get a feasible schedule. The procedures use the operators to improve the solutions after the solutions are initialized. However, one of the difficulties with these methods is tuning the parameters of the algorithms. For example, in a simulated annealing

135. Bidisha Roy and Asim Kumar Sen, « A novel metaheuristic approach for resource constrained project scheduling problem », in: *Soft Computing: Theories and Applications*, Springer, 2020, pp. 535–544.

136. Koulinas, Kotsikas, and Anagnostopoulos, *op. cit.*

137. Jędrzejowicz and Ratajczak-Ropel, *op. cit.*

138. Golab et al., *op. cit.*

139. Alipouri et al., *op. cit.*

algorithm, it is important to tune the parameters for the initial temperature, the number of neighboring solutions, and the attenuation factor of the temperature in each step to make the SA work efficiently.

In addition, setting an appropriate termination condition for these algorithms can affect the final answer. The termination conditions may be the total time of procedure execution, the total number of neighbor solutions generated, or a defined condition for the objective value that should be tested by experiments obtained^{140 141}. Moreover, these algorithms are not always successful in finding the near optimum or the optimum. Sometimes they discover a local solution, or the algorithms even fail to repeat the procedure when the algorithm is run. For example, a meta-heuristic may obtain the optimal solutions for the 89 instances by running 120 instances of¹⁴².

From our point of view, the generation of an optimal or near-optimal activity sequence is the key to an appropriate method. Therefore, using new approaches such as neural networks based on learning and prediction may be more useful than traditional metaheuristics to achieve the objective. Therefore, in this thesis, I focus on the algorithms that benefit from neural network approaches.

140. Bouleimen and Lecocq, *op. cit.*

141. Golab et al., *op. cit.*

142. Kadri and Boctor, *op. cit.*

OVERVIEW OF NEURAL NETWORKS, INPUTS AND OUTPUTS

In this chapter, readers are given a brief explanation of neural networks. In what follows, I focus on two types of neural networks, namely multilayer feed-forward neural networks and convolutional neural networks. The reason for focusing on these two types is that I benefit from these neural networks in Chapters 3 and 4. In addition, the eight inputs of the proposed networks are explained in Chapters 3 and 4, which are referred to as project parametric characterizations. Finally, the last subsection of this chapter describes the priority rules that are the outputs of the developed neural networks.

2.1 Overview of neural networks

Machine learning (ML) is a subfield of artificial intelligence (AI) and artificial neural networks (ANN) are flexible mathematical models and powerful machine learning methods originally inspired by the functional structure of the human brain. The human nervous system contains cells called neurons that are interconnected. These networks take over the learning mechanism in the biological organism, where billions of interconnected neurons process data in parallel. This mechanism is simulated in artificial neural networks. Artificial neural networks consist of computational units called neurons. These computational units or neurons process data while connected to each other by adjustable weight connections. Each input to a neuron is scaled by a weight that affects the function being computed in that unit or neuron. An artificial neural network (ANN) computes a function of the inputs by propagating the evaluated values from the input neurons to the output neurons, using the weights as middle parameters. The learning process takes place by changing the weights that connect the neurons.

The training process provides feedback on the accuracy of the computed weights in the neural network based on how well the predicted output for a given input is labeled in the

training data. Similarly, weights are adjusted between computational units or neurons in a neural network in response to prediction errors. The goal of changing the weights is to adjust the computed function to make the predictions more accurate. Therefore, weights are carefully changed in a mathematical way to reduce computational errors. Artificial neural networks (ANNs) are popular methods and helpful models for optimization, prediction, classification, clustering, etc.^{1 2 3 4}

Artificial neural networks are divided into two groups: feed-forward neural networks and feed-backward neural networks. In this work, two feed-forward neural networks are used. A feed-forward neural network is an algorithm that consists of ordered layers similar to the neural processing units of the human brain. Each processing unit or neuron in one layer is connected to the other neurons in the other layers of neural network. In a feed-forward NN, data enters from the inputs and flows from one layer to the next layer it reaches the output units. In feed-forward NNs, the information is transmitted in one direction, from the input neurons to the hidden neurons and then to the output neurons. Therefore, they are also called feed-forward neural networks. Each connection between computational units can have a different weight, so the connections between neurons are not all the same. The weights of the network connections control the possible part of the knowledge of the network^{5 6 7}.

A crucial feature of neural networks is the adaptation of the network size to a given problem. The network size specifies the number of layers, the number of neurons or nodes per layer, and the number of connections between neurons. The accuracy of prediction in a neural network also depends on the type of activation function or transfer function used in the NN. Activation functions are employed to calculate the weighted sum of inputs and biases. Activation functions are used to control the outputs of neural networks in various domains. Activation functions help in learning complicated mappings between inputs and corresponding outputs. The activation functions also redirect the input signals to the

1. Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal, « Introduction to multi-layer feed-forward neural networks », *in: Chemometrics and intelligent laboratory systems 39.1* (1997), pp. 43–62.

2. Sun-Chong Wang, « Artificial neural network », *in: Interdisciplinary computing in java programming*, Springer, 2003, pp. 81–100.

3. Charu C Aggarwal et al., « Neural networks and deep learning », *in: Springer 10* (2018), pp. 978–3.

4. Rene Y Choi et al., « Introduction to machine learning, neural networks, and deep learning », *in: Translational Vision Science & Technology 9.2* (2020), pp. 14–14.

5. Oludare Isaac Abiodun et al., « State-of-the-art in artificial neural network applications: A survey », *in: Heliyon 4.11* (2018), e00938.

6. Aggarwal et al., *op. cit.*

7. Choi et al., *op. cit.*

output signals. Therefore, these 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 methods to reduce the errors^{8 9 10}.

In my research, I use two types of neural networks, the multilayer neural network and the convolutional neural network 1D. Both algorithms are subsets of feed-forward neural networks. They are employed to predict the corresponding outputs.

Unlike single-layer neural networks, which contain a single input layer and an output node, multilayer neural networks contain more than one computational layer that benefits from activation functions. Multilayer neural networks usually consist of the input layer, the output layer, and the layers in between, which are called hidden layers because they are not visible. In other words, there are multiple computational layers in multilayer neural networks, and the additional layers between input and output are called hidden layers. In these types of NNs, the input layer passes the data to the output layer. As mentioned earlier, multilayer neural networks are feed-forward networks because in a multilayer feed-forward network (MLFNN), all nodes in one layer are connected to those in the next layer. In simple multilayer neural networks, there are one to three hidden layers, but deep neural networks can include dozens or hundreds of hidden layers. Multilayer feedforward neural networks (MLFNN).

A convolutional neural network (CNN) is a type of feedforward neural network for solving the problem. A convolutional neural network (CNN) is one of the most remarkable neural networks in the field of Deep Learning that has the ability to extract features from data with convolutional structures. In a CNN, each individual neuron benefits from local connections, meaning that each neuron is not connected to all the neurons or computational units in the previous layer, but only to a small number of neurons. This helps to reduce parameters and speed up convergence. There is also weight sharing between a group of connections. Weight sharing can lead to a further reduction of the parameters. In a convolutional neural network, there are pooling layers that are able to reduce the amount of data while preserving useful information by reducing the dimensions (down

8. Sagar Sharma, Simone Sharma, and Anidhya Athaiya, « Activation functions in neural networks », *in: towards data science 6.12* (2017), pp. 310–316.

9. Aggarwal et al., *op. cit.*

10. Chigozie Nwankpa et al., « Activation functions: Comparison of trends in practice and research for deep learning », *in: arXiv preprint arXiv:1811.03378* (2018).

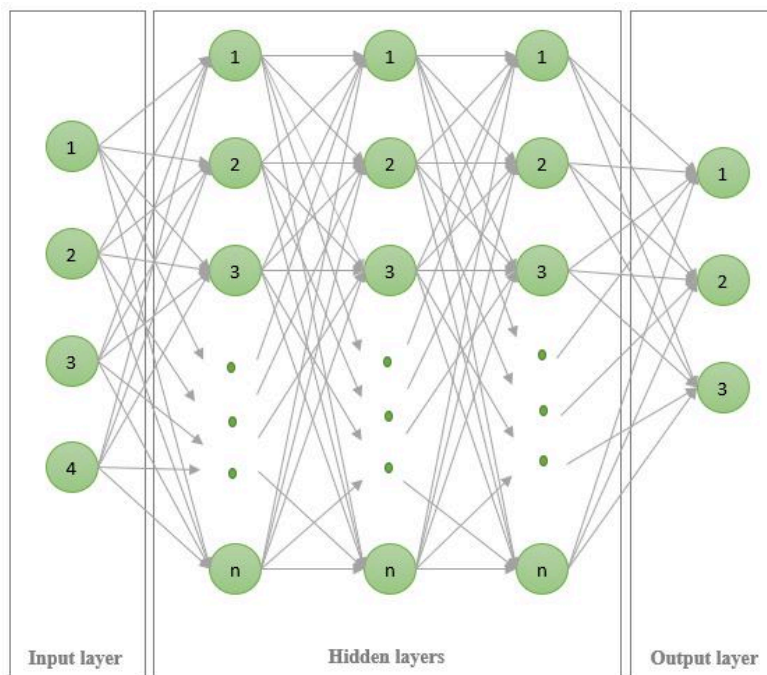


Figure 2.1 – An example of a multilayer feed-forward neural network with an input layer, three hidden layers, and an output layer.

sampling). This phenomenon can reduce the number of parameters by removing trivial features ¹¹.

Data is input directly into a convolutional neural network (CNN) and then processed by convolutional layers and pooling layers. The processed data feeds into one or more fully linked layers, as in a normal multilayer neural network. Finally, the final output of the fully connected layer is the desired output ^{12 13}.

As mentioned earlier, a CNN consists of three types of layers: the convolutional layer, the pooling layer, and the fully connected layer. The main design unit of a convolutional neural network is its convolutional structure. The convolutional structure in a CNN 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 with this filter. The CNN can quickly learn

11. Zewen Li et al., « A survey of convolutional neural networks: analysis, applications, and prospects », *in: IEEE transactions on neural networks and learning systems* (2021).

12. Waseem Rawat and Zenghui Wang, « Deep convolutional neural networks for image classification: A comprehensive review », *in: Neural computation* 29.9 (2017), pp. 2352–2449.

13. Golab et al., *op. cit.*

the suitable 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 computations and parameters in the model. This avoids the problem of overfitting and increases the speed of the calculations. The most common form of pooling layer is the so-called max-pooling. The final part of a CNN is the fully connected layers that are the standard neural network or the multilayer neural network that leads to predictions or classifications. The fully connected structure obtains the full connections through each neuron in that part of the network^{14 15 16 17}.

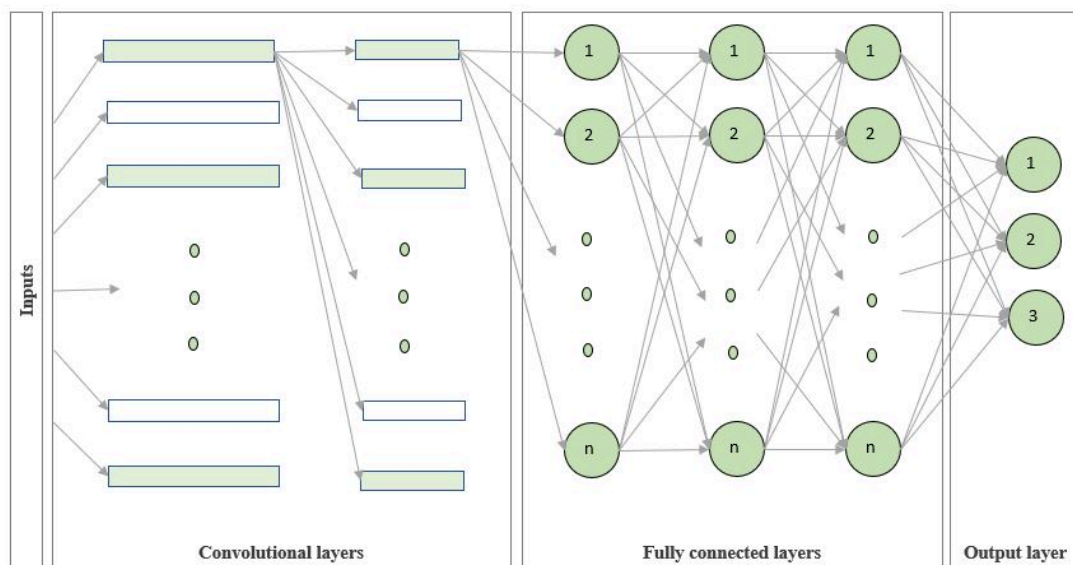


Figure 2.2 – An example of a convolutional neural network (CNN) with two convolutional layers, three fully connected layers, and an output layer.

In general, CNNs are trained in a supervised manner by the so-called backpropagation algorithm (BP), which includes two phases. The first phase of the BP algorithm is called the forward phase and the second phase is called the backpropagation phase. In the forward phase, the activation functions are propagated from the input layer to the output layer, and in the backward phase, the identified errors between the detected actual value

14. Rawat and Wang, *op. cit.*

15. Neena Aloysius and M Geetha, « A review on deep convolutional neural networks », *in: 2017 international conference on communication and signal processing (ICCSP)*, IEEE, 2017, pp. 0588–0592.

16. Anamika Dhillon and Gyanendra K Verma, « Convolutional neural network: a review of models, methodologies and applications to object detection », *in: Progress in Artificial Intelligence 9.2* (2020), pp. 85–112.

17. Li et al., *op. cit.*

and the target value are backpropagated in the output layer to update the weights and bias values^{18 19}.

2.2 The inputs: project parametric characterizations

There are three project parametric characterizations, called network complexity (NC), resource factor (RF), and resource strength (RS). These parameters were re-explained by Kolisch et al.²⁰ and they illustrate the structure of a project network. These parameters describe the constraints of the average number of successors per unscheduled activity, the average proportion of resources requested per activity, and the measure of resource availability during the project, respectively²¹. However, these parameters are used to calculate the factors of a complete project, but can be modified and used to equip the scenarios of partial schedules. In addition to the three parameters illustrated, we also employ other parameters, namely average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of successors remaining, and average units per day, which can be calculated using the project data. They can also be recalculated to equip the partial schedule scenarios^{22 23 24}. The elements used in the formulas are defined in the table 2.2.

$$\text{Network complexity (NC)} = \frac{\sum_{i \in US} S_i}{|US|} \quad (2.1)$$

$$\text{Resource factor (RF)} = \frac{1}{|US|} \frac{1}{|R|} \sum_{i \in US} \sum_{r \in R} \begin{cases} 1 & \text{if } u_{ir} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

18. Aggarwal et al., *op. cit.*

19. Serkan Kiranyaz et al., « 1D convolutional neural networks and applications: A survey », *in: Mechanical systems and signal processing* 151 (2021), pp. 107–398.

20. Rainer Kolisch, Arno Sprecher, and Andreas Drexler, « Characterization and generation of a general class of resource-constrained project scheduling problems », *in: Management science* 41.10 (1995), pp. 1693–1703.

21. Rainer Kolisch and Arno Sprecher, « PSPLIB—a project scheduling problem library: OR software-ORSEP operations research software exchange program », *in: European journal of operational research* 96.1 (1997), pp. 205–216.

22. Golab et al., « Investigating the performance of an artificial neural network for solving the resource constrained project scheduling problem (RCPSP) ».

23. Golab et al., « A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP) ».

24. *Idem*, « A convolutional neural network for the resource-constrained project scheduling problem (RCPSP): A new approach ».

Item	Definition
US	Set of unscheduled activities
A	Set of project activities
R	Set of renewable resources
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_i	Work-content of activity i calculated by $w_i = d_i * u_{ir}$
W	Work-content of the project
S_i	Number of the immediate successors of activity i
d_i	Duration of activity i
u_{ir}	Amount of renewable resource r consumed by activity i

Table 2.1 – The definitions of the employed items in the formulas' parameters.

$$Resource\ strength\ (RS) = \sum_{r \in R} \frac{U_r - U_r^{max}}{U_r^{max} - U_r^{min}} \quad (2.3)$$

$$Average\ work\ per\ activity\ (AWA) = \frac{\sum_{i \in US} w_i}{|US|} \quad (2.4)$$

$$Percentage\ of\ remaining\ work\ (PRW) = \frac{\sum_{i \in US} w_i}{W} \times 100 \quad (2.5)$$

$$Percentage\ of\ unscheduled\ activities\ (PUA) = \frac{|US|}{|A|} \times 100 \quad (2.6)$$

$$Percentage\ of\ remaining\ successors\ (PRS) = \frac{\sum_{i \in US} S_i}{\sum_{i \in A} S_i} \times 100 \quad (2.7)$$

$$Average\ units\ per\ day\ (AUD) = \frac{\sum_{i \in US} w_i}{\sum_{i \in US} d_i} \quad (2.8)$$

The eight parameters introduced are recalculated at each stage of project scheduling

to characterize the existing subproject. This means that after an activity is scheduled, these parameters are recalculated to characterize the new step or subproject. In Chapter 3 and 4, I will explain how these parameters are the inputs of the developed multilayer feed-forward neural network (MLFNN) and the convolutional neural network (CNN).

