UNIVERSIDAD DEL NORTE

DOCTORAL THESIS

Reactive Scheduling to Treat Disruptive Events in the MRCPSP

Author:
LUIS FERNANDO MACHADO DOMINGUEZ

Supervisor:
Dr. CARLOS PATERNINA-ARBOLEDA

Co-Supervisor:
Dr. AGUSTIN BARRIOS SARMIENTO

A thesis submitted in fulfillment of the requirements for the degree of Ph.D. in Industrial Engineering

in the

Industrial Engineering Department
Faculty of Engineering

July 29, 2021

Reactive Scheduling to Treat Disruptive Events in the MRCPSP

by

LUIS FERNANDO MACHADO DOMINGUEZ

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Ph.D. in Industrial Engineering

at

Universidad del Norte
July, 2021

Declaration

I, LUIS FERNANDO MACHADO DOMINGUEZ, declare that this thesis titled, “Reactive Scheduling to Treat Disruptive Events in the MRCPSP”, which is submitted in fulfillment of the requirements for the Degree of Ph.D. in Industrial Engineering, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Luis F. Machado D.
Signed: ______________________

Date: July 29, 2021

For Emmy y Lisse

Acknowledgements

To God, I owe everything, faith, and hope to move forward and not faint in those hard moments that only you know my Lord. For the strength to finish this work entitled "Reactive Scheduling to Treat Disruptive Events in the MRCPSP".

I want to thank my supervisor Dr. Carlos Paternina for accepting to be his doctoral student. For his wise advice, and for guiding my way in the dark moments.

I also want to thank my co-supervisor Dr. Agustin Barrios and his wife Dr. Mildred Santiago, a married couple who always watched over my Ph.D. A call, a voice of encouragement, a phrase in the hallway, they were always there.

Thanks to Dr. Jorge Velez for his continuous advice and criticism. For his immense knowledge shared with everyone. For the meetings and the willingness for everything.I want to thank his family: Yolima Espinosa, wife and Gabriela Perez Espinosa, daughter. Your friendship is an incalculable treasure.

To my daughter and wife, my family whom I love deeply, they are my motivation to fight in life; I have always wanted to get ahead to someday reward all the sacrifices they have made for me. To my mother and sister, for their unconditional support. I love them immensely.

I have the honor to thank the Universidad del Norte, Barranquilla, Colombia, for giving me a tuition scholarship and the opportunity to work as a professor, in order to support my professional training process. I also thank the Colombian Agency for Research and Development (COLCIENCIAS), contract FP44842-068-2018 for supporting me with a doctoral scholarship at the Universidad del Norte, Barranquilla, Colombia.

I would like to thank all my colleagues in the Ph.D. and M.S. in Engineering programs for sharing pleasant moments and scientific and non-scientific discussions. Special thanks to Darwin Peña for being a friend in arms.

LUIS FERNANDO MACHADO DOMINGUEZ
Universidad del Norte
July 29, 2021

List of Publications

Journals:

[1] **Machado-Domínguez**, L.F., Paternina-Arboleda, C.D., Vélez, J.I. et al. A memetic algorithm to address the multi-node resource-constrained project scheduling problem. *J Sched* (2021). https://doi.org/10.1007/s10951-021-00696-5

[2] **Machado-Domínguez**, L.F., Paternina-Arboleda, C.D., Vélez, J.I. et al. An adaptative bacterial foraging optimization algorithm for solving the MRCPSP with discounted cash flows. *TOP* (2021). https://doi.org/10.1007/s11750-021-00612-2

Conferences:

[1] **Machado-Domínguez**, Luis, Paternina-Arboleda, Carlos, An Adaptative Bacterial Foraging Optimization Algorithm for Solving the MRCPSP with Discounted Cash Flows, in *XIII Metaheuristics International Conference MIC 2019*, Book of abstract, pp. 35-37, 2019.

[2] **Machado-Domínguez**, Luis, Paternina-Arboleda, Carlos, Barrios Agustin, Memetic Algorithm for Solving MRCPSP, in *II Congreso Internacional de Ciencias Básicas e Ingeniería - CICI 2018*, 2018.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Project scheduling has been defined as "*the discipline that plans, organizes and manages resources to successfully carry out the specific goals and objectives of a project*" (Vanhoucke, 2012). This discipline is considered of great importance given the scope of strategic planning in various areas of theoretical and practical knowledge: research and development, construction, operations and maintenance, product development, etc. This chapter presents the components of the problem, the objectives and scope of the research and, finally, describes the contents of each chapter.

1.1 Background to the research

In the literature we find different concepts of what should be understood as a project. According to Erik L. Demeulemeester, 2002 it is defined as "*a single process, consisting of a set of controlled and coordinated activities with start and end dates, carried out to achieve an objective in accordance with specific requirements, including time and resource constraints*". In Larson and Gray, 2015 it is defined as " *an effort to create a unique product, service or result. Due to its temporary nature, it has a defined start and completion time associated with it. Completion of a project is reached when the project objectives have been achieved or cannot be met, or when the need for the project no longer exists*". With these two definitions in mind, a project must meet the following criteria:

1. Start and end time.
2. Specific objectives based on time, quality and cost.
3. Time, resource and physical constraints.
4. Risk.

When defining a project, we must consider a tool that helps to integrate the execution of activities, the resources to be used, and the deadlines in which they must be carried out. A schedule should reflect all possible scenarios for the timely delivery of the process being developed, which is why the project scheduling phase is so important since it provides us with an adequate schedule for its execution.

Project scheduling has had a remarkable development in recent years due to the use of technological tools and techniques based on heuristic and metaheuristic optimization, with the objective of modeling new practical situations and executing more efficient algorithms. In project scheduling there are many factors that produce unfeasible schedules, among them uncertainty. According to Deblaere, Demeulemeester, and Herroelen, 2011b, uncertainty in project scheduling can take many forms, for example: differences between the start dates of planned and actual activities, their respective costs, resources not available on the agreed date, identification of new activities, etc. This type of inconvenience in project implementation can lead to an adjustment of the initial schedule, delays, very high costs and possible cancellation of the project. Several companies annually show their project management statistics, warning about the high percentage of failures, high costs and non-compliance in the three key objectives in project management (time, budget and quality).

Project managers are obliged to look for the best strategies to find optimal solutions, whether or not there are uncertain situations that cause project disruptions. In resource-constrained project scheduling, solutions are explored that minimize project execution time, reduce costs and maintain cash flows to carry out all project activities. The characteristics of the resource-constrained project scheduling problem, considering the uncertainty, suggest using solution techniques when there are interruptions in the schedule

Researchers in project scheduling consider it necessary to improve modeling techniques and related solutions by adjusting procedures to a more realistic environment. Heuristic and metaheuristic procedures have become a very attractive class of methods for researchers, linked to increased computational power. The flexibility and tunability of these methods provide the ability to implement reactive strategies when there are uncertainty events, with a relatively fast response time. Procedures based on heuristic and metaheuristic methods for scheduling resource-constrained projects are detailed in several papers, for example: Weglarz et al., 2011, Hartmann and Briskorn, 2010, Erik L. Demeulemeester, 2002, Ratajczak-Ropel and Skakovski, 2018.

Solution methods do not usually contemplate the answer to the questions posed by interruptions. Therefore, it is pertinent to reoptimize the baseline schedule when uncertain events occur and thus reduce the impact of interruptions on project execution. These strategies are found in the reactive scheduling approach.

This thesis will consider the problem of scheduling projects with limited resources in which activities can be performed in different ways and there are uncertain events that produce interruptions, changes in activities and in the availability of resources, delaying the completion of the project or, in the worst case, forcing its cancellation.

The research proposal seeks to answer the question: ¿How, through various metaheuristic strategies and reactive scheduling, can the problem of multi-mode project scheduling with limited resources and different types of interruptions be solved in a way that optimizes multiple performance measures based on time, quality and cost?

1.2 Statement of the research problem

Project scheduling problems (PSP) refer to a wide variety of problems classified by three domains: activities, resources and various performance measures Herroelen, Demeulemeester, and De Reyck, 1999. Activity scheduling may require the use of different types of limited resources to meet scheduling objectives and thus obtain the three fundamental factors in managing a project (time, cost, and quality), Erik L. Demeulemeester, 2002.

Among the project scheduling problems is the multi-mode resource constrained project scheduling problem (MRCPSP), which consists of carrying out a project, in which the activities are subject to meet precedence, renewable resource and non-renewable resource constraints. In addition to the above constraints, each activity can be sequenced in different ways or modes, using different amounts of resources and varying their duration.

The MRCPSP is denoted by $m, 1T, va|cpm, \delta_j, disc, mu,^{\circ}|reg$, according to the classification of Herroelen, Demeulemeester, and De Reyck, 1999. In order to consider regular and non-regular measurements, we will classify it by $u, 1T, va|cpm, \delta_j, disc, mu,^{\circ}|\gamma$ and thus, we will aim to: minimize the project duration ($MRCPSP - C_{\max}$), and maximize the net present value ($MRCPSPDC$). The value u provides the number of resources; $1T$ provides the availability of renewable and non-renewable resources, of which both the and the total base of the project horizon are specified; va renewable resources are available in varying quantities; of which the time duration of each is specified and a total project horizon basis; cpm regulates the final-start precedence relationships without delays as used in the basic PERT/CPM model; δ_j time periods imposed on activities; $disc$ corresponds to the resource requirements of the activities and are a discrete function of the duration of the activity; μ activities have multiple prespecified execution modes; $^{\circ}$ there is no specification of the cash flow and γ corresponds to the regular and non-regular objective functions.

Initially, the approach with which the MRCPSP is considered is a deterministic approach, meaning that the input parameters to define the problem will invariably produce the same results, not contemplating the existence of chance or uncertainty. This approach is used as a starting point to provide an initial baseline schedule. However, it is possible that at some point there may be variations in the availability of resources or in the execution of its activities, producing delays in the delivery of the project. According to Deblaere, Demeulemeester, and Herroelen, 2011b, the project should be based on a reactive procedure so that the schedule can be reviewed and adjusted to new information and restore its viability.

Next, we will detail the deterministic model of our research on the MRCPSP under two objective functions: minimize the project duration and maximize the net present value. Then, we will define the reactive scheduling approach when uncertainty events occur in the MRCPSP.

1.2.1 MRCPSP

A project consists of a set of real activities $V = \{1, \ldots, n\}$ each of which is carried out without interruption. The dummy activities 0 and $n+1$ are introduced to represent the start and end of the project, respectively. The set of renewable resources is denoted by R^{τ} and will be used to denote the availability (units) of the type of renewable resource $k \in R^{\tau}$. Renewable resources, available period to period, are those that can be restored at a similar or superior speed than the consumption speed. Non-renewable resources are limited to the finished duration of the project without restrictions on any time period. An example of non-renewable resource is the entire budget for the achievement of a project. The set of non-renewable resource is denoted by R^{η}. The availability of a non-renewable resource type $\ell \in R^{\eta}$ is denoted by R^{η}_{ℓ}.

To complete the project satisfactorily, it is necessary to process (execute, sequence) each activity in one of several modes; the set of modes for an activity $i \in V$ is denoted by $M_i = \{1, 2, \ldots, |M_i|\}$, where every mode $m \in M_i$ represents a different way of finishing the i-th activity and $|M_i|$ represents the total number of modes. A mode $m \in M_i$ determines the duration $d_{i,m} \geq 0$ of the activity $i \in V$, measured in number of periods or units of time, which indicates the time necessary to complete the activity, and whether there exists possibility of interrupting the process of that activity. For dummy activities, $d_{0,1} = d_{(n+1),1} = 0$. If the execution mode of activity $i \in V$ is $m \in M_i$, $r^{\tau}_{i,m,k} \geq 0$, where $r^{\tau}_{i,m,k} \leq R^{\tau}_{k}$ and $r^{\tau}_{0,1,k} = r^{\tau}_{n+1,1,k} = 0$ are the units of the renewable resource k required by activity i for its realization, while $r^{\eta}_{i,m,\ell} \geq 0$ are the required units of the non-renewable resource l with $r^{\eta}_{i,m,\ell} \leq R^{\eta}_{\ell}$ and $r^{\eta}_{0,1,\ell} = r^{\eta}_{(n+1),1,\ell} = 0$. It is assumed that there is an end-start relationship with no lead times for the precedence relationships and a deadline time for the project, δ_n for the MRCPSP maximizing the net present value.

Minimize project duration

The mathematical formulations for the MRCPSP most widely used in the literature are presented by Pritsker, Waiters, and Wolfe, 1969; Talbot, 1982; Valdes and Goerlich, 1993; Mingozzi et al., 1998a. Most of the existing models for MRCPSP are adapted versions of Talbot (1982) which introduced a notation of 0-1 programming model with a binary decision variable $x_{imt} = 1$ if the i-th activity is performed in m mode and starts at time t, and $x_{imt} = 0$ otherwise. We propose to use the following mathematical model. Christofides, Alvarez-Valdes, and Tamarit, 1987; Neumann, Schwindt, and Zimmermann, 2012 present a formulation based on the start times and resource consumption of the activities, being a practical and simple model.

$$\begin{aligned}
\text{min} \quad & s_{n+1} && (1.1)\\
\text{subject to} \quad & s_i + d_{i,m_i} \leq s_j, \quad \forall (i,j) \in E, && (1.2)\\
& \sum_{i \in \mathcal{A}(S,\mu,t)} r^{\tau}_{i,m_i,k} \leq R^{\tau}_k, \quad k \in R^{\tau}, 0 \leq t \leq \bar{d}, && (1.3)\\
& \sum_{i=1}^{n} r^{\eta}_{i,m_i,\ell} \leq R^{\eta}_{\ell}, \qquad \ell \in R^{\eta}, && (1.4)\\
& m_i \in M_i, \qquad \forall i \in V && (1.5)\\
& s_i \geq 0, \qquad \forall i \in V && (1.6)\\
& s_0 = 0 && (1.7)
\end{aligned}$$

In (1.3), $\mathcal{A}(S,\mu,t)$, also called the active set, is the set of real activities which will be sequenced in time t, $\mu = (m_i)_{i \in V}$ is the vector mode, $S = (s_i)_{i \in V}$ the vector start times of each activity and $\bar{d} = \sum_{j \in V} \max_{m \in M_j} d_{j,m}$ the maximum duration of the project. The objective function (1.1) minimizes the duration of the project. The constraints represented by equation (1.2) describe the precedence relationships between activities. Equation (1.3) ensures that the availability of renewable resources is not exceeded in each period, while equation (1.4) ensures that non-renewable resources throughout the project. The restrictions (1.5) and (1.6) ensure that each activity is assigned a single mode and a single start time during the execution of the schedule. It is assumed that dummy activities start and end are only executed in a single zero-duration mode and do not consume resources (1.7).