2.3 The outputs: Priority rules

There are rules, called priority rules, that are used to select an eligible activity from the available decision set or list of eligible activities. Priority rules are the methods used to select activities based on their selection criteria for project scheduling. Therefore, a priority rule assigns values to the activities in the eligible activities list or decision set. The values assigned to the activities are used to filter out an activity according to the selection criteria, which may be the minimum or maximum value. It is obvious that these priority rules lead to different consequences depending on the project specifications and the existing project conditions, such as the number of resources, the number of relations between activities, etc. Therefore, it is important to select an appropriate priority rule during project scheduling that selects an activity from the eligible activities. The selection of different priority rules, as explained in the introduction, leads to the selection of different activities, which may affect the project duration. If there is more than one activity with the same assigned value, it is possible to randomly select the activity or filter out the activity with the smallest activity label. As presented in table 2.3, there are different priority rules that can be classified according to the selection criteria. These priority rules are the outputs of the evolved neural networks. The different priority rules and their formulas are explained below^{25 26 27 28 29 30 31}.

The early start time (EST) and early finish time (EFT) of the activity are calculated using the forward pass technique. They specify the earliest start time and the earliest

25. Olaguibel and Goerlich, *op. cit.*

26. Gündüz Ulusoy and Linet Özdamar, « Heuristic performance and network/resource characteristics in resource-constrained project scheduling », *in: Journal of the Operational Research Society* 40.12 (1989), pp. 1145–1152.

27. Rainer Kolisch, « Efficient priority rules for the resource-constrained project scheduling problem », *in: Journal of Operations Management* 14.3 (1996), pp. 179–192.

28. Kolisch and Hartmann, *op. cit.*

29. Özkan and Gülçiçek, *op. cit.*

30. Golab et al., *op. cit.*

31. Golab et al., « A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP) ».

finish time that an activity can start and finish in a project. These two priority rules are calculated using the following formulas.

$$EST_i = \text{maximum finish time of the set of all immediate predecessors activity } i + 1 \quad (2.9)$$

$$EFT_i = EST_i + d_i \quad (2.10)$$

The backward pass method is used to determine the latest start time (LST) and the latest finish time (LFT). These are the latest times an activity can start or terminate before it causes a delay in the project deadline. Forward pass and backward pass techniques are the methods of the critical path method (CPM).

$$LFT_i = \text{latest start time of all successors} - 1 \quad (2.11)$$

$$LST_i = LFT_i - d_i + 1 \quad (2.12)$$

The shortest processing time (SPT) is another priority rule that selects an activity with the shortest duration among the eligible activities. This means that the activity that takes the least amount of time is executed first.

$$SPT_i = d_i \quad (2.13)$$

The priority rule of total resource demand (TRD) selects an activity that requires the least sum of resource units r . This means that the activity that requires the least amount of resource units is implemented first.

$$TRD_i = \sum_r u_{ir} \quad (2.14)$$

The total resource scarcity (TRS) is another priority rule, which is the sum of the units of resource r required by activity i divided by the capacity of resource r . The value of this rule is determined by formula 2.15.

$$TRS_i = \sum_r u_{ir}/U_r \quad (2.15)$$

The most total successor (MTS) is the next priority rule that aims to select an activity that has the most successors among the eligible activities, i.e., the activity with the most

total successors is executed first.

$$MTS_i = |\overline{S}_i| \quad , S \text{ is defined the successors of activity } i \quad (2.16)$$

The slack time or float time specifies the time that an activity can be delayed without affecting the project duration. Therefore, this priority rule selects an activity with minimum slack time. This means that the activity with the most slack time will be executed first. The value of the rule is determined by formula 2.17.

$$MSLK_i = LST_i + EST_i \quad (2.17)$$

Greatest rank positional weight (GRPW) is another priority rule which is calculated according to formula 2.18. According to its selection criteria, the activity with the highest value among the eligible activities is selected.

$$GRPW_i = p_i + \sum_{j \in S_i} p_j \quad (2.18)$$

The last rule, called the weighted resource utilization ration and precedence (WRUP), is also calculated according to formula 2.19. According to its selection criteria, the activity with the highest value among the eligible activities is selected.

$$WRUP_i = w|\overline{S}_i| + (1 - w)TRS_i \quad (2.19)$$

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

Table 2.2 – The priority rules and the selection criteria.

2.4 Conclusion

In this chapter, I briefly explained the neural networks, and then two types of neural network were presented separately. It was mentioned that feedforward multilayer neural networks consist of an input layer, hidden layers, and an output layer, while convolutional neural networks consist of convolutional layers, fully connected layers that are similar to multilayer networks, and an output layer. The eight project parameters that characterize the project and subprojects were described. In the last subsection, the outputs of the developed neural network were explained, which are called priority rules. The illustrated priority rules give a value to the eligible activities. The values given are used to select an eligible activity at each stage of the project schedule.

A MULTILAYER FEED-FORWARD NEURAL NETWORK (MLFNN) TO SOLVE THE RCPSP

This chapter focuses on solving the resource-constrained project scheduling problem (RCPSP) using a developed multilayer feed-forward neural network (MLFNN). As explained in the first chapter, the objective of the problem is to optimize and minimize the project duration considering the resource quantities and precedence constraints in project scheduling. It means that the objective is to minimize the project duration while scheduling the project activities. In this chapter, a multilayer feedforward neural network (MLFNN) is proposed to solve the standard single-mode RCPSP. The advantage of this method presented in this chapter over evolutionary methods or metaheuristics is that there is no need to generate numerous solutions or populations. The algorithm schedules the project activities step by step considering the output of the developed MLFNN. The developed MLFNN trains based on eight parameters explained in the previous chapter, namely network complexity, resource factor, resource strength, average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of remaining successors, and average units per day. The above parameters are calculated at each step of the project schedule to identify the project character at this stage of scheduling. Therefore, the developed MLFNN selects an appropriate priority rule, which is one of the predicted outputs of the neural network. Therefore, after the training process, the developed MLFNN can automatically select an appropriate priority rule to filter an unscheduled activity from the list of eligible activities and add it to the project schedule. In this way, the algorithm can schedule all activities of the project according to the given project constraints. In the second part of the chapter, I investigate the performance of the proposed approach using standard benchmark problems from the project schedule problems library (PSPLIB).

3.1 Algorithm and the developed MLFNN

The proposed algorithm benefits from a schedule generation scheme as a function to decode and generate a feasible schedule. In general, there are two major schedule generation schemes (SGS) for generating feasible schedules. The first is the so-called serial schedule generation scheme (SSGS) and the second is the so-called parallel schedule generation scheme (PSGS). To better understand the difference between these two schemes, the reader is referred to the article by Kolisch and Hartmann¹. In this chapter, I benefit from the serial schedule generation scheme (SSGS), which constructs active schedules that contain at least one feasible or optimal solution. In other words, the SSGS can generate a sequence of activities that leads to a feasible or optimal schedule because the serial schedule generation scheme always constructs feasible schedules². As mentioned earlier, I use the SSGS in this research for the decoding function.

The serial schedule generation scheme consists of n steps during project planning, where n refers to the steps, iterations of scheduling, and number of project activities. For example, for the project shown in Figure 7, there are 32 scheduling phases or stages. Therefore, these eight parameters are recalculated 32 times to characterize the new steps or new subprojects. This method benefits from three sets so-called the un-scheduled set US_n , the eligible set E_n , and the scheduled set S_n . These sets should be updated at each step or iteration of scheduling. These sets should be updated at each step or iteration of scheduling. Therefore, the unscheduled activities can be a member of the eligible set if their predecessors were scheduled, and an eligible activity can be a member of the scheduled activity if it is added to the project schedule.

Initialization in SSGS starts with a sub schedule consisting only of the dummy start activity at time zero. Then, at each iteration or step of the project schedule, an activity is selected from the eligible set E_n by the selected priority rule. In the proposed algorithm, the selection of a priority rule as an output is the task of the developed multilayer feed-forward neural network (MLFNN), which will be explained later. The selected priority rule then selects an eligible activity to be included in the project schedule at the earliest possible time that is feasible in terms of both precedence and resource availability. The process continues to schedule all activities of the project. For better understanding, I would like to refer the reader to the scheduling of the practical example in the introduction of the thesis. The practical example in the introduction is scheduled according to the SSGS, but

1. Kolisch and Hartmann, *op. cit.*

2. Koulinas, Kotsikas, and Anagnostopoulos, *op. cit.*

in this section I introduce a multilayer feed-forward neural network (MLFNN) to select an appropriate priority rule as an output. The selected priority rule can then filter out an activity from the eligible activities.

It has already been explained that multilayer feedforward neural networks (MLFNNs) consist of three types of layers. Moreover, each layer consists of a set of interconnected neurons or nodes that use activation functions. Moreover, the data or inputs are fed into the MLFNN through the input layer. As shown in Figures 3.1, 3.2, 3.3 3.4, and 3.5, the proposed MLFNN is fed with eight different data. The input layer consists of eight neurons or nodes that are fed with eight data called project parametric characterizations, fully explained in the project parametric characterizations section of the previous chapter. As mentioned, the inputs 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 eight parameters are recalculated at each step of project scheduling to characterize the new sub-project or step. In this way, the new inputs prepare the conditions for selecting a priority rule as an output to filter out an eligible activity for project scheduling.

The different number of hidden layers is used to create the interconnected neurons of the network. As shown in the result tables 3.1, 3.2, and 3.3, the evolved MLFNNs are designed with 1, 2, and 3, hidden layers respectively. The multilayer feed-forward neural network (MLFNN) is a back-propagation algorithm that consists of two phases. The first phase is called the forward phase and the second phase is called the backward phase. In the first phase, the activation functions are propagated from the input layer to the output layer. In the second phase, the errors between the observed actual value and the eligible nominal value in the output layer are propagated backward to improve the weights and bias values^{3 4}.

It was explained that if no activation function is embedded in a multilayer feedforward neural network (MLFNN), the output would be just a simple linear function. Therefore, activation is applied to make the network dynamic and give it the ability to extract complex data. To achieve this, two different activation functions are used in the developed MLFNN. The first activation function is called relu and is a nonlinear activation. This

3. Agarwal, Colak, and Erenguc, *op. cit.*

4. Aggarwal et al., *op. cit.*

function can be mathematically calculated as follows:

$$f(x) = \max(0, x) \tag{3.1}$$

and is widely used in neural networks, and I use this function for the hidden layers of the developed MLFNN. The second type of activation function is called sigmoid, which is widely used. This type of function is also nonlinear and converts the values in the range of 0 to 1, so that the predictions are provided as outputs. The sigmoid activation function is mathematically calculated as follows:

$$f(x) = 1/e^{-x} \tag{3.2}$$

and we use it for the output layer.

The output layer of the developed MLFNN is composed of priority rules, which were presented in detail in the previous chapter. Initially, eleven priority rules are provided for the output layer, which can be seen in Figure 3.1.

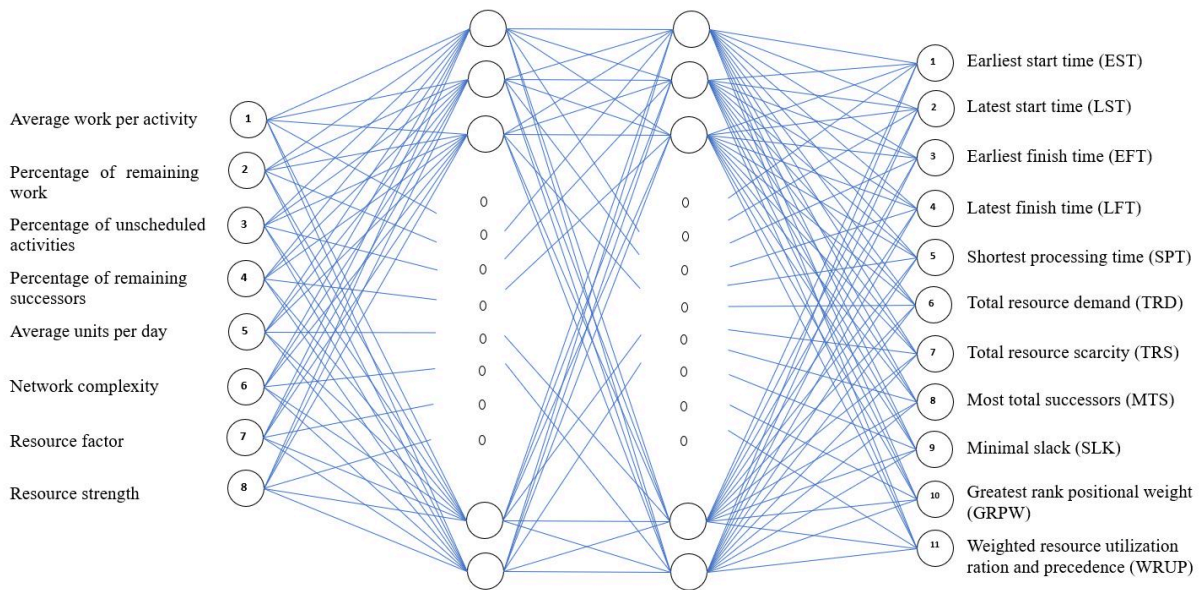


Figure 3.1 – The developed multilayer feed-forward neural network (MLFNN) consists of eight inputs, two hidden layers, and eleven outputs. The outputs or priority rules are named EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, SLK, GRPW, and WRUP.

As explained earlier, the task of the priority rules, which are the outputs of the

MLFNN, is to select an activity from the list of eligible activities at each step of project scheduling. For example, if the output of the MLFNN is the latest start time (LST), the activity with the maximum start time is selected for scheduling.

To verify the performance of the developed multilayer feed-forward neural network (MLFNN), the output layer is given a different number of neurons or priority rules. To test the performance of the proposed MLFNN, it is provided with a different number of neurons as outputs. First, it is structured with eleven priority rules as outputs, named EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, MLSK, GRPW, and WRUP, shown in Figure 3.1. Then, four of the priority rules, namely TRS, MLSK, GRPW, and WRUP, are eliminated based on their frequency in the dataset. Consequently, the next MLFNN is an MLFNN with seven outputs or priority rules named EST, LST, EFT, LFT, SPT, TRD and MTS, which is shown in Figure 3.2. Consequently, the other developed MLFNNs are structured with five, four, and three priority rules as outputs, which are shown in Figures 3.3 3.4 3.5 respectively. Also, the training performance results of these developed MLFNNs are reported in tables 3.1, 3.2, and 3.3.

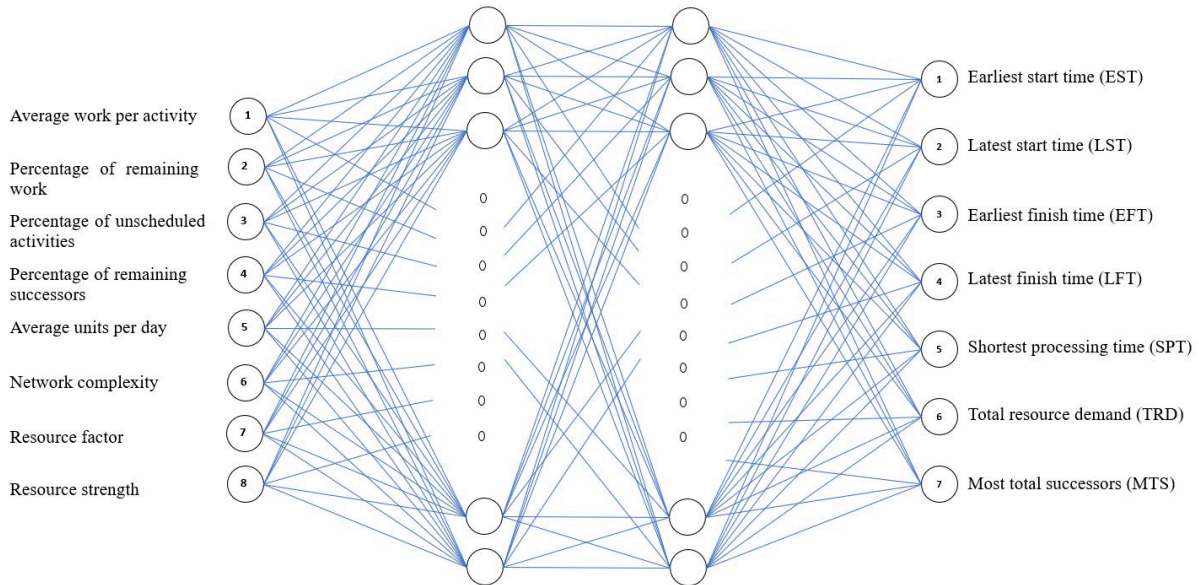


Figure 3.2 – The developed multilayer feed-forward neural network (MLFNN) consists of eight inputs, two hidden layers, and seven outputs. The outputs or priority rules are named EST, LST, EFT, LFT, SPT, TRD and MTS.