A vector mode μ is a $n+2-$tuple $\mu = (1, \mu_1, \ldots, \mu_n, 1)$ which assigns to the j-th activity, $1 \leq j \leq n$, a unique mode μ_j, $1 \leq \mu_j \leq |M_j|$. A mode vector μ such that the constraints (1.4) are satisfied is called resource feasible. Otherwise, it will be said resource non-feasible. We define the excess of non-renewable resource for a vector mode μ by

$$L^{\eta}(\mu) = \sum_{k \in\, R^{\eta}} |\min\{0, R^{\eta}_k - \sum_{j=1}^{n} r^{\eta}_{j,\mu(j),k}\}| \tag{1.8}$$

being μ is a feasible resource vector mode if and only if $L^{\eta}(\mu) = 0$.

Maximize net present value

A cash outflow $CF^{-}_{j,m}$ is associated with the performance of activity j in mode m and a profit margin γ_i. The contractor receives a cash flow amount $CF^{+}_{m,j} = CF^{-}_{m,j}\dot{(}1+\gamma)$ for each activity j in mode m that has been successfully completed. To calculate the value of net present value (Npv), the discount rate α (also called the rate of return or the opportunity cost of capital) is selected, which represents the return that is derived from investing in the project, for example, in insurance. Then the discount factor $\beta = (1+\alpha)^{-1}$ denotes the present value of one dollar to be received at the end of period 1 using a discount value α. The discount factor β is then replaced by $e^{-\alpha}$. The rule is then based on accepting the project when $Npv > 0$ and rejecting it when $Npv < 0$. The MRCPSP with objective function Npv or simply MRCPSPDC can be formulated as a nonlinear programming problem for which we will only consider the payment model: payment as a function of activity completion times (PAC).

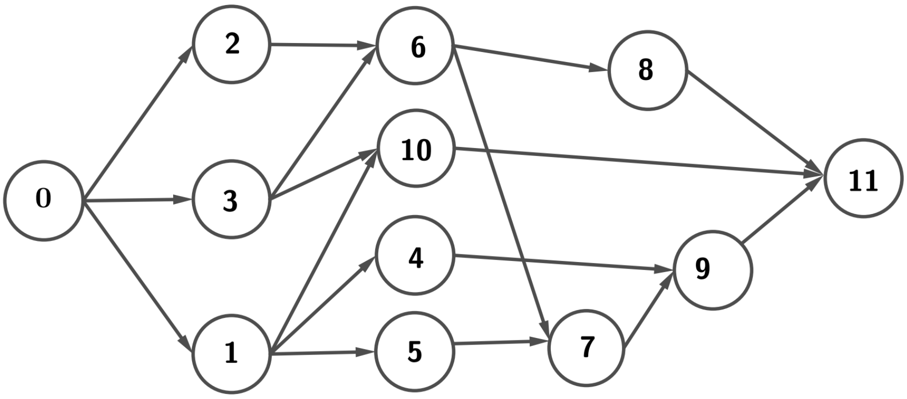
Then the model for the MRCPSPDC with PAC payment model is given by:

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{i=1}^{n}(c_{i,in} + c_{i,out}) \cdot e^{-\alpha f_i} && (1.9)\\
\text{subject to} \quad & f_i \leq f_j - d_{j,m_j}, \quad \forall (i,j) \in E, && (1.10)\\
& \sum_{i \in \mathcal{A}(F,M,t)} r^{\tau}_{i,m_i,k} \leq R^{\tau}_k, \quad k \in R^{\tau}, 0 \leq t \leq \delta_{n+1}, && (1.11)\\
& \sum_{i=1}^{n} r^{\eta}_{i,m_i,\ell} \leq R^{\eta}_{\ell}, \qquad \ell \in R^{\eta}, && (1.12)\\
& f_{n+1} \leq \delta_{n+1} && (1.13)\\
& m_i \in M_i, \qquad \forall i \in V && (1.14)\\
& f_i \in int^{+}, \qquad \forall i \in V && (1.15)\\
& && (1.16)
\end{aligned}
$$

In (1.11), $\mathcal{A}(F,M,t)$, also called the active set, is a set of real activities which will be sequenced in time t; $M = (m_i)_{i \in V}$ is the mode vector; and $F = (f_i)_{i \in V}$ is a vector of completion times for each activity. The objective function (1.9) maximizes the project's NPV by discounting the cash inflow and outflow to each activity's finish time. The constraints represented by equation (1.10) describe the precedence relationships between activities; equation (1.11) ensures that the availability of renewable resources is not exceeded in each period; and equation (1.12) ensures that non-renewable resources are available throughout the project. Restrictions in equation (1.13) ensure that the deadline for project delivery is met, and those in equation (1.14) ensure that each activity is assigned only one mode. The restriction in (1.15) states that the decision variables should be integers. It is assumed that dummy activities start and end are only executed in a single zero-duration mode and do not consume resources.

A vector mode μ is a $n+2-$tuple $\mu = (1, \mu_1, \ldots, \mu_n, 1)$, which assigns to the j-th activity, $1 \leq j \leq n$, a unique mode μ_j, $1 \leq \mu_j \leq |M_j|$. Any vector μ satisfying the constraints in (1.12) is called resource feasible. Otherwise, it will be said resource non-feasible.

According to (Kolisch and Drexl, 1997) the MRCPSPDC is NP-Complete as long as at least two non-renewable resources are included, which implies that the MRCPSPDC



Activities	1	2	3	4	5	6	7	8	9	10
$c_{i,in}(c_{i,out})$	227	426	-127	323	-263	344	250	-469	-228	-25

Figure 1.1: Project instance

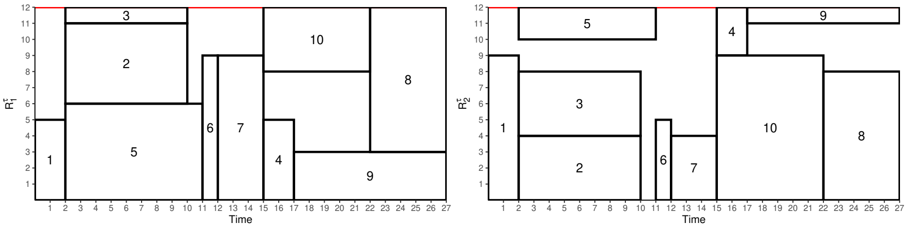


Figure 1.2: Feasible schedule MRCPSP

is also NP-complete.

Numerical example

For illustration purposes, let us consider the instance J1037_2 from the PSPLIB library. This instance contains 10 real and two dummy activities. Each of the real activities has three execution modes. For each mode, two renewable resources and two non-renewable resources are available. The availability of renewable resources is $R_1^\tau = R_2^\tau = 12$ and $R_1^\eta = 37$ $R_2^\eta = 60$ for non-renewable resources. The network of activities in the nodes, and cash flow is presented in Figure 3.2. Table 3.2 shows the durations for each execution mode and the requirements for renewable and non-renewable resources. The mode vector $\mu = (1,1,2,3,3,3,1,3,2,3,3,1)$ is resource feasible as $\sum_{i=0}^{11} r_{i,m_i,1}^{\eta} = 37$ and $\sum_{i=0}^{11} r_{i,m_i,2}^{\eta} = 60$ and $L^\eta(\mu) = 0$. However, $\mu' = (1,1,3,3,3,3,1,3,2,3,3,1)$ has non-feasible resources as $\sum_{i=0}^{11} r_{i,m_i,1}^{\eta} = 38$, $\sum_{i=1}^{12} r_{i,m_i,2}^{\eta} = 63$ and $L^\eta(\mu') = 4$. Figure 3.3 depicts a schedule with a makespan of 27 time periods and net present value of 498 units.

Act i	**Mode** m_i	d_{m_i}	r^{τ}_{1,i,m_i}	r^{τ}_{2,i,m_i}	r^{η}_{1,i,m_i}	r^{η}_{2,i,m_i}
0	1	0	0	0	0	0
1	1	2	5	9	4	7
	2	5	5	8	2	7
	3	6	5	6	1	6
2	1	1	6	5	6	8
	2	8	5	4	5	5
	3	8	4	5	4	6
3	1	1	4	9	9	5
	2	8	3	4	6	1
	3	8	1	4	2	3
4	1	1	7	3	7	8
	2	2	6	3	5	8
	3	2	5	3	6	5
5	1	1	7	9	4	10
	2	4	7	7	2	10
	3	9	6	2	1	10
6	1	1	9	5	3	6
	2	1	9	4	3	7
	3	5	9	3	3	6
7	1	2	9	5	9	8
	2	2	9	6	7	8
	3	3	9	4	4	8
8	1	1	9	9	10	6
	2	5	9	8	6	6
	3	9	9	8	4	5
9	1	2	7	2	4	9
	2	9	7	2	4	6
	3	10	3	1	3	3
10	1	3	5	10	8	10
	2	6	4	10	5	7
	3	7	4	9	3	7
11	1	0	0	0	0	0
	R^{τ}_1	R^{τ}_2	R^{η}_1	R^{η}_1		
	12	12	37	60		

Table 1.1: Project information

1.2.2 Reactive scheduling

During the last few years, research efforts have concentrated on the development of exact, heuristic and metaheuristic procedures for generating an initial schedule assuming complete information in a deterministic environment. However, during execution, projects can be subject to considerable uncertainty, which can result in numerous schedule disruptions.

Reactive scheduling refers to the modification that will be made to a program during the execution of a project (Herroelen and Leus, 2004), to update the program and provide an immediate response to an unexpected event. Reactive scheduling is sometimes referred to as predictive-reactive scheduling (Herroelen and Leus, 2004). Under reactive scheduling, it has two main stages (Vieira, Herrmann, and Lin, 2003): generating an initial schedule and then updating that schedule in response to the outage or other event to minimize its impact on system performance. The initial schedule will be subject to achieving the objective of minimizing project duration or maximizing net present value.

There are several levels and types of disruption, according to Miller and Lessard, 2001. In the case of the levels, we have the natural case, the market, the fiscal scenario, the industry, and the technical project. In our work, we will study only the technical project level since the impact on the project is minimal and its probability of occurrence is high. On the other hand, to distinguish the types of disruption associated with this level, we will have only activity disruptions and renewable resource disruptions. The activity interruption is due to events in which the duration of the activity is prolonged for a not very long period of time. In the case of renewable resource interruption, it is due to a change in the availability of the resource period by period, affecting activities that cannot dispose of the resource. This thesis will consider the problem of scheduling projects with limited resources in which activities can be performed in different ways and there are uncertain events that produce interruptions, changes in activities and in the availability of resources, delaying the completion of the project or, in the worst case, forcing its cancellation.

Reactive programming can be based on several underlying strategies. On the one hand, the reactive program can be based on very simple techniques aimed at a consistent and rapid restoration of the program. On the other hand, the reactive scheduling approach may involve a complete scheduling step of the part of the project that remains to be executed at the time the reaction starts. According to Herroelen and Leus, 2004, such an approach is called (full) reprogramming. Below are several reactive programming techniques:

- *schedule repair:* The action requested when there is a disruption or interruption of the project is to perform program correction as quickly as possible. Program repair may include simple control rules, such as the well-known right justification rule (Herroelen and Leus, 2004; Vonder et al., 2007a). This rule brings forward in time all the activities that are affected by the breakdown of the schedule since they were being executed at the resource causing the break or due to the interruption of the timing of the activities or the precedence relationships. Vonder et al., 2006 states that this reactive strategy can lead to poor results, since it does not reprogram activities. However, the virtues of these strategies are the simplicity of the techniques used and the speed of scheduling modification.

- *Rescheduling:* One of the most commonly used techniques in reactive scheduling is rescheduling, since it is responsible for updating and scheduling the part of the project that has not yet been executed, which is the part of the project that is considered to have a completion time greater than the time in which the interruption is made. Rescheduling can use any deterministic performance measure, such as new project makespan (in manufacturing systems, this approach is sometimes referred to as (full) regeneration). In a sense, scheduling repair can be viewed as a heuristic rescheduling step.

Many of the research efforts on reactive scheduling focus on stability performance measurement (ex post) that evaluates stability after the occurrence of the disturbance. The goal is to obtain a new schedule that does not deviate significantly from the original one, regardless of whether or not the initial baseline schedule was optimized and whether or not the potential occurrence of disturbances was taken into consideration through a proactive approach. Such a strategy is called "minimal disturbance strategy". Consequently, several stability measures are proposed in the scheduling literature. For example, stability can be measured by a function of the difference between the start time of each activity in the new schedule and in the original one. It can also be expressed by the number of activities to be performed in different resource units. Calhoun et al., 2002 have dealt with the generalized multi-mode resource problem in which a target scheduling model has been formulated. Finally, Deblaere, Demeulemeester, and Herroelen, 2011b have considered different types of outages separately, which are related to activity duration, renewable resources and non-renewable resources. The objective of this study is the minimization of a rescheduling cost that includes a mode failure cost and an activity start delay cost.

- *Contingent scheduling:* Many project managers prefer to make manual changes to the schedule during project execution. To aid in decision making, several algorithms have been proposed in the literature, all of which offer various possibilities for switching from one solution to another. Contingent scheduling was studied, for example, Billaut and Roubellat, 1996a; Billaut and Roubellat, 1996b; Briand, Despontin, and Roubellat, 2002; Artigues, Roubellat, and Billaut, 1999.

- *Activity crashing:* When the project is in progress, corrective strategies may consist of decreasing the duration of activities. The clash may result in a time/cost tradeoff, a time/resource problem, multiple modes or multiple clash mode problems, etc. In fact, according to Ahn and Erenguc, 1998, the duration or cost of an activity can be expressed in terms of the resource requirements and the blocked quantity. In Erik L. Demeulemeester, 2002 provides a review of models related to crashing activities. According to Gerk and Qassim, 2008 the main model used to formulate and solve crashing activity problems is linear programming.

- *Sensitivity analysis:* An interesting area of research is the sensitivity analysis of project schedules. of project schedules. Sensitivity analysis attempts to answer to answer a number of fundamental questions such as:
    1. What are the limits of change of a parameter in the execution of a project so that the optimal or feasible solution is maintained?
    2. Given some changes in the parameters, what is the new optimum cost? What would be the optimal solution to minimize this cost?
    3. Which method generates robust schedules when project interruptions occur?
    4. If we talk about resources, what type of resources have the greatest effect on project cost and time?