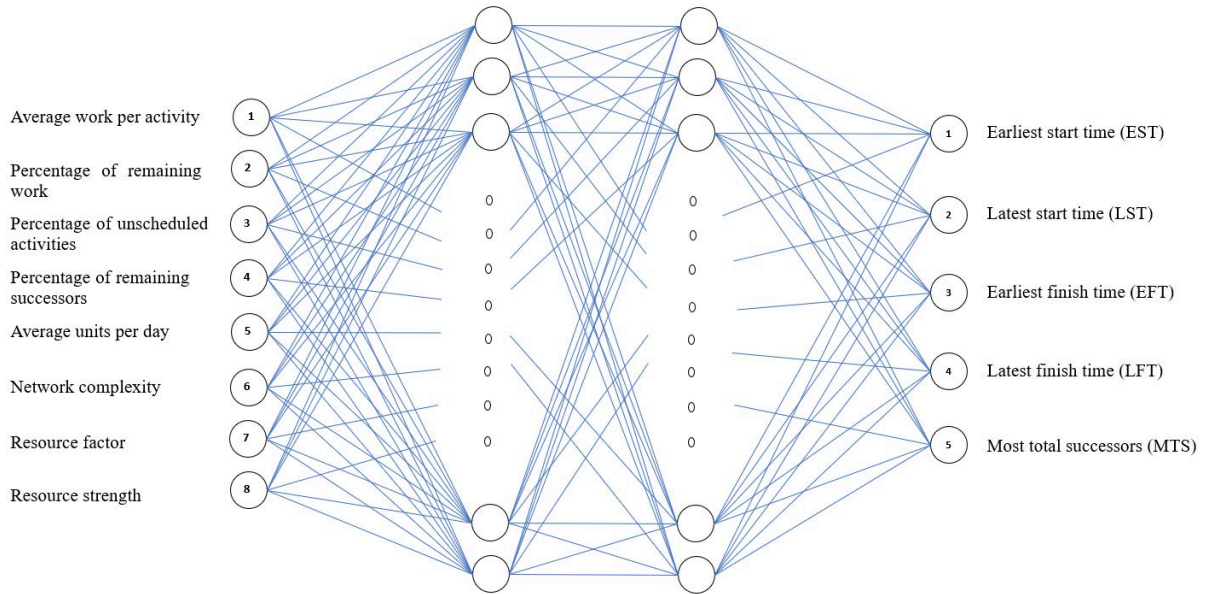


Figure 3.3 – The developed multilayer feed-forward neural network (MLFNN) is composed of eight inputs, two hidden layers, and five outputs. The outputs or priority rules are named EST, LST, EFT, LFT and MTS.

I used the PSPLIB projects to generate a dataset to feed the evolved MLFNN to train or learn the developed multilayer feed-forward neural network (MLFNN). To create the dataset, all eight project parametric characterizations are recalculated at each stage of the selected PSPLIB projects. To determine the output column of the dataset, the priority rule that provides a minimum project duration for the scheduling subproject is selected, since the ultimate goal is to minimize the project duration.

The initial dataset consists of more than 20000 data, but it is obvious that the number of priority rules in the output column is not balanced in the original dataset. Therefore, in order to achieve better performance of the developed multilayer feed-forward neural network (MLFNN), the initial dataset is balanced. This means that the number of priority rules in the output column is the same in the final dataset after the initial dataset is balanced. The objective of the training process of the multilayer feed-forward neural network (MLFNN) developed in this section is to find the optimal set of weights that preferably gives the correct output. Therefore, the inputs, the hidden layers, the outputs, and the dataset must be initialized before training the network.

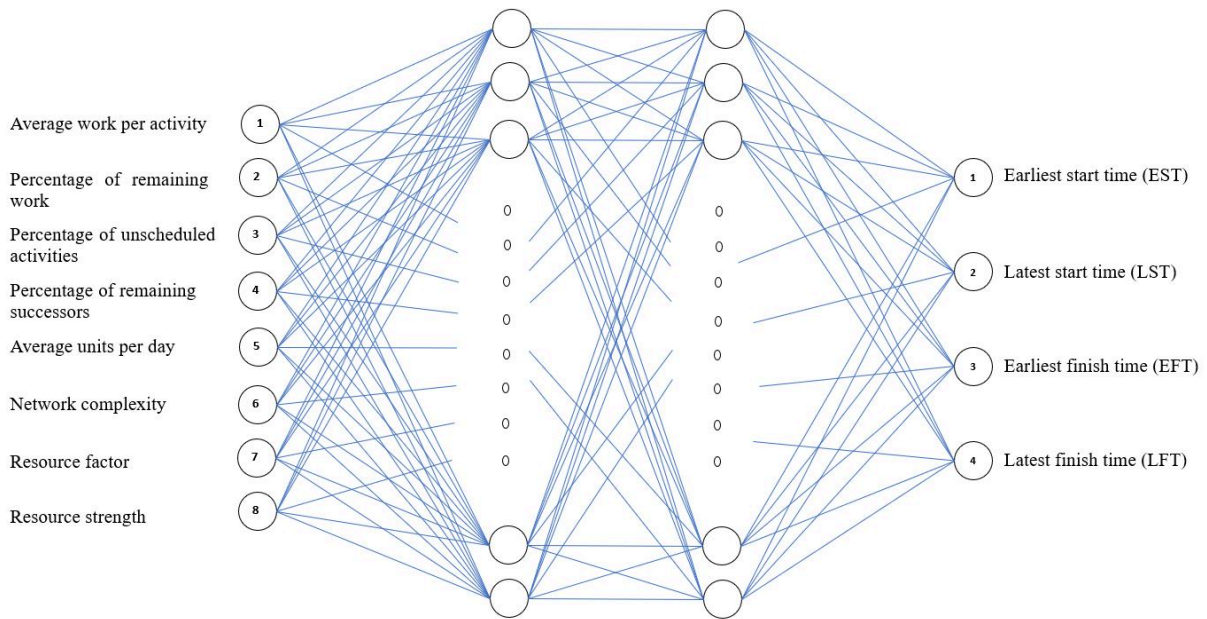


Figure 3.4 – The developed multilayer feed-forward neural network (MLFNN) consists of eight inputs, two hidden layers, and four outputs. The outputs or priority rules are so-called EST, LST, EFT and LFT.

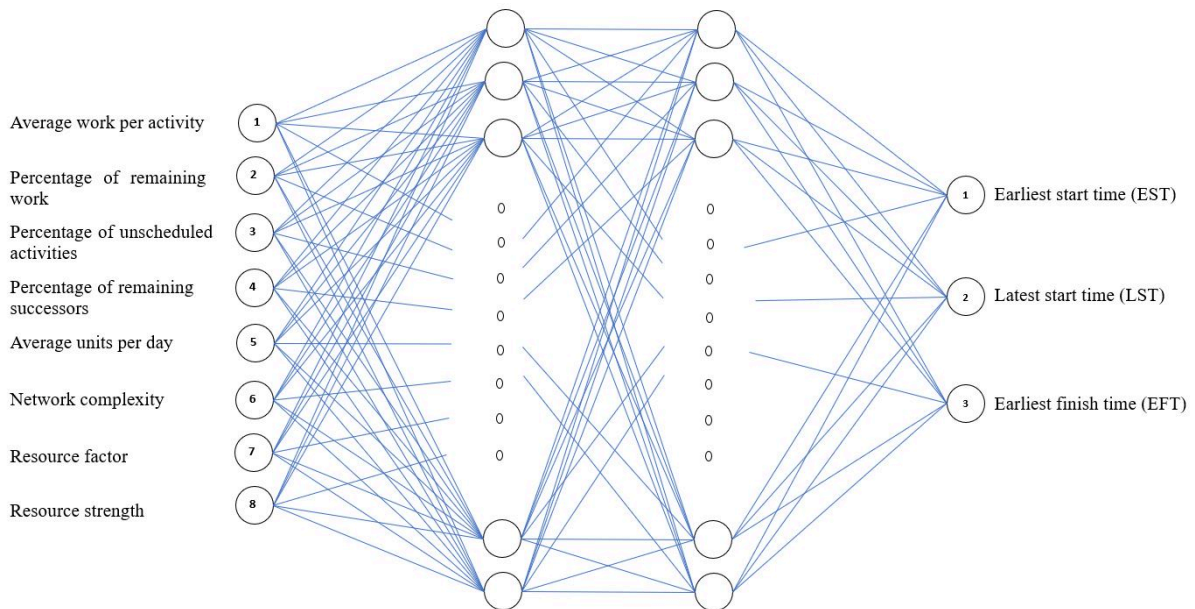
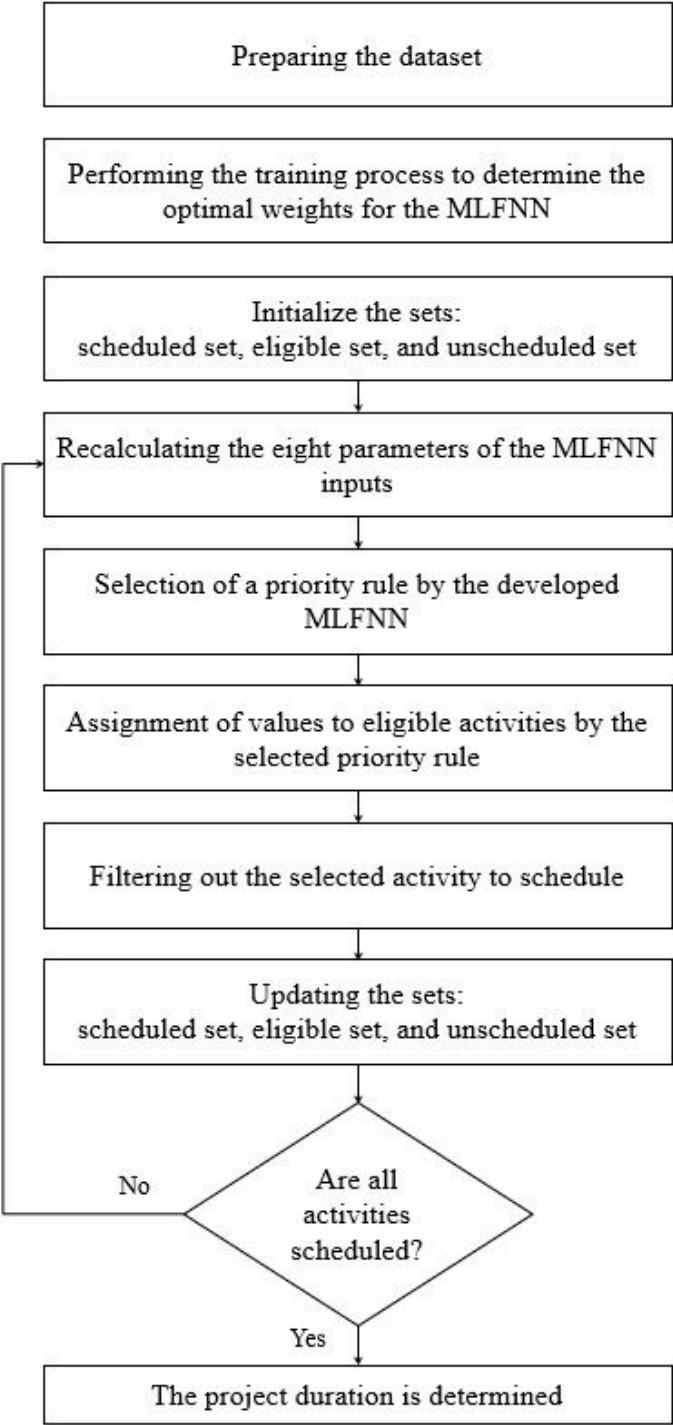


Figure 3.5 – The developed multilayer feed-forward neural network (MLFNN) consists of eight inputs, two hidden layers, and three outputs. The outputs or priority rules are the so-called EST, LST and EFT.



3.jpg

Figure 3.6 – The flowchart traces the process explained in this chapter. The process begins with training based on the prepared data set and continues with scheduling all project activities with the goal of determining the final project duration.

3.2 Computational analysis

In the computational analysis, I examine the results and discussions related to the proposed algorithm. The results include the MLFNN training results 3.2.1 and the comparative results 3.2.2. The strengths and weaknesses of the proposed algorithm and MLFNN are mentioned.

3.2.1 MLFNN training performance results and discussion

The developed multilayer feed-forward neural network (MLFNN) employs 80% of the balanced data set for training, and the remaining data, i.e., 20% of the data set, is assigned to the testing process. Moreover, the parameters of the MLFNN, called the batch size and the learning rate, are set to 64 and 0.0007, respectively. These rates lead to better results.

The multilayer feed-forward neural network (MLFNN) is executed 10 times for each mode. The training performance of the developed MLFNN is shown in Tables 3.1, 3.2, and 3.3. As explained earlier 3.1, the developed MLFNN was structured with eleven, seven, five, four, and three priority rules or neurons as the output layer. Since the network size can also affect the training performance, the developed MLFNN is analyzed with 1, 2 and 3 hidden layers and different number of epochs of 500, 1000 and 2000, respectively.

As mentioned in the section 3.1, I used the sigmoid function to determine the probability of each priority rule as an output. Therefore, we confirm the priority rules with probability greater than 0.5 in this step of scheduling the project. This means that the priority rules or outputs with probability more than 0.5 can be an allowable priority rule to filter out an activity from the eligible activities. If there is more than one output with probability greater than 0.5, the algorithm will randomly select one of them. Also, the maximum probability assigned to a priority rule is accepted to be eligible as a priority rule for selecting an activity if no probability greater than 0.5 is remarked in the output.

Based on the training performance shown in Tables 3.1, 3.2, and 3.3, it can be seen that the training performance of the developed multilayer feed-forward neural network (MLFNN) increases when the fixed number of outputs is reduced. For example, the probability of selecting an appropriate priority rule or output is higher when three neurons or priority rules are used as outputs in the evolved MLFNN than when eleven neurons are used.

Priority rules used for output layer	Hidden layers	Epoch	Min accuracy	Max accuracy	Average accuracies
Three priority rules are used as outputs: EST, LST and EFT	1	500	64%	76%	70%
	1	1000	65%	69%	67%
	1	2000	62%	71%	66%
Three priority rules are used as outputs: EST, LST, EFT and LFT	1	500	65%	71%	68%
	1	1000	62%	70%	65%
	1	2000	61%	69%	64%
Three priority rules are used as outputs: EST, LST, EFT, LFT and MTS	1	500	63%	70%	66%
	1	1000	60%	68%	64%
	1	2000	54%	62%	60%
Three priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	1	500	59%	64%	61%
	1	1000	54%	61%	58%
	1	2000	53%	58%	56%
Three priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP and TRS	1	500	48%	57%	51%
	1	1000	43%	52%	48%
	1	2000	42%	46%	44%

Table 3.1 – Training performance of the developed multilayer feedforward neural network (MLFNN) with a hidden layer. The results are obtained with different epochs 500, 1000 and 2000. Moreover, the training process is run 10 times for each mode.

Priority rules used for output layer	Hidden layers	Epoch	Min accuracy	Max accuracy	Average accuracies
Three priority rules are used as outputs: EST, LST and EFT	2	500	62%	67%	65%
	2	1000	63%	68%	66%
	2	2000	64%	70%	66%
Three priority rules are used as outputs: EST, LST, EFT and LFT	2	500	61%	67%	64%
	2	1000	61%	68%	64%
	2	2000	60%	64%	63%
Three priority rules are used as outputs: EST, LST, EFT, LFT and MTS	2	500	59%	63%	61%
	2	1000	58%	62%	61%
	2	2000	58%	61%	59%
Three priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	2	500	54%	58%	56%
	2	1000	51%	58%	54%
	2	2000	50%	57%	54%
Three priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP and TRS	2	500	41%	47%	44%
	2	1000	39%	43%	41%
	2	2000	37%	43%	40%

Table 3.2 – Training performance of the evolved multilayer feedforward neural network (MLFNN) with two hidden layers. The results are obtained with different epochs 500, 1000 and 2000. Moreover, the training process is performed 10 times for each mode.

Priority rules used for output layer	Hidden layers	Epoch	Min accuracy	Max accuracy	Average accuracies
Three priority rules are used as outputs: EST, LST and EFT	3	500	63%	68%	66%
	3	1000	63%	68%	65%
	3	2000	64%	69%	67%
Three priority rules are used as outputs: EST, LST, EFT and LFT	3	500	60%	65%	64%
	3	1000	61%	65%	63%
	3	2000	60%	65%	63%
Three priority rules are used as outputs: EST, LST, EFT, LFT and MTS	3	500	57%	54%	61%
	3	1000	59%	64%	61%
	3	2000	58%	63%	60%
Three priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	3	500	55%	61%	57%
	3	1000	51%	59%	55%
	3	2000	54%	58%	56%
Three priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP and TRS	3	500	43%	51%	46%
	3	1000	39%	45%	43%
	3	2000	38%	43%	41%

Table 3.3 – Training performance of the developed multilayer feed-forward neural network (MLFNN) with three hidden layers. The results are obtained with different epochs 500, 1000 and 2000. Moreover, the training process is run 10 times for each mode.

3.2.2 Comparative results and discussion

As explained in detail in the introduction, the objective of the research is to minimize the project duration. To this end, I used the multilayer feedforward neural network (MLFNN) as a method to select the priority rules to solve the RCPCP. It was also mentioned that the superiority of this method over evolutionary methods or metaheuristics is that it is not required to generate many solutions or populations. The MLFNN needs to be trained only once to obtain the weights only once, and these weights are then used to schedule all projects or all instances.

In the previous section 3.2.1, the developed multilayer feed-forward neural network (MLFNN) and its training performance were studied. In this section, I apply the algorithm to schedule the standard instances after training the MLFNN. For this purpose, I collected the standard problem instances, which consist of projects with four types of renewable resources and 60 or 120 activities. The standard test instances are selected from the project planning instance library, called PSPLIB.

The results obtained with the proposed algorithm are presented in terms 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.

definition:

The lower bound of the critical path refers to the sequence of all project activities, according to their precedence relationships which is the first constraint of the RCPSP.

Table 3.4 shows the results for 181 project instances of J60 and 160 project instances of J120. The results shown in Table 3.4 illustrate that the average deviations from the critical path lower bound are greater when the training performance of the developed multilayer feed-forward neural network (MLFNN) increases. Moreover, it was explained in the previous section 3.2.1 that the performance of the MLFNN increases when the fixed number of priority rules as outputs is reduced. Table 3.4 reports the average deviations from the lower bound of the critical path of 15.97% and 37.77% for J60 and J120, respectively, which are better than the other results obtained.

Table 3.5 and Table 3.6 present comparative results for project instances J60 and J120. The proposed algorithm achieves average deviations of 15.97% and 16.28% when the MLFNN uses three neurons and four neurons as outputs for J60, respectively. The obtained results are not the best among the others, but can be shown competitively in Table 3.5. Table 3.6 represents that the developed algorithm achieves better and more competitive performances. The average deviations obtained are 37.77% and 39.74% when the MLFNN uses three and four neurons as outputs for J120 instances, respectively. The comparative results in Table 3.6 present that the choice of priority rules by the developed MLFNN leads to sufficiently competitive results for larger projects.

The results obtained show that the performance of the algorithm and the solutions obtained can be improved. I suggest that this can be done by selecting appropriate priority rules as outputs and developing more neural networks. However, the main advantage of using neural networks such as the multilayer feed-forward neural network (MLFNN) proposed in this chapter is that the project can be scheduled only by generating a sequence activity corresponding to the trained MLFNN. This method prevents the generation of a large number of solutions.

Priority rules used for output layer	Number of activities	
	60	120
Three priority rules are used as outputs: EST, LST, and EFT	15.97	37.77
Four priority rules are used as outputs: EST, LST, EFT and LFT	16.28	39.74
Five priority rules are used as outputs: EST, LST, EFT, LFT and MTS	47.04	89.61
Seven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	45.59	86.39
Eleven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, GRPW, ST, WRUP, and TRS	58.96	117.42

Table 3.4 – Percentage of average deviations from critical path lower bound for the J60 and J120

Reference	Algorithm	Deviation
Our algorithm	Using of MLFNN (three neurons used as outputs)	15.97
Our algorithm	Using of MLFNN (four neurons used as outputs)	16.28
Valls, Ballestin, et al, 2008 ⁵	Hybrid GA	11.56
Alcaraz and Concepción, 2001 ⁶	Genetic Algorithm	NA
Hartmann, 2002 ⁷	Genetic Algorithm	12.21
Nonobe and Ibaraki, 2002 ⁸	Tabu Search	12.97
Bouleimen and Lecocq, 2003 ⁹	Simulated Annealing	12.75
Koulinas, Kotsikas, et al, 2014 ¹⁰	Particle Swarm Optimization-HH	11.74
Mendes, Gonçalves, et al, 2009 ¹¹	Genetic Algorithm	11.72
Chen, Shi, et al, 2010 ¹²	Hybrid (ACO and SS)	11.75
Agarwal, Colak, et al, 2011 ¹³	Neurogenetic	11.51
Mobini, Mobini, et al, 2011 ¹⁴	Artificial Immune Algorithm	11.17
Gonçalves, Resende, et al, 2011 ¹⁵	Genetic Algorithm	11.56
Wang and Fang, 2012 ¹⁶	Hybrid Estimation of Distribution Algorithm	11.44
Chen R.-M. ¹⁷ , 2011	Particle swarm optimization	12.03
Ziarati and Akbari, 2011 ¹⁸	Bee Algorithms	12.55
Proon and Jin, 2011 ¹⁹	Genetic Algorithm with Neighborhood Search	11.35
Liu, Liu, et al, 2020 ²⁰	Genetic Algorithm	11.74
Zamani, 2017 ²¹	Genetic Algorithm	11.61
Lim, Ma, et al, 2013 ²²	Hybrid Genetic Algorithm	11.73

Table 3.5 – Percentage of average deviations from critical path lower bound for the J60

Reference	Algorithm	Deviation
Our algorithm	Using of MLFNN (three neurons used as outputs)	37.77
Our algorithm	Using of MLFNN (four neurons used as outputs)	39.74
Valls, Ballestin, et al, 2008 ²³	Hybrid GA	34.07
Alcaraz and Concepción, 2001 ²⁴	Genetic Algorithm	39.36
Hartmann, 2002 ²⁵	Genetic Algorithm	37.19
Nonobe and Ibaraki, 2002 ²⁶	Tabu Search	40.86
Bouleimen and Lecocq, 2003 ²⁷	Simulated Annealing	40.82
Koulinas, Kotsikas, et al, 2014 ²⁸	Particle Swarm Optimization-HH	35.20
Mendes, Gonçalves, et al, 2009 ²⁹	Genetic Algorithm	35.87
Chen, Shi, et al, 2010 ³⁰	Hybrid (ACO and SS)	35.19
Agarwal, Colak, et al, 2011 ³¹	Neurogenetic	34.65
Mobini, Mobini, et al, 2011 ³²	Artificial Immune Algorithm	30.04
Gonçalves, Resende, et al, 2011 ³³	Genetic Algorithm	35.94
Wang and Fang, 2012 ³⁴	Hybrid Estimation of Distribution Algorithm	34.83
Chen R.-M. ³⁵ , 2011	Particle swarm optimization	35.71
Ziarati and Akbari, 2011 ³⁶	Bee Algorithms	37.72
Proon and Jin, 2011 ³⁷	Genetic Algorithm with Neighborhood Search	33.45
Liu, Liu, et al, 2020 ³⁸	Genetic Algorithm	34.88
Zamani, 2017 ³⁹	Genetic Algorithm	34.59
Lim, Ma, et al, 2013 ⁴⁰	Hybrid Genetic Algorithm	34.95

Table 3.6 – Percentage of average deviations from critical path lower bound for the J120

3.3 Conclusion

In this chapter, a multilayer feed-forward neural network (MLFNN) was presented to solve the resource constrained project scheduling problem (RCPSP). As mentioned earlier, the advantage of the proposed algorithm over evolutionary methods or meta-heuristics is that there is no need to generate populations or numerous solutions, on the contrary, the proposed algorithm generates a solution according to the training of the MLFNN. For this reason, the developed MLFNN benefits from the eight parameters as inputs and a different number of priority rules as outputs. It is understandable that the project specifications may change during the project schedule. Therefore, the performance of the priority rules to select an eligible activity depends on the project specifications, such as the existing project constraints, the number of resources, and Therefore, different priority rules are suitable for different types of subprojects or projects. Therefore, in this chapter, we used the MLFNN with a mixture of priority rules as outputs to select an eligible activity appropriated to characterized sub-project.

I used the PSPLIB projects to create the dataset for the MLFNN training and testing procedure. The developed MLFNN can select a priority rule as output after training the network. Consequently, at each step, a suitable eligible activity can be filtered from the list of eligible activities by the selected priority rule to be added to the project schedule. In the following, the training performance of the developed MLFNN was analyzed using the balanced dataset after setting the parameters of the network. It is obvious that the performance of the proposed algorithm increases when the number of outputs is reduced. Then, the proposed algorithm was applied to schedule the instances of PSPLIB after training the MLFNN. The computational results show that the results are sufficiently competitive.

Consequently, we found that the developed multilayer feed-forward neural network (MLFNN) has reasonable performance to deal with the RCPSP. Although the results are not yet the best among others, the proposed algorithm should encourage researchers to use the exciting potential of improvements to develop neural networks to deal with resource-constrained project scheduling problem.

A CONVOLUTIONAL NEURAL NETWORK (CNN) TO SOLVE THE RCPSP

This chapter focuses on answering the resource-constrained project scheduling problem (RCPSP) using a convolutional neural network (CNN). As explained in the previous chapters, the objective of the problem is to optimize and minimize the project duration or the project makespan constraining the precedence constraints and the resource quantities during project scheduling. This means that the objective of the problem is to minimize the project duration while respecting the project constraints. In this chapter, a convolutional neural network (CNN) is proposed to handle the standard RCPSP with a single mode.

The proposed algorithm schedules the project activities one by one to create a sequence of activities considering the output of the developed convolutional neural network (CNN). In other words: In this chapter, the serial schedule generation scheme (SSGS) is employed to schedule the project activities using an evolved convolutional neural network (CNN) as a mechanism to select an appropriate priority rule to filter out a candidate activity.

The developed CNN learns according to the eight project parameters explained in Chapter 2. These parameters are called network complexity, resource factor, resource strength, average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of remaining successors, and average units per day. The above parameters are the inputs of the developed CNN and are recalculated at each step of the project schedule. The outputs of the network are the priority rules, which are explained in Chapter 2. Therefore, after the learning process, the convolutional neural network can automatically select an appropriate priority rule to filter an activity from the eligible activities. In this way, the algorithm can schedule all project activities according to the given project constraints. In the second section of this chapter, the performance of the convolutional neural network (CNN) approach is investigated using standard benchmark problems from PSPLIB.

4.1 Algorithm and the developed CNN

The proposed algorithm uses a schedule generation scheme as a function to generate a feasible schedule. For generating feasible schedules, there are two special schedule generation schemes (SGS), serial schedule generation scheme (SSGS) and parallel schedule generation scheme (PSGS). For further explanation and a better understanding of the differences between these two methods, the reader is referred to the article by Kolisch and Hartmann¹. In this chapter, as with the algorithm used in Chapter 3, I benefit from the serial schedule generation scheme (SSGS), which constructs active schedules that contain at least one feasible or optimal solution. In other words, the SSGS can make a sequence of activities that leads to a feasible or optimal schedule because the SSGS always constructs feasible schedules^{2 3}.

As explained in Chapter 3, the serial schedule generation scheme is composed of n steps during the project schedule, where n indicates the steps or number of project activities. For example, in a project with 62 activities, there are 62 scheduling phases or steps. Therefore, these eight inputs or parameters are recalculated n times to recharacterize the new subprojects. To recall the SSGS structure, 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.

The project schedule process is initialized with a partial schedule containing only the dummy start activity at time zero. Then, the inputs, which are the eight parameters mentioned above, are recomputed, 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 trained convolutional neural network (CNN). The developed convolutional neural network is trained using the created dataset to determine the optimal weights before the project scheduling process begins. The weights are used for the mathematical calculations of the network to select an appropriate priority rule as output at each step of the scheduling process. The selected priority rule assigns values to the eligible activities according to the selection criteria, resulting in the selection of an eligible activity from the eligible set E_n . Then, the selected activity is added to the project

1. Kolisch and Hartmann, *op. cit.*
2. Koulinas, Kotsikas, and Anagnostopoulos, *op. cit.*
3. Golab et al., *op. cit.*

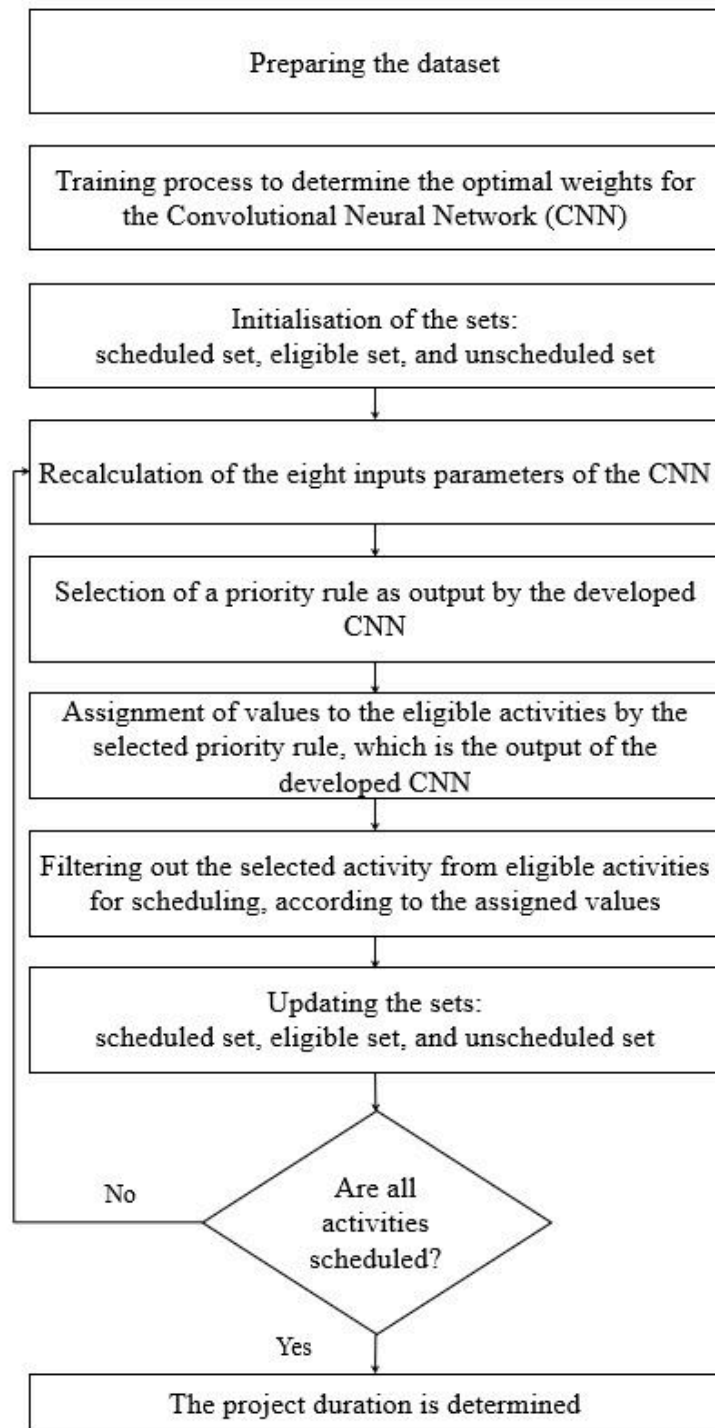
schedule at the earliest possible time that is feasible with respect to the two constraints. The algorithm proceeds to schedule all the activities in the project. In this way, at the end of the algorithm, the project duration is determined, which is the final goal of the algorithm. For a better understanding of the proposed algorithm, we refer to Figure 4.1, which explains the process of the developed algorithm.

I propose a convolutional neural network (CNN) to select an appropriate priority rule as an output. The selected priority rule can filter an activity from the eligible activities.

It was mentioned that convolutional neural networks (CNNs) are constructed in different types of layers. These layers consist of a series of interconnected neurons that benefit from activation functions. In general, a neural network is fed with data through the input layer. As shown in Figures 4.2, 4.3, 4.4, 4.5, and 4.6, the evolved convolutional neural network (CNN) is fed with eight different data. The input layer includes eight different parameters, which were explained in more detail in the section of project parametric characterizations. 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 eight parameters or inputs of the CNN are recalculated at each stage of project scheduling to recharacterize the new subproject or step; in this way, the conditions for selecting an eligible activity from the eligible list for the project schedule are prepared. For example, for a project in the introduction, there are 32 scheduling phases or steps, so these parameters or the inputs are recalculated 32 times to characterize the subprojects.

In the proposed CNN, a different number of convolutional layers is used. As can be seen from the result tables 4.1, 4.2 and 4.3, the developed CNNs are designed with 1, 2 and 3 convolutional layers. As mentioned earlier, the convolutional neural network (CNN) is a back-propagation algorithm that includes two phases. The first phase is the forward phase and the second phase is called the backward phase. The two phases try to improve the weights and bias values.