1.2.3 Problem definition

The most general problem addressed by this thesis is a reactive scheduling problem for the MRCPSP when uncertain events occur, which produces interruptions in activities and resource availability. The reactive scheduling problem for the MRCPSP when uncertainty events occur is to identify an initial baseline schedule for the project for which renewable and non-renewable resource constraints are not violated. Then, if that schedule is under uncertainty-related disruption factors, the schedule is rescheduled for those activities that have not been executed, taking into account the ability to optimize the cost of rescheduling and the net present value of the project. The fundamental questions to address this problem are: How to generate an initial schedule for the deterministic MRCPSP that optimizes the duration and the net present value, what are the corrective measures in the schedule when interruptions are generated, and what method do we use to build the missing schedule for the activities that were not executed and minimize the cost of rescheduling them. The solution approach proposed in this paper will address these three questions. The solution method to find a feasible schedule that minimizes the project duration will be a memetic algorithm and to maximize the net present value, we propose an algorithm based on adaptive bacterial foraging. It should be noted that in this first part uncertainty is not taken into account and a deterministic problem is defined. However, when starting to execute the chosen schedule, corrective measures such as rescheduling are considered if uncertain events occur that affect the schedule. Finally, to reestablish the schedule we run a multi-objective algorithm based on bacterial foraging to calculate the start times of the activities that were affected by the interruptions. If the schedule found complies with all the constraints we continue with its execution.

1.3 Research Hypothesis

It is possible to perform an integration of the MRCPSP solution methods for the deterministic case and the reactive strategies when a specific type of interruption is established. Additionally, it is possible to reduce the impact of these interruptions on the execution of a project.

This hypothesis can be tested by considering the following:

1. Solve MRCPSP with objective function to minimize project duration
2. Solve the MRCPSP with objective function to minimize the net present value of the project.
3. Generate two types of breaks in the MRCPSP when the initial baseline schedule is executed and propose a solution strategy.
4. To propose an optimization methodology that unifies the deterministic scenario and the events under uncertainty.

1.4 Research objective

1.4.1 General objective

Design and develop an optimization methodology to solve the multi-mode resource-constrained project scheduling problem with different types of interruptions, through various metaheuristics and the implementation of reactive strategies.

1.4.2 Specific objectives

- Develop a memetic algorithm that will provide satisfactory solutions to the MRCPSP subject to minimizing project duration.
- Develop a bacterial foraging algorithm that will provide satisfactory solutions to the MRCPSP subject to maximizing the net present value of the project.
- Design of a multi-objective metaheuristic algorithm to reduce the impact of outages in resource-constrained projects based on the definition of reactive scheduling.
- Integrate metaheuristic procedures into a methodological strategy.
- Evaluate and contrast the developed methods with the best procedures found in the literature, using the standard PSPLIB and MMLIB library.
- Identify the limitations of the proposed model and identify recommendations for further research in the field of reactive scheduling.

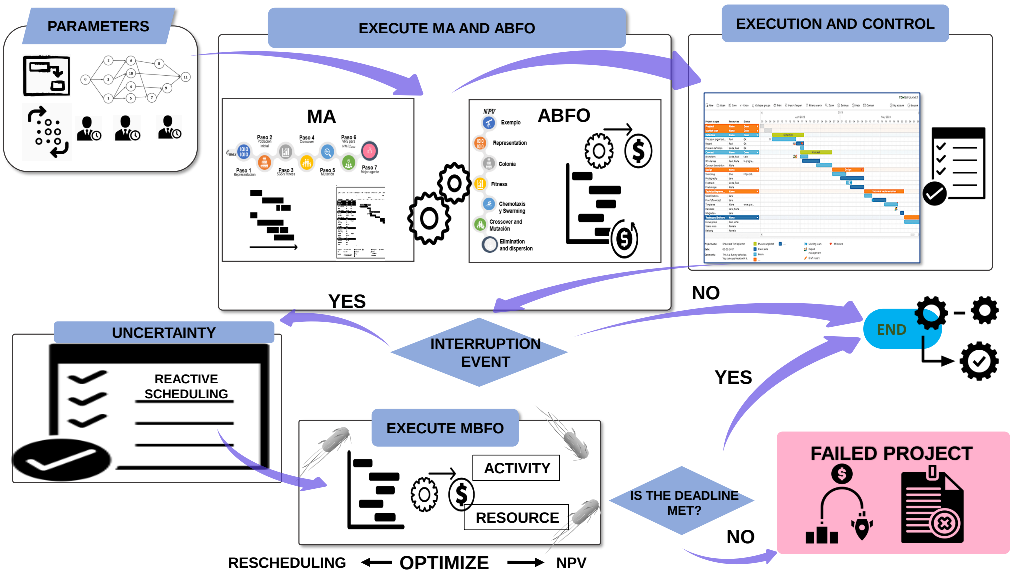


Figure 1.3: Proposed optimization methodology

1.5 Research Methodology

In this research a combination of quantitative methods is used, characterized by 4 sequential stages. The methodological approach is illustrated in Figure 1.3.

Stage 1 . This is the initial stage, which involves establishing the characteristics of the problem, defining parameters, input values, output values, durations, resource consumption, cash flows, precedence relationships and a deadline for project delivery. The definition of these concepts is found in the chapter 2 where a taxonomy and a systematic review of the problem are developed, achieving a general and deep vision of the problem.

Stage 2 . In stage two, two metaheuristic procedures are established to generate two schedules that optimize the project duration and the net present value. The first method is called memetic algorithm to solve the MRCPSP with objective function project duration. This algorithm generates an initial population built by the selection of priority rules for the vector of modes and then for the lists of activities, managing to find feasible solutions of very good quality. Then, versions of the crossover and mutation operators are developed, which are in charge of intensifying and diversifying the search. To these solutions, a variable neighborhood search is applied with justification of the activities by changing the execution mode. The description of the memetic algorithm can be found in more detail in the chapter 3. The second solution method is called adaptive bacterial foraging optimization algorithm that helps to find solutions of the MRCPSP with objective function to maximize the net present value. This algorithm is based on bacterial foraging and mimics the way bacteria forage in a nutrient landscape to perform nonaggressive parallel optimization by sensing chemical gradients in the environment (such as nutrients) and moving toward or away from specific cues. We highlight for generating the list of activities a priority rule based on providing good quality lists of activities with respect to net present value. Among its operators are: chemotaxis, swarming, crossover and mutation. The result is a feasible schedule that maximizes the net present value. The description of this chapter can be found in the chapter 4.

Stage 3 . The two solutions found are shown to the client in order to select the initial baseline schedule by the client and the contractor. After the initial baseline schedule is chosen and execution begins, it must be strictly controlled and monitored. In our system, two types of interruptions generated by uncertainty events are established: interruption of activities and interruption of resources. If, when executing a subset of activities, an uncertainty event occurs, we establish our reactive scheduling strategy.

Stage 4 . The reactive scheduling strategy we use is based on rescheduling, in charge of updating and scheduling the activities that have not been executed and were affected by the interruptions. Then, the project parameters are updated in the respective instance and a multi-objective algorithm is executed to calculate the new times that minimize the cost of rescheduling and the net present value of the project, taking into account the new project values. Otherwise, if there are no interruptions, the project is completed with the delivery of the executed project. Finally, if the schedule provided by the MBFO is feasible according to the delivery date, then the project is completed, otherwise it is marked as a failed project. The description of the MBFO applied to an experiment can be found in chapter 5.

1.6 Research Contribution

There are several contributions that this thesis proposes to the literature of multi-mode resource constrained project scheduling, consisting essentially in the development of a new reactive scheduling strategy and its solution approaches. The specific contributions are summarized below.

Memetic algorithm for solving the MRCPSP

- The implementation of a variable neighborhood search (VNS) adapted to a double justification operator, and a uniform crossing operator;
- The development of an improvement procedure that optimizes the consumption of resources and minimizes the work for the activities or their duration;
- A robust experimental design to find suitable parameters for the MA algorithm.

An adaptative bacterial foraging for solving the MRCPSP with discounted cash flows

- A dual justification operator based on cash flow, based on multi-modal justification.
- A priority rule based on multi-modal behavior.
- Adaptations of the chemotaxis and swarming operators for improved solutions.
- Taguchi experimental design to find the appropriate parameters for the ABFO algorithm.

The Reactive MRCPSP

- Proposed a modern multi-objective solution method to optimize two objectives: minimize rescheduling cost and maximize net present value.
- In order to maximize NPV and minimize the cost of rescheduling, we build reactive scheduling models to generate schedules when interruptions occur during execution.
- The development of two heuristic algorithms when activity and resource interruptions occur. Validation of their effectiveness based on a computational experiment performed with a PSPLIB instance.
- Based on the computational experiment, we find that the best solutions are found when we have 15% and 20% of deadline.

1.7 Research Limitations

This study has a number of limitations. The first limitation refers to the set of instances; the PSPLIB and MMLIB libraries are artificial libraries and it would be pertinent to apply the algorithms to real instances with more than 100 activities, several execution modes, and resources. The second limitation refers to the use of exact solution methods; due to the nature of the problem studied it is pertinent to use metaheuristic strategies, however, it would be appropriate to use exact procedures based on mathematical programming. The third limitation refers to the use of different types of interruptions in the projects, this would give a wider field to reschedule the schedules with different strategies.

1.8 Justification for the research

Reactive scheduling applied to project scheduling helps to build viable schedules when uncertainty events occur. In recent years, more and more attention has been paid to building and developing techniques to counteract uncertainty in projects and to be able to adjust their results. However, the techniques that have been used the most are constructive heuristics and exact, branch and bound procedures. Among the project scheduling problems is the multiple-mode resource-constrained project scheduling problem whose deterministic solution is based on a schedule that meets the resource constraints and precedence relationships. However, when an initial baseline schedule is constructed, it is under great uncertainty for its execution, then it is appropriate to use reactive strategies to adjust the schedule and re-execute it.

In practice, maximizing the net present value and minimizing the cost of rescheduling results in better resource utilization and reduced downtime in a schedule. These factors result in the client providing an extension to the contractor to meet the objectives set out in the new schedule.

There are several approaches, however, the approach we use is based on reactive scheduling, since it is the process that starts running from the beginning of the project until its end. There is no reactive scheduling technique that dominates all the solutions but we will be giving solutions to these questions from our approach.

1.9 Structure

Chapter 1 introduces the research problem. Likewise, the definition of the problem and its mathematical formulation are presented. Then, the objectives and the research methodology are presented with all their characteristics, giving a vision of the document. In order to make known the components of the problem, solution methods, prominent authors, and most relevant studies. From these strategies, we focus on reactive programming in order to find research gaps.

In Chapter 2, a taxonomy and a comprehensive literature review of the state of the art in the multi-mode resource-constrained project scheduling problem is performed, associating various objective functions and taking into account reactive scheduling. In addition, the literature review leads to disaggregate and classify the strategies to deal with uncertainty: reactive scheduling, stochastic scheduling, fuzzy scheduling, and proactive scheduling. This review was carried out in order to classify the existing literature, in reference to the research problem. We sought to innovate in the methodology of resolution and formulation of the problem, based on the research gaps found.

In chapter 3, a memetic algorithm for the solution of the MRCPSP is proposed. The memetic algorithm is composed of a variable neighborhood search that helps to justify the activities based on the execution modes. Then, a uniform crossover operator and a mutation operator are proposed to help diversify and intensify the search. Finally, the results were shown which are very good compared to other metaheuristics.

In chapter 4, a new and modern algorithm based on bacterial foraging is proposed. The net present value of the MRCPSP is optimized, with a payment model: payments at activities' completion times. An adaptation of the swarming and chemotaxis operators improves the results significantly. They were tested in the PSPLIB and MMLIB libraries. Their results are compared with a genetic algorithm, achieving a higher percentage of feasible solutions.

In chapter 5, the problem is approached with reactive programming in mind. A methodology is constructed to solve the MRCPSP when there are interruptions in activities and resources. This methodology consists of a multi-objective bacterial foraging algorithm that minimizes the cost of rescheduling and maximizes the net present value.

In Chapter 6, the observations and conclusions of the research and proposed studies are concluded in the last chapter, followed by future research directions as an extension of this study.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Over the past 40 years, the topic of project scheduling has attracted the attention of a considerable number of researchers. The problem of project scheduling can be defined as the allocation of resources to carry out a set of activities in a pre-established order over time. The competition of activities for the use of resources must be taken into account. It is important to highlight the relevance of meeting a timely and effective schedule, for this reason, the project manager must perform an analysis to manage the costs and resources needed to deliver the project on time. With proper scheduling techniques, some activities and tasks can also be adjusted in case a project is delayed or if there is a change in scope.

Areas of project scheduling research include good resource utilization, completion time estimation, cost management, and excellent project quality. This paper focuses on the investigation of these characteristics. The Figure 2.1 shows the life cycle of any project and the research topics that we will address in our research.

A project is a single process, consisting of a set of controlled and coordinated activities with start and end dates, undertaken to achieve an objective in accordance with specific requirements, including time and resource constraints, Erik L. Demeulemeester, 2002. Projects can have probabilistic or deterministic elements. The first originated with the United States Navy's nuclear missile program for submarines in 1958, and was called PERT (Program Evaluation and Review Technique) Malcolm et al., 1959. The second, CPM (Critical Path Method), arose from the DuPond company's efforts to better manage the construction and repair of its chemical plants between 1956 and 1959, (Kelley Jr and Walker, 1959; Malcolm et al., 1959). However, the mentioned methods do not take into account resource limitations. Due to the need to reduce the costs of a project, it is imperative to optimally manage resources, whose availability is limited. To this end, numerous solution techniques have been proposed, both exact and heuristic and metaheuristic. One of the sources of motivation for research work in this line has been the PSP Project Sequencing Problems with resource restriction (Davis, 1966; Laue, 1968; Petrović, 1968; Balas, 1968; Wiest, 1967; Wiest, 1964). A very important part of the

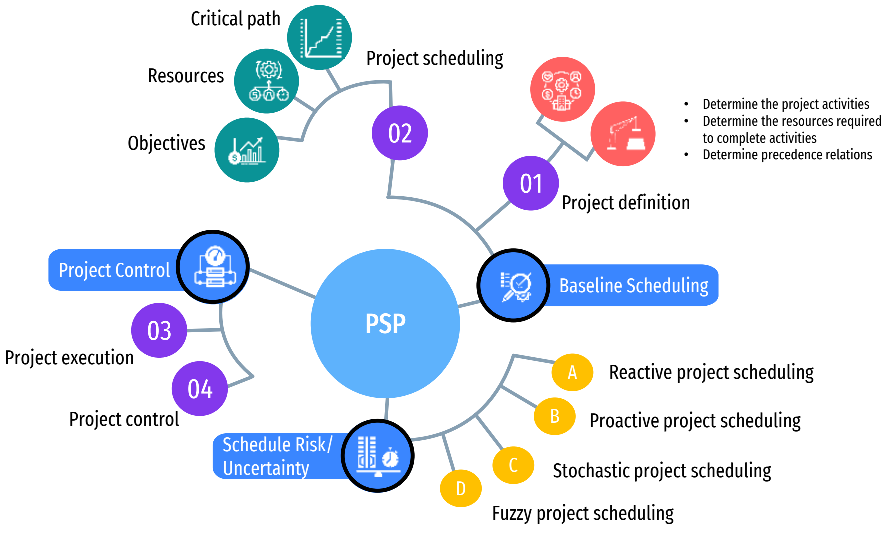

Critical path
Resources
Objectives
Project scheduling
02
Determine the project activities
Determine the resources required to complete activities
Determine precedence relations
01
Project definition
Project Control
03
Project execution
04
Project control
PSP
Baseline Scheduling
A
Reactive project scheduling
B
Proactive project scheduling
C
Stochastic project scheduling
D
Fuzzy project scheduling
Schedule Risk/ Uncertainty

Figure 2.1: The project life cycle and integrating study topics in our research.

projects is the sequencing of the activities, in which, given the objective function, it is necessary to know the best possible way to carry out the activities taking into account the restrictions imposed by the precedence relationships and limited resources.
This chapter presents a study of the scientific literature on the MRCPSP, covering the period from 1982 to 2020 and consisting of 283 documents. The reference for the following work is Eksioglu, Vural, and Reisman, 2009 which offers a taxonomy on the Vehicle Routes Problem (VRP) and Weglarz et al., 2011 which performs a very complete literary review on the MRCPSP.