The output layer of the proposed convolutional neural network consists of priority rules, which are discussed earlier. First, eleven priority rules are established for the output layer. Then, to check the performance of the developed CNN, a different number of neurons or priority rules are assigned as outputs to the output layer. The task of the priority rules is to select an eligible activity from the eligible list at each step of the project scheduling by assigning a value to the eligible activities. For example, if the output of the network is the earliest finish time (LFT), the activity with the minimum finish time is selected for



4.jpg

Figure 4.1 – The flowchart shows the algorithm explained in this section. The process begins with training based on the dataset created and continues with scheduling all project activities to determine the final project duration.

scheduling.

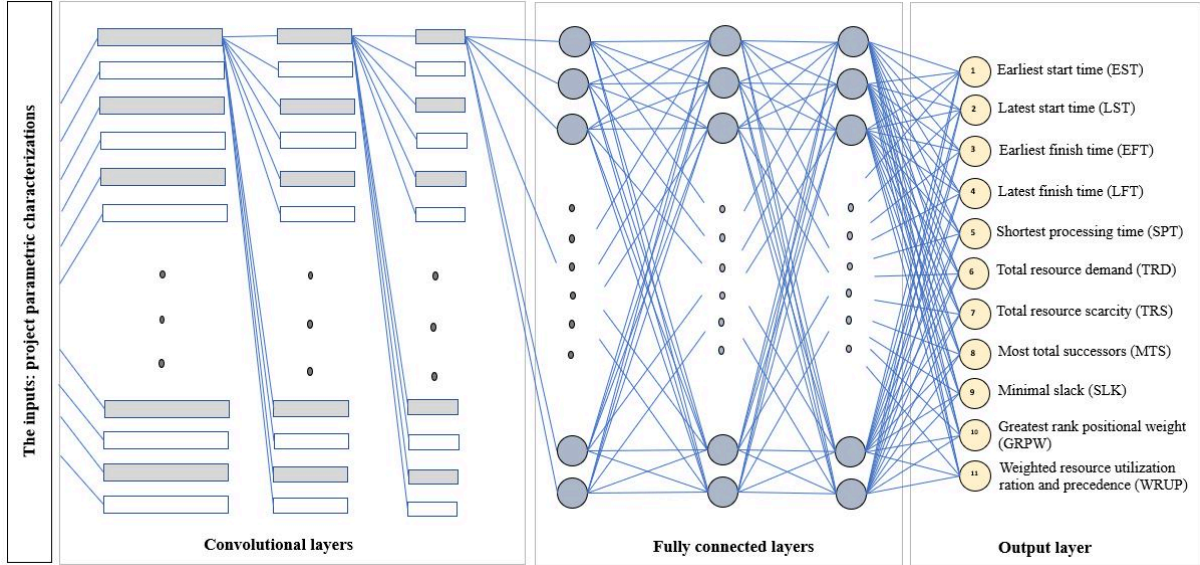


Figure 4.2 – The developed convolutional neural network (CNN) consists of eight inputs as input layer, three convolutional layers and three fully connected layers, and eleven outputs. The outputs or priority rules are named EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, SLK, GRPW, and WRUP.

If no activation function is employed in a convolutional neural network, the output would be just a simple linear function. Therefore, the activation function is applied to dynamically build the network and provide it with the ability to extract complex data. For this purpose, the relu activation function is used in the developed convolutional neural network. The relu is a nonlinear activation. This function can be mathematically calculated as

$$f(x) = \max(0, x) \quad (4.1)$$

and this function is widely used in neural networks. This activation function is used for the convolutional layers and the fully connected layers of the evolved CNN.

To verify the performance of the developed convolutional neural network (CNN), the output layer is structured with different sets of neurons or priority rules. Initially, it is structured with eleven priority rules or neurons named as outputs EST, LST, EFT, LFT, SPT, TRD, TRS, MTS, MSLK, GRPW and WRUP as shown in Figure 4.2. The next is a CNN with seven outputs or priority rules named EST, LST, EFT, LFT, SPT, TRD and MTS, which is shown in Figure 4.3. The other evolved CNNs were developed with

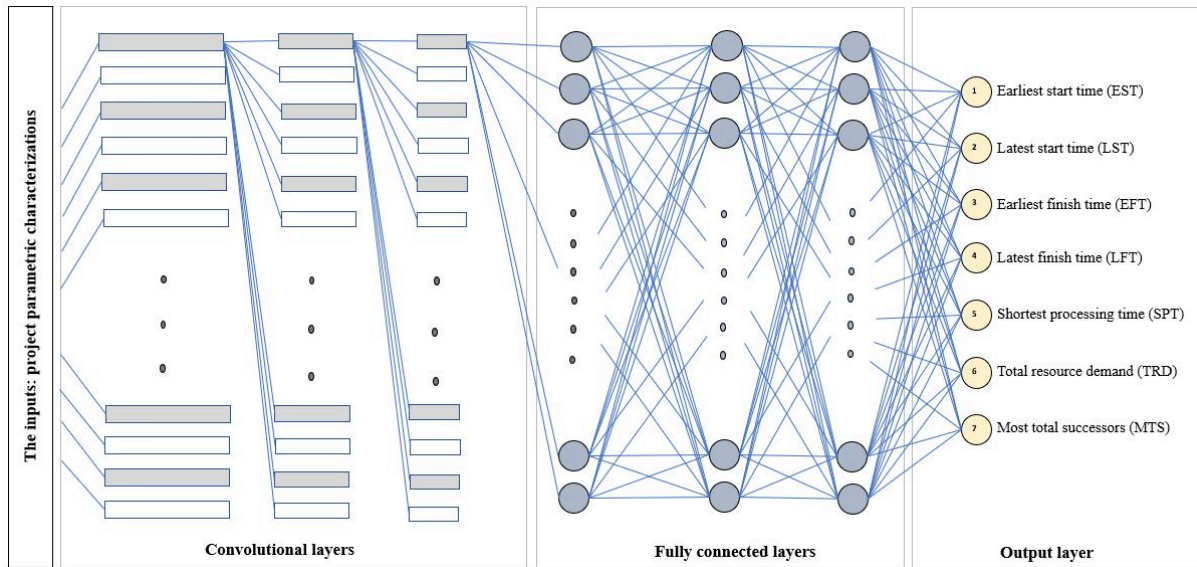


Figure 4.3 – The developed convolutional neural network (CNN) consists of eight inputs as input layer, three convolutional layers and three fully connected layers, and seven outputs. The outputs or priority rules are named EST, LST, EFT, LFT, SPT, TRD and MTS.

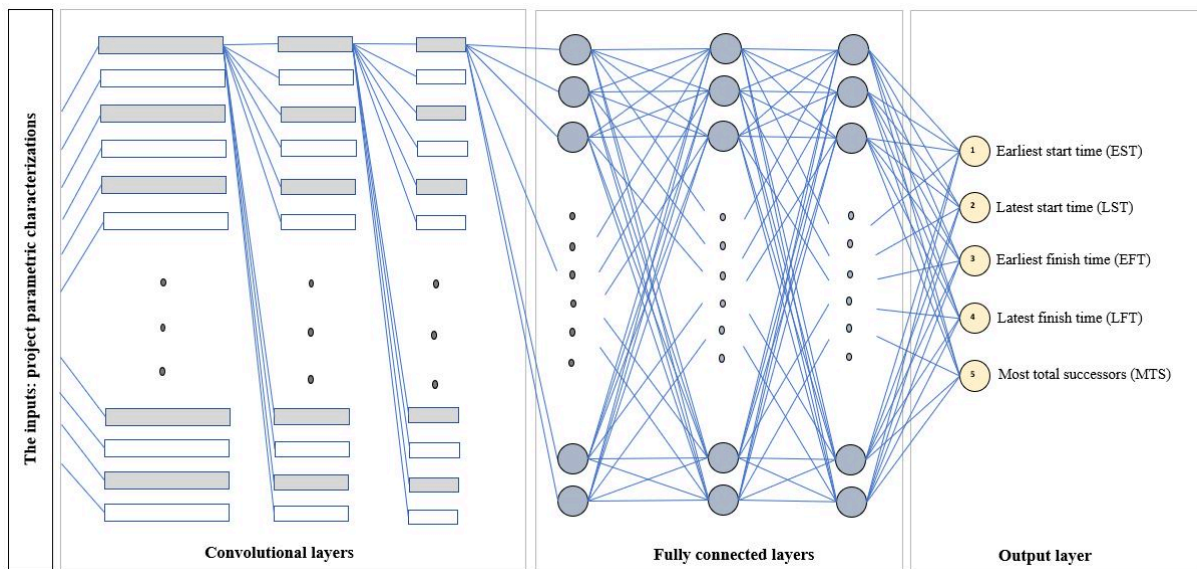


Figure 4.4 – The developed convolutional neural network (CNN) is composed of eight inputs as input layer, three convolutional layers and three fully connected layers, and five outputs. The outputs or priority rules are named EST, LST, EFT, LFT and MTS.

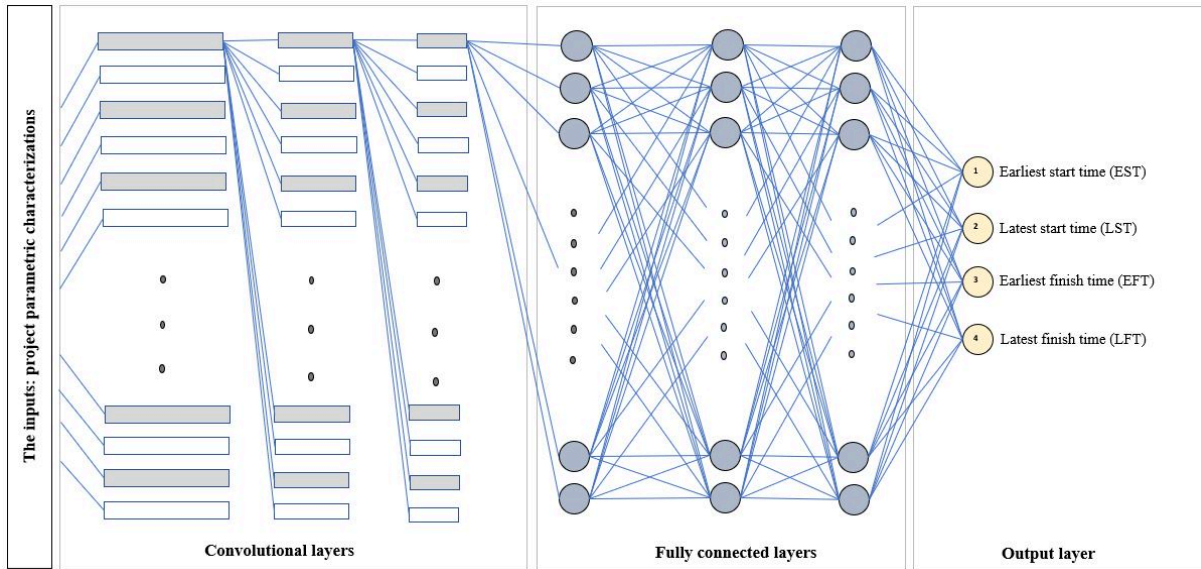


Figure 4.5 – The convolutional neural network (CNN) developed consists of eight inputs as the input layer, three convolutional layers and three fully connected layers, and four outputs. The outputs or priority rules are named EST, LST, EFT and LFT.

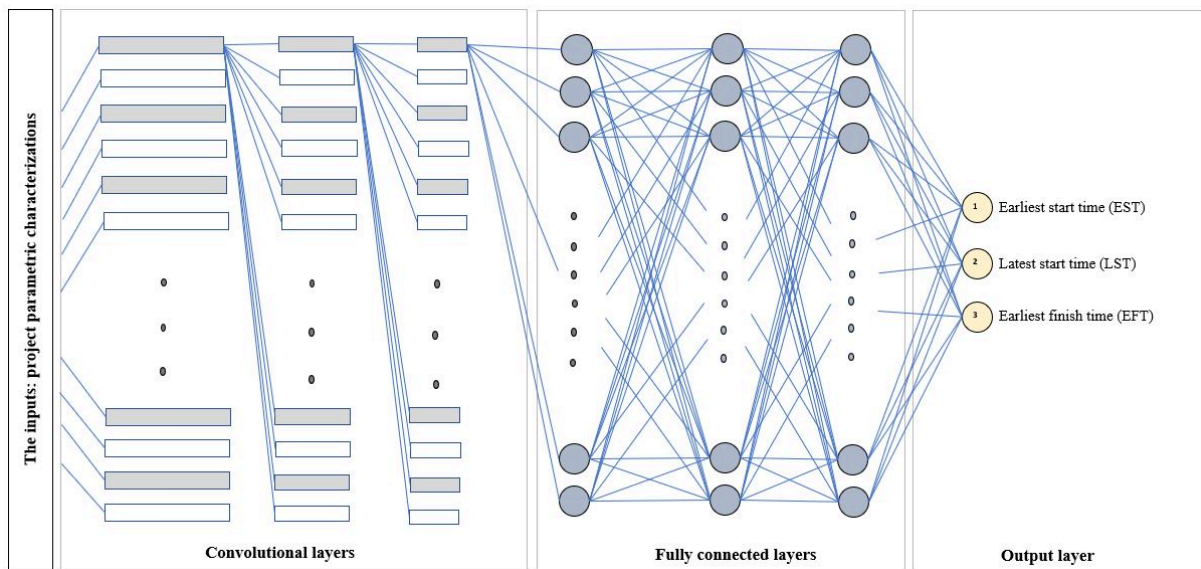


Figure 4.6 – The developed convolutional neural network (CNN) is composed of eight inputs as input layer, three convolutional layers and three fully connected layers, and three outputs. The outputs or priority rules are named EST, LST and EFT.

five, four, and three priority rules as outputs, which are shown in Figures 4.4, 4.5, and 4.6, respectively. The training performance results of the developed CNNs are reported in Tables 4.1, 4.2, and 4.3.

PSPLIB projects are used to create a dataset to feed the developed CNN. The developed convolutional neural network (CNN) uses the dataset for training or learning. To create the dataset, all eight project parameters are recalculated at each stage of the selected PSPLIB projects. In addition, to control the output column of the dataset, the priority rule is selected to provide a minimum project duration for the scheduling subproject, since the ultimate goal is to minimize the project duration.

The initial dataset has more than 20000 data, but to get better performance we should balance the initial data set because the number of priority rules in the output column is not balanced. This means that the number of priority rules in the output column will be the same in the final dataset after the output dataset is balanced. As mentioned earlier, the objective of the training process of the convolutional neural network (CNN) developed in this chapter is to find the optimal set of weights that preferably gives the correct output. Therefore, the inputs, the convolutional layers and the fully connected layers, the outputs, and the dataset must be initialized before training the network.

4.2 Computational analysis

This section presents the results and discussions related to the proposed algorithm. The results include the CNN training performance results 4.2.1 and the comparative results 4.2.2. The strengths and weaknesses of the developed algorithm and the proposed convolutional network are mentioned below.

4.2.1 The convolutional neural network training performance results and discussion

As explained in section 4.1, the developed convolutional neural network (CNN) is trained with the created dataset using the projects of PSPLIB. To create the dataset, all eight project parametric characterizations are recomputed at each step of the selected PSPLIB projects. The dataset is used as input for training the developed CNN. To regulate the output column of the dataset for each step or subproject, the priority rule that provides minimum project duration in that step of scheduling is selected. Also, the initial dataset

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%
Three 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%
Three 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%
Three 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%
Three 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 4.1 – Training performance of the evolved convolutional neural network (CNN) with a convolutional layer. The results are obtained with different epochs 500, 1000 and 2000. Moreover, the training process is performed 10 times for each mode.

should be equalized to achieve better performance of the developed convolutional neural network (CNN).

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

The results of the training performance are presented in Table 4.1, Table 4.2 and Table 4.3. 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. Also, the inputs, outputs, test sets, and required parameters must be initialized before training.

To obtain the training performance and optimal weights, 80% of the data from the balanced dataset were used for training and 20% of the data, i.e., the remaining data, was assigned to the testing process of the developed CNN. 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 as the final output

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	57%	71%	69%
	2	2000	68%	74%	70%
Three 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%
Three 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%
Three priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	2	500	55%	60%	56%
	2	1000	59%	67%	63%
	2	2000	62%	74%	65%
Three 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 4.2 – Training performance of developed convolutional neural network (CNN) with two convolutional layers. The results are obtained with different epochs 500, 1000 and 2000. Moreover, the training process is run 10 times for each mode.