2.2 Background of revisions

This part describes a methodology to classify the literature that addresses the problem of scheduling projects with limited resources, in a multiple way (MRCPSP). This model encompasses formulations, their components, solution methods, and complexity. This taxonomy is done in order to cover, to a large extent, the different approaches of the MRCPSP. In general, the bibliographic reviews carried out to date summarize previous publications, losing sight of the existing relationships between the different approaches proposed by researchers and the directions of future lines of research.

In Özdamar and Ulusoy, 1995, scheduling problems are classified according to two attributes associated with optimization problems: the objective and the constraints. The classification associated with the goal takes two forms into account: time-based goal (makespan) and time-related goal (mean delay, mean time to finish, and weighted or total delay in a multiproject environment). In sequencing problems you work with one or more objectives. The classification associated with constraints refers to resources and precedence relationships. The resource restrictions are given by three categories:

renewable, non-renewable and doubly restricted. The precedence restrictions are given by the relationships that exist between each of the activities, however, their classification does not provide information on the precedence relationships. In Herroelen, Demeulemeester, and Reyck, 1997 a conceptual description of the classification of the problem of scheduling projects with limited resources and some important extensions is made. His approach is to make known the approaches, procedures and strategies to solve the different problems associated with project scheduling, whether they are relaxed problems or change of objective functions. In the quotation Herroelen, Demeulemeester, and De Reyck, 1999 the need to classify the problems of project sequencing is raised, since it would greatly facilitate the presentation and discussion of the problems of project sequencing. This characterization is based on three fields $\alpha|\beta|\gamma$:

- α field: resource characteristics.
- β field: characteristics of the activities.
- γ field: performance measures.

$$\alpha\begin{cases}\alpha_1\begin{cases}\circ & \text{: No resource types in the scheduling problem}\\ 1 & \text{: Only one type of resource}\\ m & \text{: The number of resource types is } m\end{cases}\\ \alpha_2\begin{cases}\circ & \text{: No resource type specifications}\\ 1 & \text{: Renewable resources}\\ T & \text{: Non-renewable resources}\\ 1T & \text{: Renewable and non-renewable resources}\\ v & \text{: Partially (non)renewable resources}\end{cases}\\ \alpha_3\begin{cases}\circ & \text{: (partially) renewable resources available in constant quantity.}\\ va & \text{: (partially) renewable resources available in variable quantity.}\end{cases}\end{cases}$$

$$\beta \begin{cases} \beta_1 \begin{cases} \circ & : \text{It is not allowed to split the activity} \\ pmtn & : \text{It is allowed to interrupt the activity and resume it where it was interrupted} \\ pmtn - rep & : \text{Activity can be interrupted and resumed from the beginning of the activity} \end{cases} \\ \beta_2 \begin{cases} \circ & : \text{No precedence relationships} \\ cpm & : \text{Final-start precedence relationships} \\ min & : \text{Precedence relationships with minimum time delays} \\ gpr & : \text{Generalized precedence relationships with minimum and maximum time delays} \\ prob & : \text{Probabilistic type project network} \end{cases} \\ \beta_3 \begin{cases} \circ & : \text{Waiting times equal to zero} \\ \rho_j & : \text{Different waiting times per activity} \end{cases} \\ \beta_4 \begin{cases} \circ & : \text{The activities have entire durations} \\ cont & : \text{The activities have continuous durations} \\ d_j = d & : \text{All activities are of equal duration} \end{cases} \\ \beta_5 \begin{cases} \circ & : \text{No delivery dates.} \\ \delta_j & : \text{Deadlines for activities} \\ \delta_n & : \text{Delivery date for the project.} \end{cases} \\ \beta_6 \begin{cases} \circ & : \text{Activities require resources in a constant discrete amount.} \\ vr & : \text{Activities require resources in a discrete variable amount.} \\ disc & : \text{Resource requirements of activities are a discrete function of activity duration} \\ cont & : \text{Activity resource requirements are a continuous function of the duration of} \\ & \text{the activity.} \\ int & : \text{The resource requirements of the activities are expressed as a function of intensity} \\ & \text{or speed.} \end{cases} \end{cases}$$

$$\beta \begin{cases} \beta_7 \begin{cases} \circ & : \text{Activities can be carried out in only one execution mode.} \\ \mu & : \text{Activities can be performed in several pre-specified modes of execution.} \\ id & : \text{Activities are subject to mode identity restrictions.} \end{cases} \\ \beta_8 \begin{cases} \circ & : \text{No cash flows} \\ c_j & : \text{Activities associated with cash flows} \\ c_j^+ & : \text{Activities associated with positive cash flows} \\ per & : \text{Periodic cash flows for the project} \\ sched & : \text{Determine the amount and timing of cash flows.} \end{cases} \\ \beta_9 \begin{cases} \circ & : \text{No changeover times (transport),} \\ s_{jk} & : \text{Sequence-dependent changeover times.} \end{cases} \end{cases}$$

$$\gamma \begin{cases} reg & \text{Performance measure is any measure of early (regular) completion} \\ nonreg & \text{Performance measure is any measure of free (non-regular) completion.} \end{cases}$$

For this reason, in Reyck and Herroelen, 1999 performs the classification based on the MRCPSP with generalized precedence relations, without leaving the based on the MRCPSP with generalized precedence relationships, without neglecting the the ranking of the most important PSPs. In the table 2.1, shows some of the most important PSPs found in the literature (Reyck and Herroelen, 1999).

	Single mode		Multiple mode		
	No resource types	Multiple renewable resource types	1 non-renewable resource type	1 renewable resource type	Multiple renewable and non-renewable resource types
	no trade-offs	no trade-offs	time/cost trade-offs	time/resource trade-offs	time/cost trade-offs, time/resource trade-offs, resource/resource trade-offs
ZERO-LAG FS	CPM/PERT $cpm\|C_{\max}$	RCPSP $m,1\|cpm\|C_{\max}$	DTCTP $1,T\|cpm,disc,mu\|C_{\max}$	DTRTP $1,1\|cpm,disc,mu\|C_{\max}$	MRCPSP $m,1T\|cpm,disc,mu\|C_{\max}$
MIN SS, SF, FS,FF	PDM $min\|C_{\max}$	GRCPSP $m,1,va\|min,\rho_i,\delta_i\|C_{\max}$	GDTCTP $1,T\|min,\rho_i,disc,mu\|C_{\max}$	GDTRTP $1,1\|min,\rho_i,disc,mu\|C_{\max}$	GMRCPSP $m,1T\|min,\rho_i,disc,mu\|C_{\max}$
MIN+MAX SS, SF,FS,FF	MPM $gpr\|C_{\max}$	RCSP-GPR $m,1,va\|gpr,\rho_i,\delta_i,vr\|C_{\max}$	DTCTP-GPR $1,T\|gpr,\rho_i,\delta_i,disc,mu\|C_{\max}$	DTRTP-GPR $1,1,va\|gpr,\rho_i,\delta_i,disc,mu\|C_{\max}$	MRCPSP-GPR $m,1T,va\|gpr,\rho_i,disc,\mu\|C_{\max}$

Table 2.1: A classification of project scheduling problems

In Brucker et al., 1999, the objective is to provide a new classification scheme to describe the resource environment, the characteristics of the activities and the objective function, managing to bridge the gap between project sequencing problems and machine scheduling problems. It expresses the intention to place the single mode case and the multiple mode case at the same level, at the next level are the solution methods: branch and bound methods, lower bounds, heuristic methods, for the multiple mode case the dominance rules are denoted. However Herroelen, Demeulemeester, and Reyck, 2001 makes a criticism of the classification proposed by Brucker et al., 1999, describing the drawbacks in its classification and comparing it with the classification proposed in Herroelen, Demeulemeester, and De Reyck, 1999. The classification addressed in Hartmann and Kolisch, 2000 is based on heuristic algorithms to solve the RCPSP. Therefore, this bibliographic review is part of the research, largely denoting the advances, both for heuristic algorithms, priority rules and metaheuristic algorithms. Since the MRCPSP is a generalization of the RCPSP, most of the solution methods can be adapted and configured to solve the MRCPSP. This classification of solution methods will be used to carry out the MRCPSP taxonomy. Six years later, Kolisch and Hartmann, 2006 the same authors, again carry out a review of the advances in solution methods for the RCPSP. Most notable in these two investigations is the comparison of the computational results of the RCPSP solution methods. In 2007, Lancaster and Ozbayrak, 2007 expanded on the work of Kolisch and Hartmann, 2006, using the classification framework of Herroelen, Demeulemeester, and De Reyck, 1999.

The documents that classify the problem have made very important contributions, since their approach is directed to the formulation and variants of sequencing problems. Most of the documents carry out a modeling process and algorithmic contributions to solve the problem, considering various types of objectives. In 2001, the work carried out by Kolisch and Padman, 2001 examines the literature integrating exact and heuristic models, data types and algorithms, which leads to a contribution in the classification of the problem. In the proposed classification, in the first level we can find the elements of the PSP, the objectives, the graphic representations of the problem and the solution methods. For the solution methods node we find two branches: the exact methods and the heuristics, without taking into account that the metaheuristic methods were going to create their own branch of research due to their considerable increase in research work and contributions. The data considered in this document are only deterministic. In Herroelen and Leus, 2005 a very important classification based on stochastic data is performed. Reviews PSP approaches under uncertainty, such as: reactive scheduling, stochastic project scheduling, fuzzy project scheduling, robust (proactive) scheduling, and sensitivity analysis. Describe for each of the proposed approaches, solution methods given in the literature. In 2010, Hartmann and Briskorn, 2010 provided some variants and extensions of the RCPSP. The classification addressed in Hartmann and Kolisch, 2000 is based on heuristic algorithms to solve the RCPSP. Therefore, this bibliographic review is part of the research, largely denoting the advances, both for heuristic algorithms, as well as for priority rules and metaheuristic algorithms. Since the MRCPSP is a generalization of the RCPSP, most of the solution methods can be adapted and configured to solve the MRCPSP. This classification of solution methods will be used to carry out the MRCPSP taxonomy. Six years later, Kolisch and Hartmann, 2006 the same authors, again carry out a review of the advances in solution methods for the RCPSP. Most notable in these two investigations is the comparison of the computational results of the RCPSP solution methods. In 2007, Lancaster and Ozbayrak, 2007 expanded on the work of Kolisch and Hartmann, 2006, using the classification framework of Herroelen, Demeulemeester, and De Reyck, 1999.

2.3 MRCPSP Taxonomy

A taxonomy is a process based on the principles, methods, and purposes of classification. In general, taxonomies are used to help understand the relationships between all the domains in an investigation. In addition, taxonomies represent hierarchical structures of the terms that represent the domains, indicating the subsets of terms and the relationship of these subsets. The domains in research are many, among them, we have scheduling problems, solution methods, formulation, etc.

2.3.1 Epistemology of MRCPSP literature

We initially searched articles, books, theses, etc., using various databases such as Scopus, IEEE, JSTOR, Web of Science, CiteSeer, EBSCO and ELSEVIER. Search engines such as Google Scholar, SIBILA, Springer and the list of academic databases and reference list provided by Weglarz et al., 2011 were used to find and access scientific research papers about the MRCPSP. Once databases were identified, specific searches by keywords, authors, title and abstract related to "Project scheduling resource constraints", "RCPSP", "MRCPSP", "heuristic AND MRCPSP", "metaheuristic AND MRCPSP", etc. were conducted. Finally, we organize this information by author, title, keywords, summary and other options. Although various specialized software such as JabRef, Zotero, Mendeley, Endnote and Reference Manager are available for this purpose Luna et al., 2014, JabRef was chosen as it provides an agile, complete interface and is the standard LaTeX bibliography format Kopka, Daly, and Rahtz, 2004.

A total of 321 documents were reviewed, including articles, books, book chapters, technical reports and conferences; approximately 82% of documents corresponds to articles, 8% to books, 4% corresponds to Proceedings, 3% corresponds to Collection, 2% to PhD Theses and $<$ 1% to Technical Reports (Figure 2.2a).

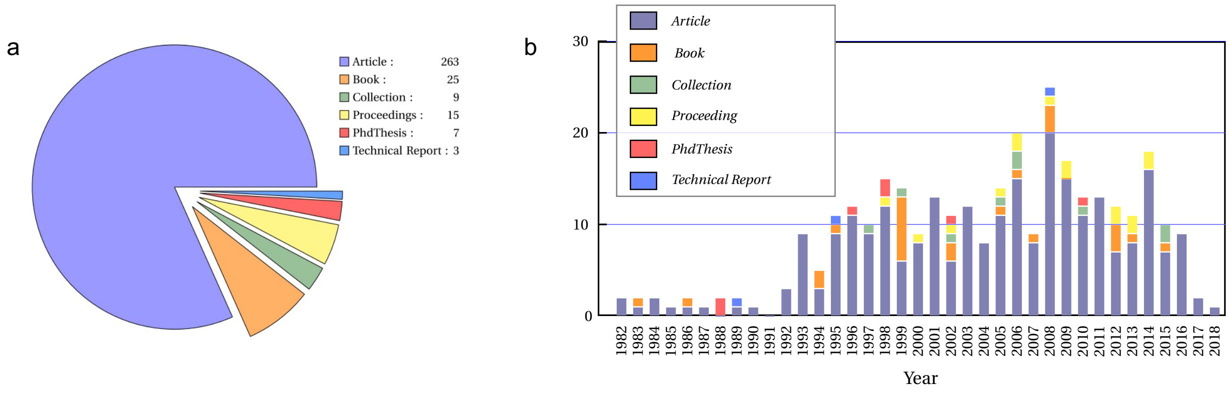


Figure 2.2: (a) Percentages of informational material; (b) number of publications per year.

Figure 2.2b shows the number of research papers per year. In the last two decades, with the advancement of technology and computing capacity of computers, research conducted around the MRCPSP has dramatically increased mainly because it has been possible to develop more efficient algorithms and heuristics for increasingly larger problems. During the 90's, the focus was on making a formal classification of the problem and solving the problems by means of exact algorithms. The advances in recent years have led to the use of various techniques, such as metaheuristic algorithms, with outstanding results. In addition, due to the need of companies and the industry to carry

Title of the journal	Quantity
European Journal of Operational Research	72
Computers & Operations Research	13
Annals of Operations Research	10
Computers & Chemical Engineering	10
Journal of Scheduling	9
Journal of the Operational Research Society	9
Management Science	9
IIE transactions	6
Naval Research Logistics (NRL)	6
International Journal of Production Research	5
Computers & Industrial Engineering	4
Omega	4
OR Spectrum	4
OR Spektrum	4
Applied Mathematics and Computation	3
Information Sciences	3
Operational Research	3
Advances in project scheduling	2
Other Journals	80
Total	256

Table 2.2: List of articles regarding academic journals covering the MRCPSP. The "Other Journals" class groups journals in which only one research paper on the subject.

out processes more quickly, efficiently and reliably, researchers have focused their efforts on the study of the sequencing of projects with resource constraints.

Out of the 321 documents reviewed, we identified 256 research articles that were published between 1982 and 2018 in 83 journals (Table 2.2) . The main source of information is the European Journal of Operational Research with a 28% of the total of articles, followed by Computers & Operations Research with a 5%. There are several journals that usually contain only one article and make up for 31% of research papers.

2.3.2 Need for a taxonomy for MRCPSP

There are several documents that classify the MRCPSP focusing on 3 areas: resources, activities, and objective function; others classify the exact or heuristic solution methods Özdamar and Ulusoy, 1995; Herroelen, Demeulemeester, and Reyck, 1997; Herroelen, Demeulemeester, and De Reyck, 1999; Brucker et al., 1999; Hartmann and Kolisch, 2000; Kolisch and Padman, 2001; Herroelen and Leus, 2005; Kolisch and Hartmann, 2006; Lancaster and Ozbayrak, 2007; Hartmann and Briskorn, 2010, and Weglarz et al., 2011. In this document, we have made a classification by grouping various dimensions and concepts.

The advantages of creating a taxonomy for the MRCPSP will be taken from Bailey, 1994 and adapt to the object of study.

1. **Description**. Classifying a problem is the main descriptive tool. A good classification allows the researcher to provide a comprehensive and perhaps even definitive list of classes and subclasses of the problem. It would facilitate the presentation and discussion of project sequencing problems. For example, the taxonomy presented in Tables 2.4 and 2.5 shows that the object of study of an article can

be classified as research in the dimension "*project components*" to non-renewable resources, in another dimension "*metaheuristic solution*" based on population (genetic algorithm).

2. **Reduction of complexity**. One of the most important reasons for making a taxonomy is to reduce the degree of complexity of a problem by classifying each of the underlying dimensions associated with the problem such as project components, formulation, solution methods, and algorithmic complexity. The taxonomy presented allowed us to condense the information to classify the topic addressed in the articles.

3. **Identification of similarities**. The taxonomy presented allows us to recognize similarities in the research conducted around the MRCPSP. For example, a researcher makes a contribution based on metaheuristic solution methods, a particular case of "genetic algorithm". Subsequent work based on variations of the metaheuristic algorithm can be considered in the same case and thus observe the contributions of this method.

4. **Identification of differences**. Similarly, classification procedures allow us to differentiate cases so that dissimilar cases can be separated for analysis, rather than remaining mixed. For example, using different solution approaches, Figure 2.3:

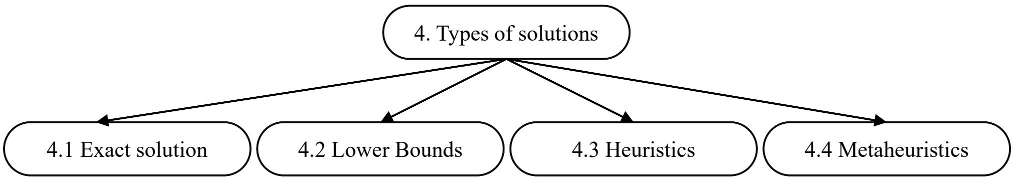


Figure 2.3: Solution methods.

5. **Relationship study**. Although the taxonomy presented is purely descriptive, it provides the relevant relationships for the classification of each of the documents and reveals their contributions in a more meaningful way. However, there may be subclasses that are empty or with few cases; this is important because it indicates that there is no relationship between some of the variables used.

6. **Versatility**.The taxonomy presented above is perhaps a very appropriate tool to show the concepts and joint classifications of the documents associated with the MRCPSP.

7. **Solution methods**. A taxonomy facilitates the adaptation of solution procedures to the problem setting and, as such, facilitates the preparation of literature reviews.

2.3.3 MRCPSP Taxonomy

In the proposed taxonomy, level 1 comprises the classes or categories “Project components”, “mathematical formulation”, “complexity” and “types of solution”. The ramification for each of the classes and sub-levels depend on the area of study and the contribution given to the development of the theory related to the MRCPSP. An important characteristic that must be highlighted is that classes and subclasses that make up the different levels may not be exclusive, that is, a study document can enter into the four existing categories. For instance, Sprecher Sprecher and Drexl, 1998 considers the class *project components*: renewable resources and non-renewable resources, but also in the class *types of solution* uses an algorithm *B&B* (exact method) with some boundary rules (heuristic methods). Table 2.3 shows the articles that helped to carry out the taxonomy for each of the first level classes and their contribution. There may be empty cells, since articles do not address the subject or do not imply the attributes of that cell.

Regarding the Table 2.3, a heat map with dendrogram was used, which organizes the data into subcategories that are divided into others until the desired level of detail is reached. This graphic representation, shown in Figure 2.4, allows us to clearly appreciate the grouping relationships between the four categories and even between the articles and their authors. Observing the successive subdivisions we can get an idea about their grouping criteria.

The articles are ordered from top to bottom depending on the relationship and the fulfillment of the categories, for example the articles Lawler et al., 1993; Zapata, Hodge, and Reklaitis, 2008; Waligóra, 2014 they make contributions simultaneously to categories [1] and [2]. Now if we consider an author, for example Herroelen W. has made contributions for the components of the project Herroelen and Leus, 2004; Herroelen, Demeulemeester, and De Reyck, 1999 and in the solution methods Herroelen, Demeulemeester, and Reyck, 1997; Herroelen, 2005; Herroelen and Leus, 2005. Two groups formed by categories are observed, the first group consists of categories [1] and [4], and the second group consists of categories [3] and [2]. This behavior is due to the fact that, once the formulation and certain aspects related to complexity have been carried out, researchers have concentrated their efforts on describing solution methods and the different versions of the problem associated with project scheduling.

2.3.4 MRCPSP taxonomic classes

Tables 2.4 and 2.5 show the epistemological taxonomy as a result of the bibliographic reviews found in a group of articles that describe approaches and procedures on the MRCPSP. The classification is proposed as a knowledge model to clearly distinguish the topics and subtopics addressed in different bibliographic reviews. In addition, it suggests a block review of each document, identifying its components. For example, the *Resource* attribute are classified by category, types and numbers.

Article name	Author	Year	[1]	[2]	[3]	[4]
Scheduling subject to resource constraints: classification and complexity	Blazewicz, Lenstra, and Kan, 1983	1983	✓	✓	✓	
A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem	Patterson, 1984	1984				✓
Project scheduling problems: a survey	Icmeli, Selcuk Erenguc, and Zappe, 1993	1993	✓			
Sequencing and scheduling: Algorithms and complexity	Lawler et al., 1993	1993	✓	✓		
Resource-Constrained Project Scheduling: Exact Methods for the Multi-Mode Case	Sprecher, 1994	1994	✓	✓	✓	✓
A survey on the resource-constrained project scheduling problem	Özdamar and Ulusoy, 1995	1995	✓			✓
Project Scheduling under Resource Constraints-Efficient Heuristics for Several Problem Classes	Kolisch, 2013	1995	✓	✓	✓	✓
A genetic algorithm for resource-constrained scheduling.	M., 1996	1996	✓			✓
Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation	Kolisch, 1996b	1996				✓
Resource-constrained project scheduling: A survey of recent developments	Herroelen, Demeulemeester, and Reyck, 1997	1997	✓			✓
Project scheduling with multiple modes: A comparison of exact algorithms	Hartmann and Drexl, 1998	1998				✓
Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm	Sprecher and Drexl, 1998	1998				✓
An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation	Mingozzi et al., 1998b	1998				✓
A classification scheme for project scheduling	Herroelen, Demeulemeester, and De Reyck, 1999	1999	✓			
Resource-constrained project scheduling: Notation, classification, models, and methods	Brucker et al., 1999	1999	✓			✓
The multi-mode resource-constrained project scheduling problem with generalized precedence relations	Reyck and Herroelen, 1999	1999	✓			✓
Project scheduling under limited resources: Models, methods, and applications	Hartmann, 1999	1999	✓	✓		✓
Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem	Hartmann and Kolisch, 2000	2000				✓
An integrated survey of deterministic project scheduling	Kolisch and Padman, 2001	2001	✓			✓
Project Scheduling - A Research Handbook	Erik L. Demeulemeester, 2002	2002				
Robust and reactive project scheduling: a review and classification of procedures	Herroelen and Leus, 2004	2004	✓			
Project scheduling: Theory and practice	Herroelen, 2005	2005	✓			✓
Project scheduling under uncertainty: Survey and research potentials	Herroelen and Leus, 2005	2005	✓			✓
Multi-mode resource constrained project scheduling: scheduling schemes, priority rules and mode selection rules.	Lova, Tormos, and Sanchís, 2006	2006				✓
Experimental investigation of heuristics for resource-constrained project scheduling: An update	Kolisch and Hartmann, 2006	2006				✓
Perspectives in modern project scheduling	Józefowska and Weglarz, 2006	2006	✓			✓
Evolutionary algorithms applied to project scheduling problems - a survey of the state-of-the-art	Lancaster and Ozbayrak, 2007	2007				✓
The multimode resource constrained multiproject scheduling problem: Alternative formulations	Zapata, Hodge, and Reklaitis, 2008	2008	✓	✓		
Process scheduling under uncertainty: Review and challenges	Li and Ierapetritou, 2008	2008	✓			✓
A survey of variants and extensions of the resource-constrained project scheduling problem	Hartmann and Briskorn, 2010	2010	✓			✓
Project scheduling with finite or infinite number of activity processing modes: A survey	Weglarz et al., 2011	2011	✓	✓	✓	✓
Project scheduling with time windows and scarce resources: temporal and resource-constrained project scheduling with regular and nonregular objective functions	Neumann, Schwindt, and Zimmermann, 2012	2012	✓	✓		✓
Resource-constrained project scheduling: models, algorithms, extensions and applications	Artigues, Demassey, and Neron, 2013	2013	✓	✓	✓	✓
A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions	Abdolshah, 2014	2014	✓			✓
Discrete-continuous project scheduling with discounted cash inflows and various payment models—a review of recent results	Waligóra, 2014	2014	✓	✓		
Resource-constrained project scheduling problem: review of past and recent developments	Farhad Habibi and Sadjadi, 2018	2018	✓			✓

Table 2.3: [1] : Components of the project; [2] : Mathematical formulation; [3] : Complexity; [4] Types of solution. Papers covering specific domains or attributes are marked with "✓".

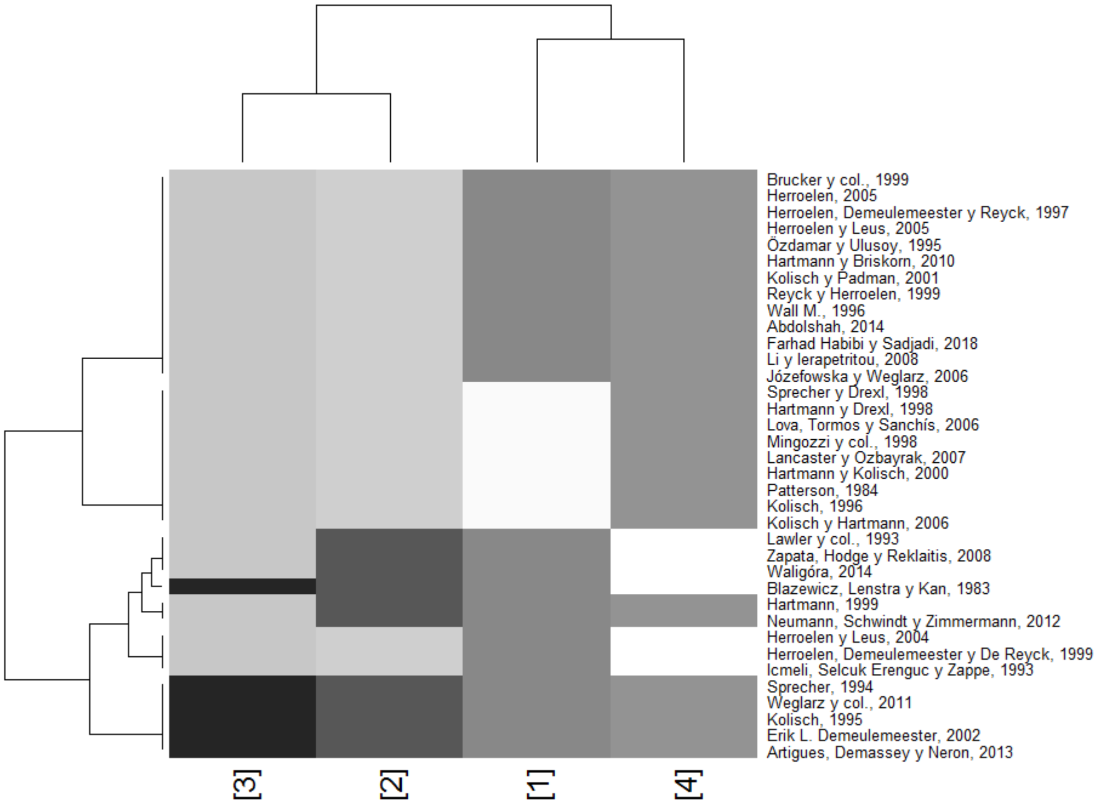


Figure 2.4: Graphic description of the base documents of the taxonomy.

1. **Project components:** The components of the project are related to the description made in Herroelen, Demeulemeester, and De Reyck, 1999, and Brucker et al., 1999, where they provide the characteristics of the resources, activities, performance measures and classifications to the various problems. Without leaving aside, the important contributions shown in Kolisch and Padman, 2001; Kolisch and Hartmann, 2006; Lancaster and Ozbayrak, 2007; Hartmann and Briskorn, 2010; Weglarz et al., 2011; Farhad Habibi and Sadjadi, 2018 and others.