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%
Three 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%
Three 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%
Three 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%
Three 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%

Table 4.3 – Training performance of the developed convolutional neural network (CNN) with three convolutional layers. The results are obtained with different epochs 500, 1000 and 2000. Also, the training process is performed 10 times for each mode.

of the developed CNN at each stage of scheduling. According to the obtained training performance results, it is clear that the performance of the developed convolutional neural network (CNN) increases when the fixed number of outputs is reduced. According to the training results, the probability of selecting the correct priority rule is higher when three neurons are used as outputs than when eleven neurons were used.

4.2.2 Comparative results and discussion

In this chapter, we apply a convolutional neural network (CNN) embedded in an algorithm to solve the resource-constrained project scheduling problem (RCPCP). The superiority of the proposed algorithm against evolutionary methods or metaheuristics is that there is no requirement to generate many solutions or populations. The developed convolutional neural network (CNN) trains the weights only once at the beginning and the obtained weights are then used to select an appropriate priority rule at each step of the project schedule, which leads to the scheduling of all project instances.

The training performance was presented in section 4.2.1. After training the developed convolutional neural network, the algorithm is used to schedule the standard instances. The standard problem instances include projects with four types of renewable resources and 60 and 120 activities. The instances are selected from the PSPLIB.

The competitive results are formed 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. As mentioned in section 3.2.2, the critical path lower bound is simply the sequencing of all activities of a project according to their precedence relations without considering resource constraints. That is, if a project is scheduled based only on precedence relationships, the project duration is determined based on the critical path lower bound.

Table 4.4 summarizes the results obtained for 181 project instances of J60 and 160 project instances of J120. From Table 4.4, it can be seen that the average deviations from the critical path lower bound are more acceptable as the training performance of the developed convolutional neural network (CNN) increases. In the previous section 4.2.1, it was explained that the performance of the CNN increases when the fixed number of priority rules as outputs is reduced. Table 4.4 reports the average deviations from the lower bound of the critical path of 15.97% and 37.77% for J60 and J120, respectively, which are better than the other obtained results.

Table 4.5 contains the comparative results for J60 standard project instances. These results declare that the algorithm with the evolved CNN achieves average deviations of

Priority rules used for output layer	Number of activities	
	60	120
Three priority rules are used as outputs: EST, LST, and EFT	16.57	38.39
Four priority rules are used as outputs: EST, LST, EFT and LFT	15.97	37.77
Five priority rules are used as outputs: EST, LST, EFT, LFT and MTS	16.19	39.48
Seven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD and PT	20.48	53.06
Eleven priority rules are used as outputs: EST, LST, EFT, LFT, MTS, TRD, PT, ST, TRS, GRPW, and WRUP	63.33	124.72

Table 4.4 – Percentage of average deviations from critical path lower bound for the J60 and J120

16.57%, 15.97%, and 16.19% when the evolved CNN uses three, four, and five neurons as outputs for J60, respectively. The presented results are not the best, but they can be quite competitive. Table 4.6 summarizes the comparative results for J120 standard project instances. The presented results confirm that our algorithm achieves better performance when processing larger projects. The average deviations obtained are 38.39%, 37.77% and 39.48% when the proposed CNN employs three, four and five neurons as outputs for J120 instances, respectively. The comparative results in Table 4.6 confirm that the selection of priority rules by the developed convolutional neural network leads to sufficiently competitive results.

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

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

Reference	Algorithm	Deviation
Our algorithm	Using of CNN (three neurons used as outputs)	16.57
Our algorithm	Using of CNN (four neurons used as outputs)	15.97
Our algorithm	Using of CNN (five neurons used as outputs)	16.19
Golab et al, 2022 ⁴	MLFNN (three neurons used as outputs)	15.97
Golab et al, 2022 ⁵	MLFNN (four neurons used as outputs)	16.28
Valls, Ballestin, et al, 2008 ⁶	Hybrid GA	11.56
Alcaraz and Concepción, 2001 ⁷	Genetic Algorithm	NA
Hartmann, 2002 ⁸	Genetic Algorithm	12.21
Nonobe and Ibaraki, 2002 ⁹	Tabu Search	12.97
Bouleimen and Lecocq, 2003 ¹⁰	Simulated Annealing	12.75
Koulinas, Kotsikas, et al, 2014 ¹¹	Particle Swarm Optimization-HH	11.74
Mendes, Gonçalves, et al, 2009 ¹²	Genetic Algorithm	11.72
Chen, Shi, et al, 2010 ¹³	Hybrid (ACO and SS)	11.75
Agarwal, Colak, et al, 2011 ¹⁴	Neurogenetic	11.51
Mobini, Mobini, et al, 2011 ¹⁵	Artificial Immune Algorithm	11.17
Gonçalves, Resende, et al, 2011 ¹⁶	Genetic Algorithm	11.56
Wang and Fang, 2012 ¹⁷	Hybrid Estimation of Distribution Algorithm	11.44
Chen R.-M. ¹⁸ , 2011	Particle swarm optimization	12.03
Ziarati and Akbari, 2011 ¹⁹	Bee Algorithms	12.55
Proon and Jin, 2011 ²⁰	Genetic Algorithm with Neighborhood Search	11.35
Liu, Liu, et al, 2020 ²¹	Genetic Algorithm	11.74
Zamani, 2017 ²²	Genetic Algorithm	11.61
Lim, Ma, et al, 2013 ²³	Hybrid Genetic Algorithm	11.73

Table 4.5 – Percentage of average deviations from critical path lower bound for the J60

Reference	Algorithm	Deviation
Our algorithm	Using of CNN (three neurons used as outputs)	38.39
Our algorithm	Using of CNN (four neurons used as outputs)	37.77
Our algorithm	Using of CNN (five neurons used as outputs)	39.48
Golab et al, 2022 ²⁴	MLFNN (three neurons used as outputs)	37.77
Golab et al, 2022 ²⁵	MLFNN (four neurons used as outputs)	39.79
Valls, Ballestin, et al, 2008 ²⁶	Hybrid GA	34.07
Alcaraz and Concepción, 2001 ²⁷	Genetic Algorithm	39.36
Hartmann, 2002 ²⁸	Genetic Algorithm	37.19
Nonobe and Ibaraki, 2002 ²⁹	Tabu Search	40.86
Bouleimen and Lecocq, 2003 ³⁰	Simulated Annealing	40.82
Koulinas, Kotsikas, et al, 2014 ³¹	Particle Swarm Optimization-HH	35.20
Mendes, Gonçalves, et al, 2009 ³²	Genetic Algorithm	35.87
Chen, Shi, et al, 2010 ³³	Hybrid (ACO and SS)	35.19
Agarwal, Colak, et al, 2011 ³⁴	Neurogenetic	34.65
Mobini, Mobini, et al, 2011 ³⁵	Artificial Immune Algorithm	30.04
Gonçalves, Resende, et al, 2011 ³⁶	Genetic Algorithm	35.94
Wang and Fang, 2012 ³⁷	Hybrid Estimation of Distribution Algorithm	34.83
Chen R.-M. ³⁸ , 2011	Particle swarm optimization	35.71
Ziarati and Akbari, 2011 ³⁹	Bee Algorithms	37.72
Proon and Jin, 2011 ⁴⁰	Genetic Algorithm with Neighborhood Search	33.45
Liu, Liu, et al, 2020 ⁴¹	Genetic Algorithm	34.88
Zamani, 2017 ⁴²	Genetic Algorithm	34.59
Lim, Ma, et al, 2013 ⁴³	Hybrid Genetic Algorithm	34.95

Table 4.6 – Percentage of average deviations from critical path lower bound for the J120

the training dataset is specialized.

4.3 Conclusion

In this chapter, an algorithm equipped with a convolutional neural network was employed to solve the resource-constrained project scheduling problem (RCPSP). As mentioned earlier, the superiority of the proposed algorithm over other methods such as evolutionary methods or meta-heuristics is that it does not require generating numerous solutions or populations. On the contrary, the proposed algorithm generates only one solution based on the trained CNN. The proposed convolutional neural network (CNN) uses eight inputs, called project parametric characterizations and a different number of outputs, called priority rules. 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 existing project constraints. Therefore, different priority rules or activity selection methods are satisfactory for different types of subprojects or projects. In this chapter, we proposed a convolutional neural network with different number of priority rules as outputs to select a suitable eligible activity for the characterized subproject.

We employed the same dataset used in Chapter 3. 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 analyzed using the balanced dataset after the parameters of the developed CNN were set. It was found that the performance of the CNN increased when the number of outputs was reduced (this performance is similar to that of the evolved MLFNN). The convolutional neural network selects a priority rule as output after training the network. In this way, based on the selected priority rule, an acceptable activity can be selected from the list of eligible activities to be added to the project schedule. In the following chapter, we tested the algorithm for scheduling the standard PSPLIB projects after training the CNN. The obtained results confirm that the results are competitive. Although the presented results are not yet the best, we believe that the proposed algorithm, which uses a convolutional neural network, has a reasonable performance to deal with the RCPSP.

CONCLUSION

This chapter draws the conclusion of the research conducted in the main body of the dissertation. To do so, I summarize the researches discussed in Chapters 1 and 2 and the related research. Then, I try to present two proposed algorithms and two developed neural networks. Also, the results obtained with the two algorithms are compared. Finally, the limitations are pointed out and recommendations for future works are given.

The problem and related works

This research deals with the well-known project schedule known problem, called the resource-constrained project scheduling problem (RCPSP). This problem is defined in the context of the project schedule, which is placed in the platform of project management processes. As explained earlier, the purpose of the RCPSP is to determine an order of activities in order to minimize and optimize the project duration according to the given project constraints, which are precedence relationships between activities and resource constraints.

I have benefited from an example in which a project is planned according to a method called the serial schedule generation scheme. In the example given, three priority rules were used to assign a value to each eligible activity. Then, an activity is selected based on the determined value and the selection criteria. It was mentioned that the difficulty of the problem becomes visible when there is more than one eligible activity, since in the serial schedule generation scheme it is only possible to filter out one activity in each step. Therefore, when there are different priority rules and more than one eligible activity in the set of eligible activities, it is not clear which activity is more suitable for scheduling. Moreover, the priority rules lead to different results depending on the project specifications and constraints and conditions. Therefore, it is necessary to choose an appropriate priority rule to select an activity from the set of eligible activities. The use of the different priority rules may affect the project duration, which in turn may affect the objective of the problem.

The objective of the problem

The objective of the resource constrained project scheduling problem (RCPSP) is to determine a sequence of activities to minimize and optimize project duration considering two constraints.

The ineffectiveness of the exact method to solve large RCPSP led researchers to use heuristics and metaheuristics for large problems. They have developed various conventional metaheuristic algorithms such as genetic algorithms, particle swarm optimization, ant colony optimization, bee colony optimization, simulated annealing, and evolutionary algorithms to solve the RCPSP. As discussed in Chapter 1, these methods typically start with a solution or set of solutions as a population and then attempt to evolve the solutions during the process. However, from our point of view, generating an optimal or near-optimal sequence of activities is the key to finding the right method. Therefore, in this work, I have focused on methods based on neural networks.

Employing the two neural networks to project schedule

Readers are given a brief description of neural networks. As mentioned earlier, artificial neural networks (ANNs) consist of computational units called neurons. These computational units process data while being connected to each other by adjustable weight connections. Then, a brief classification of neural networks was presented. I presented two types of neural networks used in this work: multilayer feed-forward neural network (MLFNN) and convolutional neural network 1D (CNN).

In the following, the eight project parametric characterizations of the developed neural networks were presented. These parameters are the so-called network complexity (NC), resource factor (RF), resource strength (RS), average work per activity, percentage of remaining work, percentage of unscheduled activities, percentage of remaining successors, and average units per day. It was also emphasized that these parameters can be recalculated at each stage of project scheduling to characterize the existing subproject. In addition, the eleven rules, called priority rules, that were employed as outputs of the developed neural networks were described. These priority rules assign values to the activities

that are used to select an activity according to the selection criteria, which can be the minimum or maximum value.

In Chapters 3 and 4, I benefit from two algorithms to solve the resource-constrained project scheduling problem (RCPSP), using an evolved multilayer feed-forward neural network (MLFNN) and a developed convolutional neural network (CNN). I used the standard projects to generate the dataset for training the two aforementioned networks.

The training performance of the developed MLFNN was analyzed with 1, 2 and 3 hidden layers and the different number of epochs. The proposed MLFNN benefits from the eight parameters as inputs and different number of priority rules as outputs. The developed MLFNN was structured with eleven, seven, five, four and three priority rules or neurons as output layer. It was found that the training performance of the MLFNN increases when the fixed number of outputs is reduced, and consequently the proposed algorithm has reasonable performance in coping with the RCPSP when the number of outputs used is reduced.

The developed convolutional neural network is also trained on the created dataset to determine the optimal weights. Like the MLFNN, the evolved convolutional neural network (CNN) is fed with eight different project parameters. Moreover, To verify the performance of the proposed CNN, a different number of neurons or priority rules are assigned as outputs to the output layer. The developed CNN is equipped with one, two and three convolutional layers to analyze the training performance. This means that the developed convolutional network was verified with one, two and three convolutional layers and three fully connected layers. It was found that the training performance of the proposed CNN increases when the number of outputs is reduced, and consequently, the developed algorithm has reasonable performance to solve the RCPSP when the number of neural network outputs used is reduced.

Table 4.7 summarizes the results obtained with the two algorithms proposed in Chapters 3 and 4. In general, the results declare that the developed CNN proposed in Chapter 4 performs better than the MLFNN developed in Chapter 3 in selecting an appropriate priority rule as output. The results presented in Table 4.7 also claim that the average deviations from the critical path lower bound are better when the performance of the developed neural networks increases.

The results obtained by the two proposed algorithms are not the best among other results obtained by other researchers, but as mentioned before, the advantage of the proposed algorithms over evolutionary methods or metaheuristics is that there is no need

Priority rules used for output layer	Number of activities			
	J60		J120	
	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, ST, TRS, GRPW, and WRUP	63.33	58.96	124.72	117.42

Table 4.7 – Percentage of average deviations from critical path lower bound for the J60 and J120. The average deviations obtained by the algorithm using the CNN and MLFNN approaches show that the CNN approach performs better compared to the MLFNN approach

to generate populations or repetitive iterations, on the contrary, the proposed algorithm generates an activity sequence according to the trained MLFNN and CNN.

The results obtained in this thesis declare that the performance of the proposed algorithms and the results can be improved. This work can serve as a basis for researchers who want to use neural networks to schedule projects. To improve the results, I propose to select appropriate priority rules as outputs and develop more neural networks. In addition, the result of the proposed algorithms can be more competitive if the training dataset is specialized.

BIBLIOGRAPHY

- Abiodun, Oludare Isaac et al., « State-of-the-art in artificial neural network applications: A survey », *in: Heliyon* 4.11 (2018), e00938.
- Afshar-Nadjafi, Behrouz, Mehdi Yazdani, and Mahyar Majlesi, « A hybrid of tabu search and simulated annealing algorithms for preemptive project scheduling problem », *in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, 2017, pp. 102–111.
- Agarwal, Anurag, Selcuk Colak, and Selcuk Erenguc, « A neurogenetic approach for the resource-constrained project scheduling problem », *in: Computers & operations research* 38.1 (2011), pp. 44–50.
- « Metaheuristic methods », *in: Handbook on Project Management and Scheduling Vol. 1*, Springer, 2015, pp. 57–74.
- Agarwal, Anurag, Hasan Pirkul, and Varghese S Jacob, « Augmented neural networks for task scheduling », *in: European Journal of Operational Research* 151.3 (2003), pp. 481–502.
- Aggarwal, Charu C et al., « Neural networks and deep learning », *in: Springer* 10 (2018), pp. 978–3.
- Akbari, Reza, Vahid Zeighami, and Koorush Ziarati, « A Cooperative Artificial Bee Colony Optimizer », *in: 7th International Industrial Engineering Conference*, 2010, pp. 1–10.
- « Artificial bee colony for resource constrained project scheduling problem », *in: International Journal of Industrial Engineering Computations* 2.1 (2011), pp. 45–60.
- Alcaraz, Javier and Concepción Maroto, « A hybrid genetic algorithm based on intelligent encoding for project scheduling », *in: Perspectives in modern project scheduling*, Springer, 2006, pp. 249–274.
- « A robust genetic algorithm for resource allocation in project scheduling », *in: Annals of operations Research* 102.1 (2001), pp. 83–109.
- Ali, Ismail M et al., « Memetic algorithm for solving resource constrained project scheduling problems », *in: 2015 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2015, pp. 2761–2767.