   1.1 Constraints: In the problems of scheduling activities, tasks or other work, it must always be kept in mind that these must be processed using a source or supply from which the performance of the activities or tasks will occur. These supplies are called resources. Among the resources we have machinery, workers, money, materials, etc. Since most of the resources used in a project have a limited availability or each activity requires the use of a percentage of the resource, projects introduce resource constraints in their scheme. In addition, each activity must be performed within a certain time, with the objective of providing an optimal schedule subject to resource and time constraints. Among the time constraints, we have the project completion date or a deadline, fixed processing times, waiting times, etc. The object of study is multi-mode resource-constrained project scheduling problems, which link time and resource constraints in substitution, Talbot, 1982; Blazewicz, Lenstra, and Kan, 1983

1.2 **Precedence relationships:** Technological reasons derive in the sequencing of an activity that depends on the execution of another. These relationships between activities are called **precedence relationships**, because the usual (and simplest) way in which they appear is based on the fact that one activity cannot start before another(s) finishes. Precedence relationships between activities can be visualized by representing the project by means of a directed graph where each activity is represented by a node or vertex and a precedence relationship between two activities by a directed edge. This representation is known as RAN. The β_2 characteristic of the activities in the Herroelen classification. Herroelen, Demeulemeester, and De Reyck, 1999 has 5 groupo $\{^\circ, cpm, \min, gpr, prob\}$. The cpm relationships are those in which the activities have precedence relationships with zero delays or the activities start immediately after the predecessors finish. According to Reyck and Herroelen, 1999, a min relation specifies that an activity can only start (finish) when the predecessor activity has already started (finished) during a period of time. For the gpr $(\min + \max)$case, the max relation specifies that an activity must start (finish) no later than a certain number of time periods beyond the start (finish) of another activity. A minimum delay is essentially a generalized precedence relation that can be transformed into a nonnegative start-start time delay, with the additional assumption that no cycles can occur in the network. In the $prob$ group, the network of activities is of the probabilistic type in which the evolution of the corresponding project is not uniquely determined in advance, see Elmaghraby, 1977; Neumann and Steinhardt, 1979.

1.3 **Resources**: Resources are classified by category, type, and number. The classification by numbers depends on how many units of resources are to be used. Among these, we have discrete resources that can be assigned to activities in discrete quantities. For example machines, tools, workers, etc. We also have continuous resources which can be allocated in arbitrary, a priori unknown quantities from a given interval. For example energy, liquids, and money. Another category is that of preferred resources. A resource is preferred if each of its units can be removed from the processing of the current activity, assigned to another activity, and then returned to the previously interrupted activity whose processing can be resumed as if the resource interruption had not occurred. resource interruption had not occurred. Resources without this property are called non-preferred. Note that the occurrence of non-preferential resources may result in deadlocks, as considered in the theory of operating systems (see for example (see, for example Coffman and

Denning, 1973). Consequently, resources of this type are of particular importance in computer systems.

Resources by category are considered according to their limitations. The basic categories of resources are renewable, non-renewable, and doubly limited resources. This category is the one taken into account in any project planning and programming documents in recent decades. Renewable resources are those that can be restored at a rate similar to or greater than the rate of consumption while being available from period to period. Non-renewable resources are limited over the entire duration of the project, with no period-by-period restrictions. Doubly constrained resources are constrained over both the planning horizon and over the planning horizon and on a period-by-period basis. For a more detailed look at these categories, see Weglarz et al., 2011.

1.4 **Activities:** in general, it is usually convenient to divide the project into work packages, as it allows to work packages, as this allows the project to be broken down into clearly identifiable parts. identifiable parts. Each of these parts can be broken down into interdependent activities or tasks to be performed. activities or tasks to be performed, which are interdependent among them. The activities must have the following characteristics:

- Be measurable in terms of time, resources, effort, and cost.
- To have a final product as a result.
- Have a clear starting point and an end.
- To be the responsibility of a person.

The information we need for each activity can be summarized as follows:

- Description of the task.
- Inputs or necessary preconditions.
- Resource requirements with costs.
- Estimated time.

The classification made by Herroelen, Demeulemeester, and De Reyck, 1999 is one of the most complete. However, in this document, we will only categorize activities by resource requirements, durations, modes of execution, and preventive activities. For the case of resource requirements, we will consider those activities that have a discrete or continuous demand for a certain type of resource. The duration of the activities is handled in a deterministic or stochastic way. We make the distinction of execution modes to classify single-mode and multi-mode problems. In many project scheduling models it is assumed that some activities are not interrupted (non-preferred) and others can be interrupted (preferred).

1.5 Objectives: All optimization problems consist of finding a *best* solution concerning a given criterion. Generally, an evaluation function or performance measure is the criterion by which the quality of a solution is quantified. Objective functions are based on time, resource, quality, and cost criteria. For more details, see Sprecher, 1994; Slowinski, 1989; Kolisch and Padman, 2001; Weglarz et al., 2011; Ballestín and Blanco, 2011. Let's look at some performance measures that will be studied later.

- Minimize project duration: this is undoubtedly the most widely applied measure in the project sequencing domain. Duration is defined as the time interval between the beginning and the end of the project. Since the start of the project is usually assumed to be at time $t = 0$, minimizing the duration is equivalent to minimizing the maximum of the completion times of all activities.
- Maximize the net present value of the project: when large, long-term projects involve significant amounts of cash flow, in the form of expenditures to initiate activities and payments to complete parts of the project, the net present value (NPV) is an appropriate criterion to measure the project's optimality. This criterion generates a critical path of cost and not the critical path of time generated when duration is minimized.
- Maximize project quality: this objective is very important for project managers. The quality of a project is given by the fact that the project is done within the planned deadlines, meets the budget, and that the client is satisfied with the product. The formulation of this problem has focused on minimizing the deviation from the deadlines and the budget due to activities that must be preprocessed.
- Minimize project cost: This objective has attracted much attention from researchers because of its practical relevance. The cost of activities can be minimized since different ways of developing an activity result in different direct costs, which must be minimized (cost-duration tradeoff). On the other hand, it can be considered to minimize the cost of resources which is determined by the sequencing of activities, which influences the cost indirectly through the resources.

1.6 Representation: According to Erik L. Demeulemeester, 2002, there are two ways of representing the relationship between activities, precedence relationships, and resource consumption: Activities at nodes (AoN) and activities on arcs (AoA). An AoA diagram is based on the idea that each activity is a transition of events, whereas AoN activities are established at the nodes, being the arcs of the existing precedence relationship between activities. Kyriakidis, Kopanos, and Georgiadis, 2012 proposed a new form of representation based on activities-resources (RTN). This representation considers three types of nodes for logical linkage, dependencies (activities and decision tables), and a new node to include resources.

2. **Mathematical formulation:** This class seeks to separate the different approaches in the mathematical modeling of problems associated with project scheduling. For example, Talbot, 1982 made a formulation for the MRCPSP that consisted of an integer linear programming model. Among the types of formulations are those based on the type of variables: discrete (integer), continuous and mixed.

3. **Complexity:** this class concentrates all the efforts of the researchers to determine whether the respective project scheduling problem is easy or difficult to solve. The contributions of the researchers will be taken into account by investigating the algorithmic complexity of the problems. For example, Blazewicz, Lenstra, and Kan, 1983 was one of the researchers who contributed to the development of computational complexity.

4. **Solution methods:** For solution, types can be differentiated by types of methods, described in Figure 2.3. For example, the exact methods for solving the MRCPSP, shown by (Hartmann and Drexl, 1998). Most of these methods are based on *branch and bound* procedures, i.e., dividing the problem into subproblems and trying to solve the subproblems by computing lower bounds or using dominance rules. Exact methods ensure the optimal solution of the problem, although, in general, they require a large computational effort, since they are NP-hard problems (Blazewicz, 1986). However, it is possible to obtain sequences of good good quality sequences using approximate methods: heuristics, metaheuristics, or hybrids.

The articles that are selected in the Tables 2.6 and 2.7, are part of the total number of review of the total number of reviewed documents to which the taxonomy was applied.

For the types of solution can be differentiated by methods being utilized (i.e., exact, lower bound, heuristics and metaheuristic solutions) for the MRCPSP, escribed in Figure 2.3 and Table 2.5. These solution methods are presented by Abdolshah, 2014 focused on the resource-constrained project scheduling problem (RCPSP). For example, most of the exact methods to solve the MRCPSP shown by (Hartmann and Drexl, 1998) are based on the procedures of *branch and bound*, that is, to divide the problem into subproblems and try to solve the subproblems by calculating lower bounds or using rules of dominance. The exact methods ensure the optimal solution of the problem, although, in general, they require a great computational effort, because they are NP-hard problems (Blazewicz Blazewicz, Lenstra, and Kan, 1983). However, it is possible to obtain good quality schedules using approximate methods such as heuristic, metaheuristic or hybrid algorithms. Figure 2.5 shows that metaheuristic procedures in 70% of research papers, heuristic procedures are used in 10%, exact procedures in 18% and lower bound procedures in 2%. Selected papers are presented in Tables 2.6 and 2.7.

- 1. Project components
  - 1.1 Restrictions
    - 1.1.1 Time
    - 1.1.2 Resources
  - 1.2 Precedence relations
    - 1.2.1 *cpm*
    - 1.2.2 min
    - 1.2.3 *gpr*
    - 1.2.4 *prob*
  - 1.3 Resources
    - 1.3.1 Resources by numbers
    - 1.3.2 Preferred resources
    - 1.3.3 Resource categories
      - 1.3.3.1 Renewables
      - 1.3.3.2 Non-renewable
      - 1.3.3.3 Doubly restricted
      - 1.3.3.4 Partially renewable
    - 1.3.4 Other Categories
      - 1.3.4.1 Dedicated
      - 1.3.4.2 Space
      - 1.3.4.3 Adjacent
      - 1.3.4.4 Acumulative
      - 1.3.4.5 Reusables
      - 1.3.4.6 Synchronized
      - 1.3.4.7 Multiple Skills
      - 1.3.4.8 Heterogeneous
      - 1.3.4.9 Exchange Resources
  - 1.4 Activities
    - 1.4.1 Resource requirements
    - 1.4.2 No-preemted
    - 1.4.3 Preempted
      - 1.4.3.1 preemted-resume
      - 1.4.3.2 preemted-repeat
    - 1.4.4 Durations
      - 1.4.4.1 Deterministic
      - 1.4.4.2 Stochastics
    - 1.4.5 Execution modes
      - 1.4.5.1 Single mode
      - 1.4.5.2 Multi-mode
  - 1.5 Objectives
    - 1.5.1 Time-based
      - 1.5.1.1 Minimize project duration
      - 1.5.1.2 Minimize expected project completion time
      - 1.5.1.3 Minimization of weighted average flow time
      - 1.5.1.4 Minimize weighted sum total of start times
      - 1.5.1.5 Minimize maximum tardiness
      - 1.5.1.6 Minimize the average (weighted) tardiness
      - 1.5.1.7 Minimize weighted total tardiness
      - 1.5.1.8 Minimize total weighted flow time
      - 1.5.1.9 Minimize early (weighted) average
      - 1.5.1.10 Minimize the number of late activities
      - 1.5.1.11 Minimize sum of start times
      - 1.5.1.12 Minimize the weighted earliest and latest completion time.
    - 1.5.2 Resource-based
      - 1.5.2.1 Minimize cost of renewable resource availability
      - 1.5.2.2 Minimize average resource use
      - 1.5.2.3 Minimize total adjustment cost
      - 1.5.2.4 Minimize total cost of resource use
      - 1.5.2.5 Minimization of project cost dependent on mode or weighted resource consumption
      - 1.5.2.6 Minimize total rental cost
    - 1.5.3 Financial
      - 1.5.3.1 Maximize net present value
      - 1.5.3.2 Minimize project implementation cost
      - 1.5.3.3 Minimize project completion cost
      - 1.5.3.4 Minimize the total cost of penalties
      - 1.5.3.5 Minimize the total expected cost of the project
      - 1.5.3.6 Minimize the total weighted penalty cost for advancing-delaying the project
    - 1.5.4 Based on quality
      - 1.5.4.1 Maximizing the total weighted sum of the quality of activities
    - 1.5.5 Other objectives:
      - 1.5.5.1 Based on reprogramming of activities.
    - 1.5.6 Regular and non-regular measurements
      - 1.5.6.1 Regular
      - 1.5.6.2 No-regular
  - 1.6 Representation
    - 1.6.1 Representation of activities in arcs (AoA)
    - 1.6.2 Representation of Activities in nodes (AoN)
    - 1.6.3 Resource-Task Network (RTN) Representation
- 2. Mathematical formulation
  - 2.1 Entera
  - 2.2 Continues
  - 2.3 Mixed
- 3. Complexity

Table 2.4: Project components, formulation and complexity

2.4 Systematic literature review on reactive project scheduling.

Reactive scheduling refers to a situation where we cannot (or do not) plan ahead for problems or opportunities. Simply put, we only react when such problems or opportunities occur. In contrast, proactive scheduling occurs when we plan ahead to manage or avoid problems. During execution, projects can be subject to considerable uncertainty, which can lead to numerous schedule disruptions. Among these disruptions are:

1. Interruptions in the project network.

- 4. Types of solutions
  - 4.1 Exact solution
    - 4.1.1 Mathematical planning
      - 4.1.1.1 Linear planning
      - 4.1.1.2 Integer planning
      - 4.1.1.3 Zero-one method
    - 4.1.2 Numerical methods
      - 4.1.2.1 Dynamic planning
      - 4.1.2.2 Branch and Bound method
        - **4.1.2.2.1** Branch and bound
        - **4.1.2.2.2** Branch and cut
        - **4.1.2.2.3** Branch and price
    - 4.1.3 Synthetic methods
      - 4.1.3.1 Critical path-PERT
      - 4.1.3.2 Simulation
      - 4.1.3.3 Synthetic
    - 4.1.4 Stochastic
      - 4.1.4.1 Markov chain
      - 4.1.4.2 Goal theory
      - 4.1.4.3 System analysis
  - 4.2 Lower bounds
  - 4.3 Heurística
    - 4.3.1 Search based
      - 4.3.1.1 Constructive
        - **4.3.1.1.1** Generation scheduling
          - **4.3.1.1.1.1** Parallel scheduling
          - **4.3.1.1.1.2** Serial scheduling
          - **4.3.1.1.1.3** Double scheduling
        - **4.3.1.1.2** Priority based
          - **4.3.1.1.2.1** Single pass
          - **4.3.1.1.2.2** Multi pass
            - **4.3.1.1.2.2.1** Priority rule
            - **4.3.1.1.2.2.2** Forward backward
            - **4.3.1.1.2.2.3** Simulation methods
      - 4.3.1.2 Improvement
        - **4.3.1.2.1** Neighborhood search
        - **4.3.1.2.2** Forward bacward improvemente
    - 4.3.2 Based on exact methods
      - 4.3.2.1 Descomposition
        - **4.3.2.1.1** Iteration
        - **4.3.2.1.2** Column genation
      - 4.3.2.2 Relaxation
        - **4.3.2.2.1** Exact method
        - **4.3.2.2.2** Lagrange
    - 4.3.3 Hybrid
      - 4.3.3.1 Combination with several heuristic
  - 4.4 Metaheuristics
    - 4.4.1 Based construction
      - 4.4.1.1 GRASP
      - 4.4.1.2 Variable neighboard search (VNS)
      - 4.4.1.3 Ant Colony Optimization (ACO)
      - 4.4.1.4 Guided local search (GLS)
      - 4.4.1.5 Iterated local search (ILS)
    - 4.4.2 Local search
      - 4.4.2.1 Simulated annealing (SA)
      - 4.4.2.2 Hill climbing
      - 4.4.2.3 Tabu search (TS)
      - 4.4.2.4 Random optimization
    - 4.4.3 Population based
      - 4.4.3.1 Evolutionary algorithm
      - 4.4.3.2 Genetic algorithm
      - 4.4.3.3 Genetic programming
      - 4.4.3.4 Evolution Strategies
      - 4.4.3.5 Evolutionary programming
      - 4.4.3.6 Otros
        - **4.4.3.6.1** Estimation of distribution
        - **4.4.3.6.2** Differential evolution
        - **4.4.3.6.3** Coevolutionary algorithms
        - **4.4.3.6.4** Memetics algorithms
    - 4.4.4 Swarm inteligence
      - 4.4.4.1 Artificial ant colony optimization
      - 4.4.4.2 Particle swarm optimization
      - 4.4.4.3 Artificial bee colony optimization
      - 4.4.4.4 Artificial immune system
    - 4.4.5 Artificial intelligence
      - 4.4.5.1 Artificial neural network
    - 4.4.6 Hybrid
      - 4.4.6.1 Exact methods
      - 4.4.6.2 Metaheuristic
    - 4.4.7 Metahybrid
      - 4.4.7.1 Heuristic

Table 2.5: Types of Solutions.

2. Interruption of a new activity
3. Precedence interruption
4. Activity interruptions.
    - Interruption of the duration of the activity
    - Interruption of activity resource
5. Resource interruptions
    - Renewable resource disruptions
    - Nonrenewable resource disruptions
6. Multiple disruptions

All of these types of interruptions can result in the impossibility of the project's reference scheduling. In general, project management wants to avoid these scheduled breaks. schedule breaks. This can be accomplished by generating a baseline schedule

		1.1		1.2			1.3														1.4						1.5							1.6			2			3
Author	Year	1.1.1	1.1.2	1.2.1	1.2.2	1.2.3	1.3.1	1.3.2	1.3.3.1	1.3.3.2	1.3.3.3	1.3.3.4	1.3.4.1	1.3.4.2	1.3.4.3	1.3.4.4	1.3.4.5	1.3.4.6	1.3.4.7	1.3.4.8	1.3.4.9	1.4.1	1.4.2	1.4.3	1.4.4	1.4.5.1	1.4.5.2	1.5.1	1.5.2	1.5.3	1.5.4	1.5.5	1.5.6	1.5.7	1.6.1	1.6.2	1.6.3	2.1	2.2	2.3	3
J. Blazewicz Blazewicz, Lenstra, and Kan, 1983	1983	✓	✓	✓			✓		✓	✓													✓	✓				✓	✓	✓	✓	✓	✓	✓		✓					✓
R. Kolisch Kolisch and Sprecher, 1996	1996	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓		✓			
M. Mori Mori, 1997	1997	✓	✓	✓					✓	✓													✓			✓		✓								✓					
L. Bianco Bianco, Dell'Olmo, and Speranza, 1998	1998	✓	✓	✓			✓						✓										✓			✓		✓								✓		✓			
S. Hartmann Hartmann and Drexl, 1998	1998	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓					
L. Ozdamar Özdamar, 1999	1999	✓	✓	✓					✓	✓													✓			✓		✓								✓					
J. Jozefowska Jozefowska et al., 2001	2001	✓	✓	✓					✓	✓													✓			✓		✓								✓					
S. Hartmann Hartmann, 2001	2001	✓	✓	✓					✓	✓													✓			✓		✓								✓					
K. Nonobe Nonobe and Ibaraki, 2002	2002	✓	✓				✓																																		
J. Alcaraz Alcaraz, Maroto, and Ruiz, 2003	2003	✓	✓	✓					✓	✓													✓			✓		✓								✓					
J. Choi Choi, Realff, and Lee, 2004	2004	✓	✓	✓			✓		✓																		✓	✓								✓		✓			
H. Ke Ke and Liu, 2005	2005	✓	✓	✓																							✓	✓													
G. Zhu Zhu, Bard, and Yu, 2006	2006	✓	✓	✓			✓					✓											✓			✓		✓								✓		✓			
H. Zhang Zhang, Tam, and Li, 2006	2006	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓					
G. Zhu Zhu, Bard, and Yu, 2007	2007	✓	✓	✓			✓		✓																		✓	✓													
J. Zapata Zapata, Hodge, and Reklaitis, 2008	2008	✓	✓	✓			✓		✓														✓			✓														✓	
F. Ballestín Ballestín, Valls, and Quintanilla, 2008	2008	✓	✓	✓					✓															✓		✓		✓								✓					
V. Tiwari Tiwari, Patterson, and Mabert, 2009	2009	✓	✓	✓			✓													✓			✓			✓		✓								✓		✓			
N. Damak Damak et al., 2009	2009	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓					
T. Wauters Wauters et al., 2009	2009	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓					
A. Lova Lova et al., 2009	2009	✓	✓	✓					✓	✓													✓			✓		✓								✓					
M. Ranjbar Ranjbar, De Reyck, and Kianfar, 2009	2009	✓	✓	✓					✓	✓	✓												✓			✓		✓								✓					
L. Tseng Tseng and Chen, 2009	2009	✓	✓	✓					✓	✓													✓			✓		✓								✓					
S. Elloumi Elloumi and Fortemps, 2010	2010	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓					
W. Chen Chen et al., 2010a	2010	✓	✓	✓			✓		✓														✓			✓		✓								✓					
P. Van Van Peteghem and Vanhoucke, 2010	2010	✓	✓	✓					✓	✓													✓	✓		✓		✓								✓					
L. Wang Wang and Fang, 2011	2011	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓					
P. Van Van Peteghem and Vanhoucke, 2011	2011	✓	✓	✓					✓	✓													✓	✓												✓					
F. Deblaere Deblaere, Demeulemeester, and Herroelen, 2011b	2011	✓	✓	✓					✓	✓															✓																
T. Kyriakidis Kyriakidis, Kopanos, and Georgiadis, 2012	2012	✓	✓	✓			✓		✓	✓													✓			✓		✓									✓			✓	
L. Wang Wang and Fang, 2012	2012	✓	✓	✓			✓		✓	✓													✓			✓		✓								✓					
M. Sebt Sebt, Alipouri, and Alipouri, 2013	2013	✓	✓	✓			✓		✓														✓			✓		✓								✓					
H. Gomes Gomes, Neves, and Souza, 2014	2014	✓	✓	✓			✓		✓														✓			✓		✓								✓					
G. Koulinas Koulinas, Kotsikas, and Anagnostopoulos, 2014	2014	✓	✓	✓			✓		✓														✓			✓		✓								✓					
R. Chakrabortty Chakrabortty, Sarker, and Essam, 2016	2016	✓	✓	✓				✓	✓	✓													✓	✓		✓		✓								✓				✓	
Z. Zhang Zhang and Xu, 2016	2016																										✓		✓		✓										
M. Sebt Sebt, Afshar, and Alipouri, 2016	2016	✓	✓	✓					✓	✓													✓			✓		✓								✓					

Table 2.6: Attribute classification (1.1.1 to 3).

		4																																																				
		4.1												4.2	4.3													4.4																								
Author	Year	4.1.1.1	4.1.1.2	4.1.1.3	4.1.2.1	4.1.2.2.1	4.1.2.2.2	4.1.2.2.3	4.1.3.1	4.1.3.2	4.1.3.3	4.1.4.1	4.1.4.2	4.1.4.3	4.2	4.3.1.1.1.1	4.3.1.1.1.2	4.3.1.1.1.3	4.3.1.1.2.1	4.3.1.1.2.2.1	4.3.1.1.2.2.2	4.3.1.1.2.2.3	4.3.1.2.1	4.3.1.2.2	4.3.2.1.1	4.3.2.1.2	4.3.2.2.1	4.3.2.2.2	4.3.3.1	4.4.1.1	4.4.1.2	4.4.1.3	4.4.1.4	4.4.1.5	4.4.2.1	4.4.2.2	4.4.2.3	4.4.2.4	4.4.3.1	4.4.3.2	4.4.3.3	4.4.3.4	4.4.3.5	4.4.3.6.1	4.4.3.6.2	4.4.3.6.3	4.4.3.6.4	4.4.4.1	4.4.4.2	4.4.4.3	4.4.4.4	4.4.5.1	4.4.6.1	4.4.6.2	4.4.7.1
R. Słowiński Słowiński, Soniewicki, and Weglarz, 1994	1994					✓										✓																				✓																				
F. Boctor Boctor, 1996	1996															✓																				✓																				
M. Mori Mori, 1997	1997															✓			✓																				✓															
L. Bianco Bianco, Dell'Olmo, and Speranza, 1998	1998								✓												✓									✓																										
S. Hartmann Hartmann and Drexl, 1998	1998					✓									✓		✓		✓	✓																																			
A. Sprecher Sprecher and Drexl, 1998	1998					✓										✓		✓	✓																																			
L. Ozdamar Özdamar, 1999	1999															✓				✓	✓																				✓															
J. Jozefowska Jozefowska et al., 2001	2001															✓		✓																✓																				
S. Hartmann Hartmann, 2001	2001															✓		✓	✓																				✓															
K. Nonobe Nonobe and Ibaraki, 2002	2002																																				✓																		
J. Alcaraz Alcaraz, Maroto, and Ruiz, 2003	2003															✓		✓	✓	✓																			✓															
K. Bouleimen Bouleimen and Lecocq, 2003	2003															✓																				✓																				
J. Choi Choi, Realff, and Lee, 2004	2004									✓		✓																																										
H. Ke Ke and Liu, 2005	2005									✓																																												
G. Zhu Zhu, Bard, and Yu, 2006	2006						✓								✓					✓																																		
H. Zhang Zhang, Tam, and Li, 2006	2006															✓																																✓						
G. Zhu Zhu, Bard, and Yu, 2007	2007																																																						
F. Ballestín Ballestín, Valls, and Quintanilla, 2008	2008															✓		✓																																					
B. Jarboui Jarboui et al., 2008	2008																																																	✓						
C. Chiang Chiang, Huang, and Wang, 2008	2008																															✓																						
V. Tiwari Tiwari, Patterson, and Mabert, 2009	2009		✓												✓																																							
N. Damak Damak et al., 2009	2009																																													✓										
T. Wauters Wauters et al., 2009	2009															✓																								✓															
A. Lova Lova et al., 2009	2009																✓	✓	✓				✓										✓						✓															
V. Pethegem Van Peteghem and Vanhoucke, 2009	2009															✓																																		✓				
M. Ranjbar Ranjbar, De Reyck, and Kianfar, 2009	2009															✓																							✓																
L. Tseng Tseng and Chen, 2009	2009																																							✓															
S. Elloumi Elloumi and Fortemps, 2010	2010																			✓																			✓																
W. Chen Chen et al., 2010a	2010																✓		✓				✓								✓		✓																				
V. Peteghem Van Peteghem and Vanhoucke, 2010	2010															✓		✓	✓																				✓															
L. Wang Wang and Fang, 2011	2011															✓			✓				✓																							✓							
V. Peteghem Van Peteghem and Vanhoucke, 2011	2011								✓						✓		✓													✓								✓																
F. Deblaere Deblaere, Demeulemeester, and Herroelen, 2011b	2011																																																						
H. Zang Zhang, 2011	2011															✓			✓												✓																						
J. Coelho Coelho and Vanhoucke, 2011	2011																																							✓															
L. Wang Wang and Fang, 2012	2012															✓				✓																							✓												
C. Chiang Chiang and Huang, 2012	2012								✓						✓				✓												✓																						
M. Sebt Sebt, Alipouri, and Alipouri, 2013	2013															✓																										✓													
O. Mirzaei Mirzaei and Akbarzadeh-T, 2013	2013															✓																																					✓	
H. Li Li and Zhang, 2013	2013															✓																															✓							
H. Shen Shen and Li, 2013	2013																			✓																													✓						
R. Chen Chen and Wang, 2013	2013																✓																															✓						
H. Gomes Gomes, Neves, and Souza, 2014	2014															✓		✓	✓										✓	✓			✓																				
G. Koulinas Koulinas, Kotsikas, and Anagnostopoulos, 2014	2014															✓		✓	✓	✓								✓																				✓						
A. Vartouni Vartouni and Khanli, 2014	2014																																							✓														✓	
O. Soliman Soliman and Elgendi, 2014	2014															✓						✓																				✓										
Y. Xu Xu et al., 2014	2014																																																			✓			✓	
L. Wang Wang, Liu, and Zhou, 2014	2014																																														✓									
A. Chen Chen, Liang, and Padilla, 2014	2014																																																		✓					
J. Cui Cui and Yu, 2014	2014															✓																																✓						
S. Asta **Asta2015**	2015															✓											✓		✓				✓												✓							
L. Zhang Zhang, Luo, and Zhang, 2015	2015															✓																												✓				✓				✓	
P. Jedrzejowicz Jedrzejowicz and Ratajczak-Ropel, 2015	2015																																																						✓	
M. Sebt Sebt, Afshar, and Alipouri, 2015	2015															✓						✓																✓															
R. Chakrabortty Chakrabortty, Sarker, and Essam, 2016	2016					✓									✓																									✓															
Z. Zhang Zhang and Xu, 2016	2016																																						✓										✓						
M. Sebt Sebt, Afshar, and Alipouri, 2016	2016															✓			✓				✓																✓									✓					✓	

Table 2.7: Attribute classification (4).