-
- Alipouri, Yagub et al., « Solving the FS-RCPSP with hyper-heuristics: A policy-driven approach », *in: Journal of the Operational Research Society* 70.3 (2019), pp. 403–419.
- Aloysius, Neena and M Geetha, « A review on deep convolutional neural networks », *in: 2017 international conference on communication and signal processing (ICCSP)*, IEEE, 2017, pp. 0588–0592.
- Anagnostopoulos, Konstantinos P and Georgios K Koulinas, « A genetic hyperheuristic algorithm for the resource constrained project scheduling problem », *in: IEEE Congress on Evolutionary Computation*, IEEE, 2010, pp. 1–6.
- Anantathanvit, Mana and Mud-Armeen Munlin, « Radius particle swarm optimization for resource constrained project scheduling problem », *in: 16th Int'l Conf. Computer and Information Technology*, IEEE, 2014, pp. 24–29.
- Atli, Omer, « Tabu search and an exact algorithm for the solutions of resource-constrained project scheduling problems », *in: International Journal of Computational Intelligence Systems* 4.2 (2011), pp. 255–267.
- Ballestin, Francisco, « A genetic algorithm for the resource renting problem with minimum and maximum time lags », *in: European Conference on Evolutionary Computation in Combinatorial Optimization*, Springer, 2007, pp. 25–35.
- Ballestin, Francisco, Agustin Barrios, and Vicente Valls, « An evolutionary algorithm for the resource-constrained project scheduling problem with minimum and maximum time lags », *in: Journal of scheduling* 14.4 (2011), pp. 391–406.
- Bettemir, Önder Halis and Rifat Sonmez, « Hybrid genetic algorithm with simulated annealing for resource-constrained project scheduling », *in: Journal of Management in Engineering* 31.5 (2015), p. 04014082.
- Blazewicz, Jacek, Jan Karel Lenstra, and AHG Rinnooy Kan, « Scheduling subject to resource constraints: classification and complexity », *in: Discrete applied mathematics* 5.1 (1983), pp. 11–24.
- Boctor, Fayed F, « Resource-constrained project scheduling by simulated annealing », *in: International Journal of Production Research* 34.8 (1996), pp. 2335–2351.
- Bouleimen, KLEIN and HOUSNI Lecocq, « A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version », *in: European journal of operational research* 149.2 (2003), pp. 268–281.
- Chen, Ruey-Maw, « Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem », *in: Expert Systems with Applications* 38.6 (2011), pp. 7102–7111.

-
- Chen, Ruey-Maw and Shih-Tang Lo, « Using an enhanced ant colony system to solve resource-constrained project scheduling problem », *in: Int. J. Comput. Sci. Netw. Secur* 6 (2006), pp. 75–84.
- Chen, Ruey-Maw et al., « Using novel particle swarm optimization scheme to solve resource-constrained scheduling problem in PSPLIB », *in: Expert systems with applications* 37.3 (2010), pp. 1899–1910.
- Chen, Wang et al., « An efficient hybrid algorithm for resource-constrained project scheduling », *in: Information Sciences* 180.6 (2010), pp. 1031–1039.
- Cho, JH and Yeong-Dae Kim, « A simulated annealing algorithm for resource constrained project scheduling problems », *in: Journal of the Operational Research Society* 48.7 (1997), pp. 736–744.
- Choi, Rene Y et al., « Introduction to machine learning, neural networks, and deep learning », *in: Translational Vision Science & Technology* 9.2 (2020), pp. 14–14.
- Crawford, Broderick et al., « An artificial bee colony algorithm for the resource constrained project scheduling problem », *in: International Conference on Human-Computer Interaction*, Springer, 2015, pp. 582–586.
- Das, Partha Pratim and Sriyankar Acharyya, « Simulated annealing variants for solving resource constrained project scheduling problem: A comparative study », *in: 14th International Conference on Computer and Information Technology (ICIT 2011)*, IEEE, 2011, pp. 469–474.
- Debels, Dieter and Mario Vanhoucke, « A bi-population based genetic algorithm for the resource-constrained project scheduling problem », *in: International Conference on Computational Science and Its Applications*, Springer, 2005, pp. 378–387.
- « A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem », *in: Operations Research* 55.3 (2007), pp. 457–469.
- Delgoshaei, Aidin et al., « Minimizing makespan of a resource-constrained scheduling problem: A hybrid greedy and genetic algorithms », *in: International Journal of Industrial Engineering Computations* 6.4 (2015), pp. 503–520.
- Deng, Linyi, Van Lin, and Ming Chen, « Hybrid ant colony optimization for the resource-constrained project scheduling problem », *in: Journal of Systems Engineering and Electronics* 21.1 (2010), pp. 67–71.
- Dhillon, Anamika and Gyanendra K Verma, « Convolutional neural network: a review of models, methodologies and applications to object detection », *in: Progress in Artificial Intelligence* 9.2 (2020), pp. 85–112.

-
- Diana, S, L Ganapathy, and Ashok K Pundir, « An improved genetic algorithm for resource constrained project scheduling problem », *in: International Journal of Computer Applications* 78.9 (2013).
- Dong, Ning et al., « A genetic algorithm-based method for look-ahead scheduling in the finishing phase of construction projects », *in: Advanced Engineering Informatics* 26.4 (2012), pp. 737–748.
- Dorigo, Marco and Gianni Di Caro, « Ant colony optimization: a new meta-heuristic », *in: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 2, IEEE, 1999, pp. 1470–1477.
- Dridi, Olfa, Saoussen Krichen, and Adel Guitouni, « Solving resource-constrained project scheduling problem by a genetic local search approach », *in: 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, IEEE, 2013, pp. 1–5.
- Eberhart, Russell and James Kennedy, « Particle swarm optimization », *in: Proceedings of the IEEE international conference on neural networks*, vol. 4, Citeseer, 1995, pp. 1942–1948.
- Eshraghi, Arian, « A new approach for solving resource constrained project scheduling problems using differential evolution algorithm », *in: International journal of industrial engineering computations* 7.2 (2016), pp. 205–216.
- Fang, Chen, Ling Wang, and Ye Xu, « An estimation of distribution algorithm for resource-constrained project scheduling problem », *in: 2010 Chinese Control and Decision Conference*, IEEE, 2010, pp. 265–270.
- Fang, Chen et al., « An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem », *in: Flexible Services and Manufacturing Journal* 27.4 (2015), pp. 585–605.
- Fathollahi, Fatemeh and AA Naja, « A hybrid genetic algorithm to maximize net present value of project cash flows in resource-constrained project scheduling problem with fuzzy parameters », *in: Scientia Iranica. Transaction E, Industrial Engineering* 23.4 (2016), p. 1893.
- Frankola, Toni, Marin Golub, and Domagoj Jakobovic, « Evolutionary algorithms for the resource-constrained scheduling problem », *in: ITI 2008-30th International Conference on Information Technology Interfaces*, IEEE, 2008, pp. 715–722.

-
- Gargiulo, F and D Quagliarella, « Genetic algorithms for the resource constrained project scheduling problem », *in: 2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*, IEEE, 2012, pp. 39–47.
- Gendreau, Michel, Jean-Yves Potvin, et al., *Handbook of metaheuristics*, vol. 2, Springer, 2010.
- Golab, Amir et al., « Investigating the performance of an artificial neural network for solving the resource constrained project scheduling problem (RCPSP) », *in: Pattern Recognition and Tracking XXXIII*, vol. 12101, SPIE, 2022, pp. 78–83.
- Golab, Amir et al., « Review of conventional metaheuristic techniques for resource-constrained project scheduling problem », *in: Journal of Project Management 7.2* (2022), pp. 95–110.
- Golab, A et al., « A convolutional neural network for the resource-constrained project scheduling problem (RCPSP): A new approach », *in: Decision Science Letters 12.2* (2023), pp. 225–238.
- « A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP) », *in: Decision Science Letters 11.4* (2022), pp. 407–418.
- Gonçalves, José Fernando, Mauricio GC Resende, and Jorge JM Mendes, « A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem », *in: Journal of Heuristics 17.5* (2011), pp. 467–486.
- Goncharov, Evgenii N and Valentin V Leonov, « Genetic algorithm for the resource-constrained project scheduling problem », *in: Automation and Remote Control 78.6* (2017), pp. 1101–1114.
- Habibi, Farhad, Farnaz Barzinpour, and Seyed Sadjadi, « Resource-constrained project scheduling problem: review of past and recent developments », *in: Journal of project management 3.2* (2018), pp. 55–88.
- Han, Lirong et al., « Immunomodulatory activity of a water-soluble polysaccharide obtained from highland barley on immunosuppressive mice models », *in: Food & function 10.1* (2019), pp. 304–314.
- Hartmann, Sönke, « A competitive genetic algorithm for resource-constrained project scheduling », *in: Naval Research Logistics (NRL) 45.7* (1998), pp. 733–750.
- « A self-adapting genetic algorithm for project scheduling under resource constraints », *in: Naval Research Logistics (NRL) 49.5* (2002), pp. 433–448.

-
- Hindi, Khalil S., Hongbo Yang, and Krzysztof Fleszar, « An evolutionary algorithm for resource-constrained project scheduling », *in: IEEE Transactions on evolutionary computation* 6.5 (2002), pp. 512–518.
- Jędrzejowicz, Piotr and Ewa Ratajczak-Ropel, « Reinforcement learning strategies for A-team solving the resource-constrained project scheduling problem », *in: Neurocomputing* 146 (2014), pp. 301–307.
- Jia, Qiong and Yoonho Seo, « An improved particle swarm optimization for the resource-constrained project scheduling problem », *in: The International Journal of Advanced Manufacturing Technology* 67.9 (2013), pp. 2627–2638.
- « Solving resource-constrained project scheduling problems: conceptual validation of FLP formulation and efficient permutation-based ABC computation », *in: Computers & Operations Research* 40.8 (2013), pp. 2037–2050.
- Joshi, Dheeraj et al., « An effective teaching-learning-based optimization algorithm for the multi-skill resource-constrained project scheduling problem », *in: Journal of Modelling in Management* (2019).
- Joy, Jiby, Sriyith Rajeev, and Vishnu Narayanan, « Particle swarm optimization for resource constrained-project scheduling problem with varying resource levels », *in: Procedia Technology* 25 (2016), pp. 948–954.
- Kadam, Sachin U and Narendra S Kadam, « Solving resource-constrained project scheduling problem by genetic algorithm », *in: 2014 2nd International Conference on Business and Information Management (ICBIM)*, IEEE, 2014, pp. 159–164.
- Kadam, Sachin Uttam and Sandip U Mane, « A genetic-local search algorithm approach for resource constrained project scheduling problem », *in: 2015 International Conference on Computing Communication Control and Automation*, IEEE, 2015, pp. 841–846.
- Kadri, Roubila Lilia and Fayez F Boctor, « An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case », *in: European Journal of Operational Research* 265.2 (2018), pp. 454–462.
- Kanit, Recep, Omer Ozkan, and Murat Gunduz, « Effects of project size and resource constraints on project duration through priority rule-base heuristics », *in: Artificial Intelligence Review* 32.1 (2009), pp. 115–123.
- Khalili, Somayeh, Amir Abbas Najafi, and Seyed Taghi Akhavan Niaki, « Bi-objective resource constrained project scheduling problem with makespan and net present value

-
- criteria: two meta-heuristic algorithms », *in: The International Journal of Advanced Manufacturing Technology* 69.1 (2013), pp. 617–626.
- Kim, Jin-Lee, « Improved genetic algorithm for resource-constrained scheduling of large projects », *in: Canadian journal of civil engineering* 36.6 (2009), pp. 1016–1027.
- « Permutation-based elitist genetic algorithm using serial scheme for large-sized resource-constrained project scheduling », *in: 2007 Winter Simulation Conference*, IEEE, 2007, pp. 2112–2118.
- Kim, Jin-Lee and Ralph D Ellis Jr, « Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling », *in: Journal of construction engineering and management* 134.11 (2008), pp. 904–913.
- Kim, Kwan Woo, Mitsuo Gen, and Genji Yamazaki, « Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling », *in: Applied soft computing* 2.3 (2003), pp. 174–188.
- Kiranyaz, Serkan et al., « 1D convolutional neural networks and applications: A survey », *in: Mechanical systems and signal processing* 151 (2021), pp. 107–398.
- Klimek, Marcin, « A genetic algorithm for the project scheduling with the resource constraints », *in: Annales Universitatis Mariae Curie-Skłodowska. Sectio AI, Informatica* 10.1 (2010).
- Kolisch, Rainer, « Efficient priority rules for the resource-constrained project scheduling problem », *in: Journal of Operations Management* 14.3 (1996), pp. 179–192.
- Kolisch, Rainer and Sanke Hartmann, « Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis », *in: Project scheduling*, Springer, 1999, pp. 147–178.
- Kolisch, Rainer and Sönke Hartmann, « Experimental investigation of heuristics for resource-constrained project scheduling: An update », *in: European journal of operational research* 174.1 (2006), pp. 23–37.
- Kolisch, Rainer and Arno Sprecher, « PSPLIB—a project scheduling problem library: OR software-ORSEP operations research software exchange program », *in: European journal of operational research* 96.1 (1997), pp. 205–216.
- Kolisch, Rainer, Arno Sprecher, and Andreas Drexl, « Characterization and generation of a general class of resource-constrained project scheduling problems », *in: Management science* 41.10 (1995), pp. 1693–1703.
- Koulinas, Georgios, Lazaros Kotsikas, and Konstantinos Anagnostopoulos, « A particle swarm optimization based hyper-heuristic algorithm for the classic resource con-

-
- strained project scheduling problem », *in: Information Sciences* 277 (2014), pp. 680–693.
- Kumar, Neetesh and Deo Prakash Vidyarthi, « A model for resource-constrained project scheduling using adaptive PSO », *in: Soft computing* 20.4 (2016), pp. 1565–1580.
- Li, Fei, Changtao Lai, and Yongyi Shou, « Particle swarm optimization for preemptive project scheduling with resource constraints », *in: 2011 IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE, 2011, pp. 869–873.
- Li, Ming et al., « A particle swarm optimization algorithm with crossover for resource constrained project scheduling problem », *in: 2009 IITA International Conference on Services Science, Management and Engineering*, IEEE, 2009, pp. 69–72.
- Li, Zewen et al., « A survey of convolutional neural networks: analysis, applications, and prospects », *in: IEEE transactions on neural networks and learning systems* (2021).
- Lim, Andrew et al., « New meta-heuristics for the resource-constrained project scheduling problem », *in: Flexible Services and Manufacturing Journal* 25.1 (2013), pp. 48–73.
- Liu, Jia et al., « Solving resource-constrained project scheduling problem via genetic algorithm », *in: Journal of Computing in Civil Engineering* 34.2 (2020), p. 04019055.
- Lo, Shih-Tang et al., « Using particle swarm optimization to solve resource-constrained scheduling problems », *in: 2008 IEEE Conference on Soft Computing in Industrial Applications*, IEEE, 2008, pp. 38–43.
- Luo, Shipeng, Cheng Wang, and Jinwen Wang, « Ant colony optimization for resource-constrained project scheduling with generalized precedence relations », *in: Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, 2003, pp. 284–289.
- Mendes, Jorge JM, José Fernando Gonçalves, and Mauricio GC Resende, « A random key based genetic algorithm for the resource constrained project scheduling problem », *in: Computers & operations research* 36.1 (2009), pp. 92–109.
- Merkle, Daniel, Martin Middendorf, and Hartmut Schmeck, « Ant colony optimization for resource-constrained project scheduling », *in: IEEE transactions on evolutionary computation* 6.4 (2002), pp. 333–346.
- Mobini, Mahdi, Zahra Mobini, and Masoud Rabbani, « An Artificial Immune Algorithm for the project scheduling problem under resource constraints », *in: Applied soft computing* 11.2 (2011), pp. 1975–1982.

-
- Montoya-Torres, Jairo R, Edgar Gutierrez-Franco, and Carolina Pirachicán-Mayorga, « Project scheduling with limited resources using a genetic algorithm », *in: International Journal of Project Management* 28.6 (2010), pp. 619–628.
- Munlin, M and M Anantathanavit, « Hybrid radius particle swarm optimization », *in: 2016 IEEE Region 10 Conference (TENCON)*, IEEE, 2016, pp. 2180–2184.
- Munlin, Mudarmeen, « Solving resource-constrained project scheduling problem using metaheuristic algorithm », *in: 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, IEEE, 2018, pp. 344–349.
- Myszkowski, Paweł B et al., « Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem », *in: Soft Computing* 19.12 (2015), pp. 3599–3619.
- Nasiri, Mohammad Mahdi, « A pseudo particle swarm optimization for the RCPSP », *in: The International Journal of Advanced Manufacturing Technology* 65.5 (2013), pp. 909–918.
- Nonobe, Koji and Toshihide Ibaraki, « Formulation and tabu search algorithm for the resource constrained project scheduling problem », *in: Essays and surveys in metaheuristics*, Springer, 2002, pp. 557–588.
- Nwankpa, Chigozie et al., « Activation functions: Comparison of trends in practice and research for deep learning », *in: arXiv preprint arXiv:1811.03378* (2018).
- Olaguibel, Ramon Alvarez-Valdes and Jose Manuel Tamarit Goerlich, « Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis », *in: Advances in project scheduling* (1989), pp. 113–134.
- Ortiz-Pimiento, Nestor Raul and Francisco Javier Diaz-Serna, « The project scheduling problem with non-deterministic activities duration: A literature review », *in: Journal of Industrial Engineering and Management (JIEM)* 11.1 (2018), pp. 116–134.
- Özkan, Ömer and Ümit Gülçiçek, « A neural network for resource constrained project scheduling programming », *in: Journal of Civil Engineering and Management* 21.2 (2015), pp. 193–200.
- Pan, Nai-Hsin, Po-Wen Hsiao, and Kuei-Yen Chen, « A study of project scheduling optimization using Tabu Search algorithm », *in: Engineering Applications of Artificial Intelligence* 21.7 (2008), pp. 1101–1112.
- Pan, Nai-Hsin and Yung-Yu Lin, « Using hybrid simulated annealing algorithm in resource constrained project scheduling problem », *in: Journal of Statistics and Management Systems* 14.3 (2011), pp. 555–582.

-
- Pan, Xiaoying and Hao Chen, « A multi-agent social evolutionary algorithm for resource-constrained project scheduling », *in: 2010 International Conference on Computational Intelligence and Security*, IEEE, 2010, pp. 209–213.
- Peng, Wuliang and Yonghe Wei, « PSO for solving RCPSP », *in: 2008 Chinese control and decision conference*, IEEE, 2008, pp. 818–822.
- Pham, Duc Truong et al., « The bees algorithm—a novel tool for complex optimisation problems », *in: Intelligent production machines and systems*, Elsevier, 2006, pp. 454–459.
- PMI, *A guide to the project management body of knowledge, PMBOK GUIDE Seventh Edition*, Chicago: Project management institute, 2021.
- *A guide to the project management body of knowledge, PMBOK GUIDE Sixth Edition*, Chicago: Project management institute, 2017.
- Potvin, Jean-Yves and Michel Gendreau, *Handbook of Metaheuristics*, Springer, 2018.
- Proon, Sepehr and Mingzhou Jin, « A genetic algorithm with neighborhood search for the resource-constrained project scheduling problem », *in: Naval Research Logistics (NRL) 58.2* (2011), pp. 73–82.
- Quoc, Huu Dang et al., « New effective differential evolution algorithm for the project scheduling problem », *in: 2020 2nd International Conference on Computer Communication and the Internet (ICCCI)*, IEEE, 2020, pp. 150–157.
- Rao, R Venkata, Vimal J Savsani, and DP Vakharia, « Teaching–learning–based optimization: a novel method for constrained mechanical design optimization problems », *in: Computer-aided design 43.3* (2011), pp. 303–315.
- Rawat, Waseem and Zenghui Wang, « Deep convolutional neural networks for image classification: A comprehensive review », *in: Neural computation 29.9* (2017), pp. 2352–2449.
- Ren, Yan Hua, De Cai Kong, and Wu Liang Peng, « A genetic algorithm based solution with schedule mode for RCPSP », *in: Advanced Materials Research*, vol. 268, Trans Tech Publ, 2011, pp. 1802–1805.
- Roy, Bidisha and Asim Kumar Sen, « A novel metaheuristic approach for resource constrained project scheduling problem », *in: Soft Computing: Theories and Applications*, Springer, 2020, pp. 535–544.
- « Meta-heuristic techniques to solve resource-constrained project scheduling problem », *in: International conference on innovative computing and communications*, Springer, 2019, pp. 93–99.

-
- Sadeghi, Amir et al., « Using bees algorithm to solve the resource constrained project scheduling problem in PSPLIB », *in: International Conference on Theoretical and Mathematical Foundations of Computer Science*, Springer, 2011, pp. 486–494.
- Sakalauskas, Leonidas and Gražvydas Felinskas, « Optimization of resource constrained project schedules by genetic algorithm based on the job priority list », *in: Information technology and control* 35.4 (2006).
- Sallam, Karam M, Ripon K Chakraborty, and Michael J Ryan, « A hybrid differential evolution with cuckoo search for solving resource constrained project scheduling problems », *in: 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, IEEE, 2019, pp. 1344–1348.
- « A reinforcement learning based multi-method approach for stochastic resource constrained project scheduling problems », *in: Expert Systems with Applications* 169 (2021), p. 114479.
- Sanaei, Pejman et al., « Using firefly algorithm to solve resource constrained project scheduling problem », *in: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, Springer, 2013, pp. 417–428.
- Shan, Miyuan, Juan Wu, and Danni Peng, « Particle swarm and ant colony algorithms hybridized for multi-mode resource-constrained project scheduling problem with minimum time lag », *in: 2007 International Conference on Wireless Communications, Networking and Mobile Computing*, IEEE, 2007, pp. 5898–5902.
- Sharma, Sagar, Simone Sharma, and Anidhya Athaiya, « Activation functions in neural networks », *in: towards data science* 6.12 (2017), pp. 310–316.
- Shi, Yan-jun et al., « An artificial bee colony with random key for resource-constrained project scheduling », *in: Life system modeling and intelligent computing*, Springer, 2010, pp. 148–157.
- Shou, Yongyi, « A Bi-directional Ant colony algorithm for resource constrained project scheduling », *in: 2007 IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE, 2007, pp. 1027–1031.
- Shou, Yongyi, Ying Li, and Changtao Lai, « Hybrid particle swarm optimization for preemptive resource-constrained project scheduling », *in: Neurocomputing* 148 (2015), pp. 122–128.

-
- Skowronski, Marek E et al., « Tabu search approach for multi-skill resource-constrained project scheduling problem », *in: 2013 Federated Conference on Computer Science and Information Systems*, IEEE, 2013, pp. 153–158.
- Stiti, Cyrine and Olfa Belkahlia Driss, « A new approach for the multi-site resource-constrained project scheduling problem », *in: Procedia Computer Science* 164 (2019), pp. 478–484.
- Sung, Inkyung, Bongjun Choi, and Peter Nielsen, « Reinforcement Learning for Resource Constrained Project Scheduling Problem with Activity Iterations and Crashing », *in: IFAC-PapersOnLine* 53.2 (2020), pp. 10493–10497.
- Svozil, Daniel, Vladimir Kvasnicka, and Jiri Pospichal, « Introduction to multi-layer feed-forward neural networks », *in: Chemometrics and intelligent laboratory systems* 39.1 (1997), pp. 43–62.
- Tahooneh, Amin and Koorush Ziarati, « Using artificial bee colony to solve stochastic resource constrained project scheduling problem », *in: International Conference in Swarm Intelligence*, Springer, 2011, pp. 293–302.
- Tchomte, Sylverin Kemmoe and Michel Gourgand, « Particle swarm optimization: A study of particle displacement for solving continuous and combinatorial optimization problems », *in: International Journal of Production Economics* 121.1 (2009), pp. 57–67.
- Thammano, Arit and Ajchara Phu-Ang, « A hybrid evolutionary algorithm for the resource-constrained project scheduling problem », *in: Artificial Life and Robotics* 17.2 (2012), pp. 312–316.
- Thomas, Paul R and Said Salhi, « A tabu search approach for the resource constrained project scheduling problem », *in: Journal of heuristics* 4.2 (1998), pp. 123–139.
- Tseng, Lin-Yu and Shih-Chieh Chen, « A hybrid metaheuristic for the resource-constrained project scheduling problem », *in: European Journal of Operational Research* 175.2 (2006), pp. 707–721.
- Ulusoy, Gündüz and Linet Özdamar, « Heuristic performance and network/resource characteristics in resource-constrained project scheduling », *in: Journal of the Operational Research Society* 40.12 (1989), pp. 1145–1152.
- Valls, Vicente, Francisco Ballestin, and Sacramento Quintanilla, « A hybrid genetic algorithm for the resource-constrained project scheduling problem », *in: European journal of operational research* 185.2 (2008), pp. 495–508.

-
- Wang, Hong, Tongling Li, and Dan Lin, « Efficient genetic algorithm for resource-constrained project scheduling problem », *in: Transactions of Tianjin University* 16.5 (2010), pp. 376–382.
- Wang, Hong, Dan Lin, and Minqiang Li, « A genetic algorithm for solving fuzzy resource-constrained project scheduling », *in: International Conference on Natural Computation*, Springer, 2005, pp. 171–180.
- Wang, Ling and Chen Fang, « A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem », *in: Expert Systems with Applications* 39.3 (2012), pp. 2451–2460.
- Wang, Qiang and Jianxun Qi, « Improved particle swarm optimization for RCP scheduling problem », *in: The Sixth International Symposium on Neural Networks (ISNN 2009)*, Springer, 2009, pp. 49–57.
- Wang, Sun-Chong, « Artificial neural network », *in: Interdisciplinary computing in java programming*, Springer, 2003, pp. 81–100.
- Yu, Xiaoguang et al., « A novel genetic simulated annealing algorithm for the resource-constrained project scheduling problem », *in: 2009 International Workshop on Intelligent Systems and Applications*, IEEE, 2009, pp. 1–4.
- Zamani, Reza, « A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem », *in: European journal of operational research* 229.2 (2013), pp. 552–559.
- « An evolutionary implicit enumeration procedure for solving the resource-constrained project scheduling problem », *in: International Transactions in Operational Research* 24.6 (2017), pp. 1525–1547.
- Zeighami, Vahid et al., « An ABC-Genetic method to solve resource constrained project scheduling problem », *in: Artificial Intelligence Research (AIR) Journal, SCIEDU, Canada* 1.2 (2012), pp. 185–197.
- Zhang, Hong, Heng Li, and CM Tam, « Permutation-based particle swarm optimization for resource-constrained project scheduling », *in: Journal of computing in civil engineering* 20.2 (2006), pp. 141–149.
- Zhang, Hong et al., « Particle swarm optimization-based schemes for resource-constrained project scheduling », *in: Automation in construction* 14.3 (2005), pp. 393–404.
- Zhang, Kai, Guorong Zhao, and Jing Jiang, « Particle swarm optimization method for resource-constrained project scheduling problem », *in: 2009 9th International Conference on Electronic Measurement & Instruments*, IEEE, 2009, pp. 4–792.

-
- Zheng, Huan-yu and Ling Wang, « An effective teaching–learning-based optimisation algorithm for RCPSP with ordinal interval numbers », *in: International Journal of Production Research* 53.6 (2015), pp. 1777–1790.
- Zheng, Huan-yu, Ling Wang, and Sheng-yao Wang, « A co-evolutionary teaching-learning-based optimization algorithm for stochastic RCPSP », *in: 2014 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2014, pp. 587–594.
- Zheng, Huan-yu, Ling Wang, and Xiao-long Zheng, « Teaching–learning-based optimization algorithm for multi-skill resource constrained project scheduling problem », *in: Soft Computing* 21.6 (2017), pp. 1537–1548.
- Zhou, Yumiao, Qingshun Guo, and Rongwei Gan, « Improved ACO algorithm for resource-constrained project scheduling problem », *in: 2009 international conference on artificial intelligence and computational intelligence*, vol. 3, IEEE, 2009, pp. 358–365.
- Zhu, Jie, Xiaoping Li, and Weiming Shen, « Effective genetic algorithm for resource-constrained project scheduling with limited preemptions », *in: International Journal of Machine Learning and Cybernetics* 2.2 (2011), pp. 55–65.
- Ziarati, Koorush, Reza Akbari, and Vahid Zeighami, « On the performance of bee algorithms for resource-constrained project scheduling problem », *in: Applied Soft Computing* 11.4 (2011), pp. 3720–3733.

LIST OF PUBLICATIONS

Golab, Amir et al., « Review of conventional metaheuristic techniques for resource-constrained project scheduling problem », in: *Journal of Project Management* 7.2 (2022), pp. 95–110.

Golab, Amir et al., « Investigating the performance of an artificial neural network for solving the resource-constrained project scheduling problem (RCPSP) », in: *Pattern Recognition and Tracking XXXIII*, vol. 12101, SPIE, 2022, pp. 78–83.

Golab, A et al., « A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP) », in: *Decision Science Letters* 11.4 (2022), pp. 407–418.

Golab, A et al., « A convolutional neural network for the resource-constrained project scheduling problem (RCPSP): A new approach », in: *Decision Science Letters*, 12.2 (2023), pp. 225-238.

Titre : Automatisation de la planification dynamique de projet à ressource limitée par l'intelligence artificielle

Mots clés : Planification de projet, gestion de projet, réseau neuronal artificiel, RCPSP, contrainte de ressources, planification

L'exécution de divers projets joue un rôle important dans le développement mondial et la croissance économique. En outre, tous les projets ont besoin d'une structure pour atteindre les objectifs du projet. Je me suis concentré sur le problème de planification de projet sous contrainte de ressources (RCPSP) parce qu'il fait partie de la gestion du calendrier de projet. Le (RCPSP) a deux contraintes importantes, à savoir les contraintes de ressources et les relations de préséance pendant la planification du projet. L'objectif du problème est de minimiser la durée du projet. Je développe deux approches différentes de réseaux neuronaux intégrées dans deux algorithmes pour résoudre le RCPSP. Les réseaux neuronaux développés apprennent en fonction des huit paramètres du projet qui sont recalculés à chaque étape du planning du projet. 11 règles de priorité sont les sorties des réseaux neuronaux évolués. Les réseaux neuronaux évolués sont utilisés pour sélectionner une règle de priorité appropriée pour filtrer une activité appropriée. Par conséquent, les réseaux neuronaux peuvent sélectionner automatiquement une règle de priorité pour filtrer une tâche éligible. Les algorithmes planifient toutes les tâches du projet en fonction des contraintes du projet. L'avantage de ces algorithmes par rapport aux méthodes conventionnelles telles que les métaheuristiques est qu'ils ne génèrent pas nécessairement de nombreuses solutions ou populations.

L'exécution de divers projets joue un rôle important dans le développement national et la croissance économique. En outre, tous les projets ont besoin d'une structure pour atteindre les objectifs du projet. Je me suis concentré sur le problème de planification de projet sous contrainte de ressources (RCPSP) parce qu'il fait partie de la gestion du calendrier de projet. Le (RCPSP) a deux contraintes importantes, à savoir les contraintes de ressources et les relations de préséance pendant la planification du projet. L'objectif du problème est de minimiser la durée du projet. Je développe deux approches différentes de réseaux neuronaux intégrées dans deux algorithmes pour résoudre le RCPSP. Les réseaux neuronaux développés apprennent en fonction des huit paramètres du projet qui sont recalculés à chaque étape du planning du projet. Les 11 règles de priorité sont les sorties des réseaux neuronaux évolués. Les réseaux neuronaux évolués sont utilisés pour sélectionner une règle de priorité appropriée pour filtrer une activité appropriée. Par conséquent, les réseaux neuronaux peuvent sélectionner automatiquement une règle de priorité pour filtrer une tâche éligible. Les algorithmes planifient toutes les tâches du projet en fonction des contraintes du projet. L'avantage de ces algorithmes par rapport aux méthodes conventionnelles telles que les métaheuristiques est qu'ils ne génèrent pas nécessairement de nombreuses solutions ou populations.

Title : Resource constrained project scheduling problem (RCPSP) using artificial intelligence

Keywords : Project scheduling, Project management, Artificial neural network, RCPSP, Resource constraint, scheduling

The implementation of various projects plays an important role in national development and economic growth. In addition, all projects need a structure to achieve the project goals. I focused on the resource constrained project scheduling problem (RCPSP) because it is a part of project schedule management. The (RCPSP) has two important constraints, namely resource constraints and precedence relationships during project scheduling. The objective of the problem is to minimize the project duration. I develop two different neural network approaches embedded in two algorithms to solve the RCPSP. The developed neural networks learn according to the eight project parameters that are recalculated at each step of the project schedule. 11 Priority rules are the outputs of the evolved NNs. The evolved neural networks are used to select an appropriate priority rule to filter out an appropriate activity. Therefore, the neural networks can automatically select a priority rule to filter an eligible activity. Consequently, the algorithms schedule all project activities according to the given project constraints. The advantage of these algorithms over conventional methods such as metaheuristics is that they do not necessarily generate many solutions or populations.

The implementation of various projects plays an important role in national development and economic growth. In addition, all projects need a structure to achieve the project goals. I focused on the resource constrained project scheduling problem (RCPSP) because it is a part of project schedule management. The (RCPSP) has two important constraints, namely resource constraints and precedence relationships during project scheduling. The objective of the problem is to minimize the project duration. I develop two different neural network approaches embedded in two algorithms to solve the RCPSP. The developed neural networks learn according to the eight project parameters that are recalculated at each step of the project schedule. 11 Priority rules are the outputs of the evolved NNs. The evolved neural networks are used to select an appropriate priority rule to filter out an appropriate activity. Therefore, the neural networks can automatically select a priority rule to filter an eligible activity. Consequently, the algorithms schedule all project activities according to the given project constraints. The advantage of these algorithms over conventional methods such as metaheuristics is that they do not necessarily generate many solutions or populations.