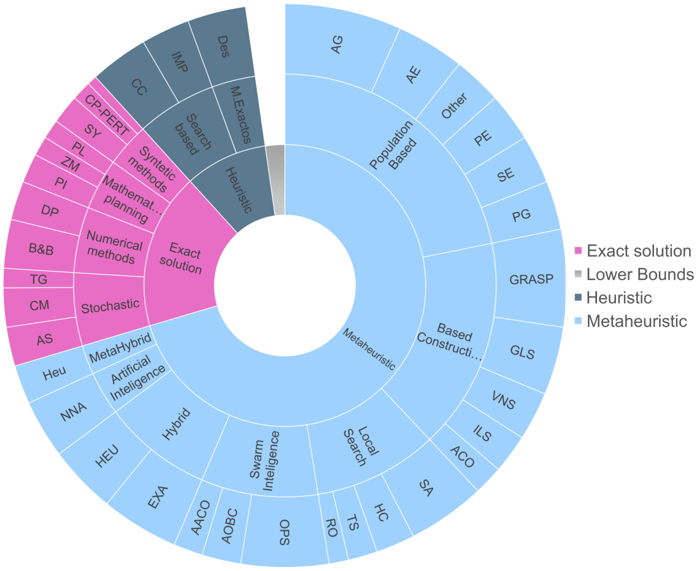


Figure 2.5: Sunburst charts to represent the solution methods

proactively, by proactively, trying to anticipate certain types of interruptions to minimize they're to minimize their effect if they occur. If the schedule still breaks down despite these proactive planning efforts, a reactive scheduling policy will be necessary to repair the unworkable schedule. unfeasible schedule. In the realm of proactive scheduling and reactive scheduling of projects for MRCPSP, some work has already been done. some work has already been done.

Extensive research has been conducted on issues related to reactive scheduling in project outage events; however, a systematic literature review reflecting the state of the art of this crucial field has not been performed. Therefore, we decided to conduct a study to highlight current theoretical and empirical developments in the domain of reactive scheduling when outage events occur in projects and to exploit questions that have not yet been answered by the literature, but researchers and practitioners are still very interested to know. In this study, we focus on the following:

1. Analyzed trends and distribution / literature patterns based on reactive scheduling of projects .
2. Propose a classification framework to highlight emerging issues and research problems not yet addressed in the field of reactive project scheduling.
3. A reactive project scheduling monitoring and evaluation framework is recommended.

2.4.1 Method

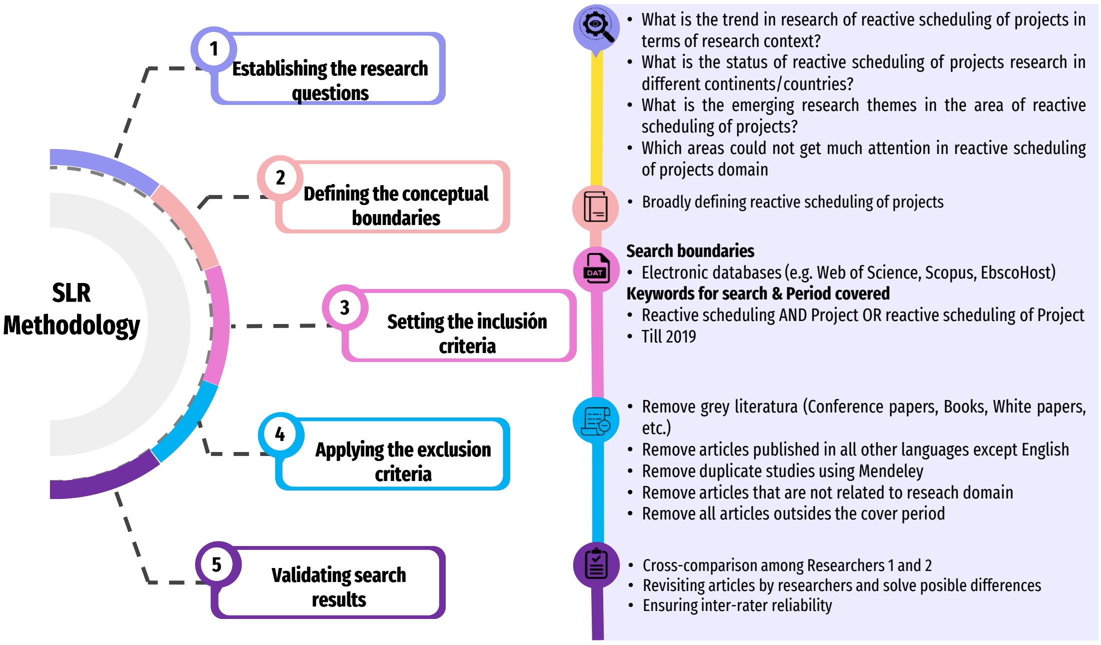
We followed the systematic literature review (SLR) methodology suggested by (Gupta et al., 2019), (Tranfield, Denyer, and Smart, 2003) and (Macpherson and Jones, 2010).

Data sources and search strategy

To collect relevant articles, we scanned the most important databases, including Web of Science and Scopus, with the search limited to 2000 to 2020. Our search in all databases focused on the following string: ***reactive*** AND ***project scheduling***. We limited our search to article titles only to select studies that had "projects" as a central part of the discussion and analysis.

Study selection

For the selection of studies, we followed the inclusion and exclusion criteria listed in Figure 2.6. The article selection process was performed in any phase. First, we included articles written in English and published as journal article, article (book chapter), article (proceedings paper) and review. Therefore, we excluded gray literature (conference papers, doctoral dissertations, workshop abstracts, books, prefaces, poster sessions, and news reports). This phase resulted in 61 publications for consideration. To take this forward in the second stage, based on the title and abstract reviews, we selected papers that had a clear focus on reactive scheduling issues related to project planning and execution. Finally, the rigorous search process resulted in 47 papers to be part of this systematic literature review. In Table 2.8 we detail the information of our data set.


SLR
Methodology
1
Establishing the research questions
2
Defining the conceptual boundaries
3
Setting the inclusión criteria
4
Applying the exclusion criteria
5
Validating search results
DAT

- What is the trend in research of reactive scheduling of projects in terms of research context?
- What is the status of reactive scheduling of projects research in different continents/countries?
- What is the emerging research themes in the area of reactive scheduling of projects?
- Which areas could not get much attention in reactive scheduling of projects domain

- Broadly defining reactive scheduling of projects

Search boundaries
- Electronic databases (e.g. Web of Science, Scopus, EbscoHost)

Keywords for search & Period covered
- Reactive scheduling AND Project OR reactive scheduling of Project
- Till 2019

- Remove grey literatura (Conference papers, Books, White papers, etc.)
- Remove articles published in all other languages except English
- Remove duplicate studies using Mendeley
- Remove articles that are not related to reseach domain
- Remove all articles outsides the cover period

- Cross-comparison among Researchers 1 and 2
- Revisiting articles by researchers and solve posible differences
- Ensuring inter-rater reliability

Figure 2.6: Info-graphics of SLR methodology

After article selection, we managed a database of publications with Mendeley and JabRef, recording relevant information in a Microsoft Excel spreadsheet and completing

Description	**Results**
MAIN INFORMATION ABOUT DATA	
Timespan	2003:2020
Sources (Journals, Books, etc)	36
Documents	47
Average years from publication	7.55
Average citations per documents	29.38
Average citations per year per doc	2.981
References	1
DOCUMENT TYPES	
article	28
article; book chapter	1
article; proceedings paper	3
proceedings paper	13
review	2
DOCUMENT CONTENTS	
Keywords Plus (ID)	85
Author's Keywords (DE)	150
AUTHORS	
Authors	109
Author Appearances	141
Authors of single-authored documents	1
Authors of multi-authored documents	108
AUTHORS COLLABORATION	
Single-authored documents	1
Documents per Author	0.431
Authors per Document	2.32
Co-Authors per Documents	3
Collaboration Index	2.35

Table 2.8: Main information about reactive project scheduling article collection

an idea diagram for each article. The flow of our methodology can be seen in Figure 2.6.

2.4.2 Literature analysis: trends and themes

This review structure is based on 47 research articles published from 2003 to 2020. Figure 2.7, part a. and b. shows the behavior of the peaks in the different years, with a very low average number of citations per year, but with an upward trend in recent years. The reason for this behavior is due to the great interest of researchers in solving reactive scheduling problems.

Studies based on reactive project scheduling have been published in a wide range of engineering and management-oriented journals, see Figure 2.8. The selected research papers belong to 36 journals. Among them, the Journal of scheduling (4), Computers & Industrial Engineering (3), European Journal of Operational Research (3), and International Journal of Production Research (3) are the top four journals (Fig. 3). The top six journals account for only about 17% of the total number of journals, which reflects the

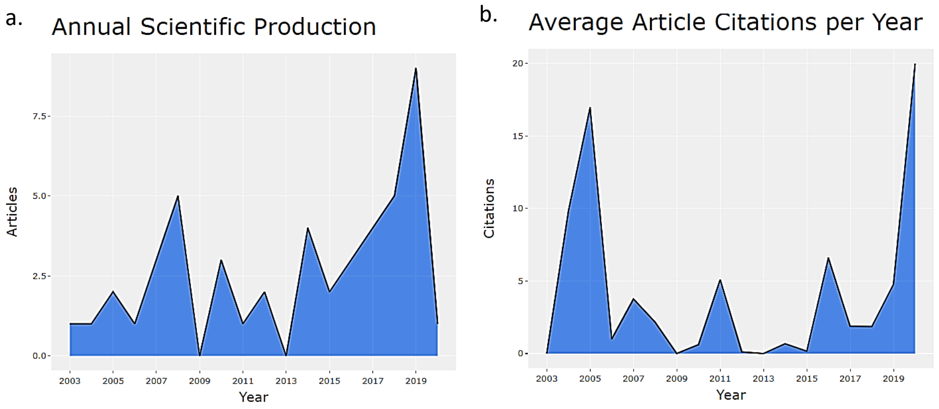


Figure 2.7: Publication trend of studies on reactive scheduling of projects

degree of diversity in reactive project programming-based publications. Most of these journals belong to the overlapping domains of engineering and management covering specialized topics in project implementation and design.

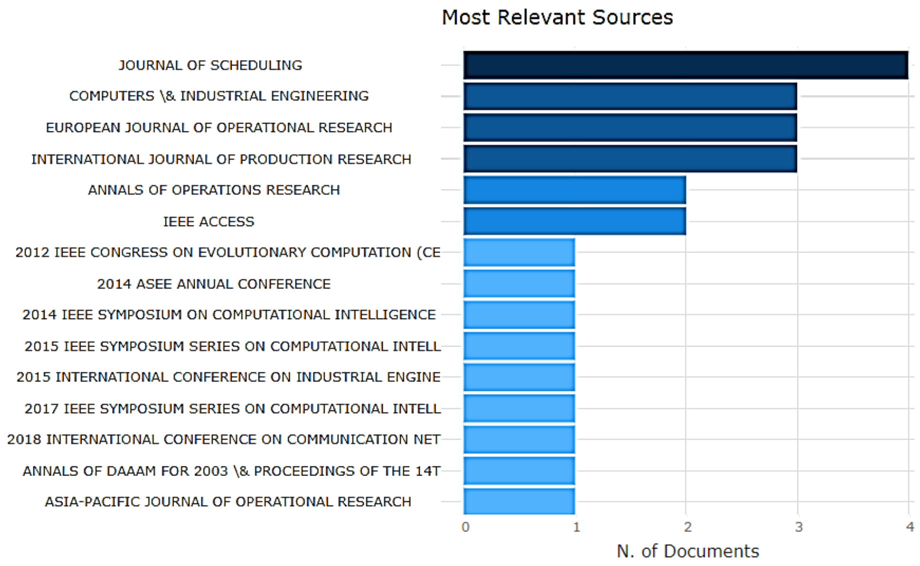


Figure 2.8: Frequency distribution of reactive project scheduling in these leading journals.

We know that the citation value for a research article is determined by how many times an article has been cited or mentioned by other articles. In performing the citation analysis, we relied on a Web of Science managed citation report that only covers articles from journals listed in this database. Our analysis reveals that in the list of the top 16 cited articles, the study published by Herroelen and Leus, 2005 which is a survey dealing with the 5 procedures that exist to deal with uncertainty: reactive scheduling, stochastic scheduling, scheduling under fuzziness, proactive scheduling, sensitivity analysis (citations = 515; Figure 2.9). Interestingly, this same article achieved the highest citation rate (30.29 citations per year) among the selected articles. This highlights the importance of addressing reactive scheduling from the point of view of the author Herroelen and Leus, 2005.

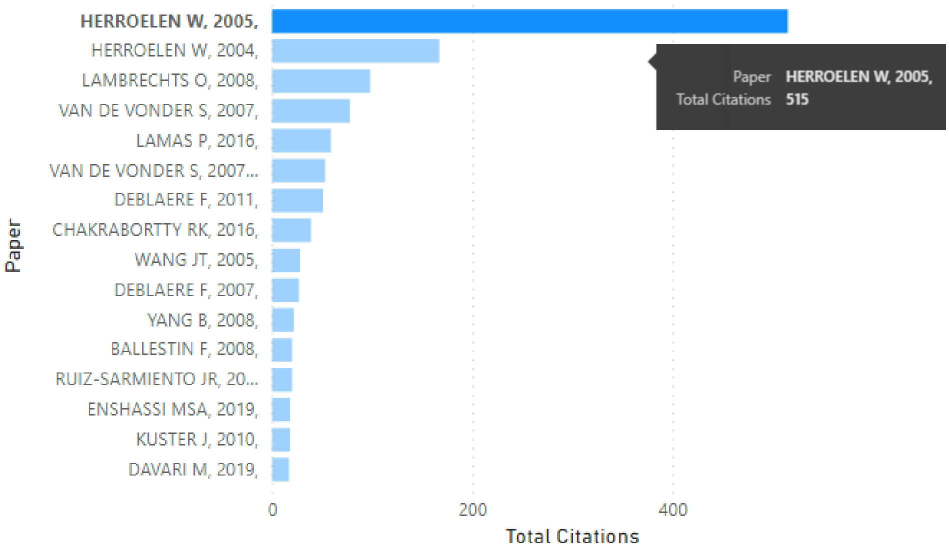


Figure 2.9: Total citations per paper.

Figure 2.10 (a. and b.) highlights the most relevant authors during the last 20 years. Their hard research work and contributions, which have guided the academic world to formulate and develop strategies to deal with reactive scheduling in projects.

2.4.3 Research lines found by the SLR

Below we show the classification we have identified in the context of reactive project scheduling, using the coding of the articles addressed in the study. It is worth highlighting the research of (Herroelen and Leus, 2005) since it has been the starting point of our study. In Table 2.9, you will find a classification for managing uncertainty in projects.

5. Approaches to dealing with uncertainty in a project scheduling environment.
   - 5.1 Proactive (Robust) Project Scheduling
   - 5.2 Generating a Baseline Schedule
   - 5.3 Stochastic Project Scheduling
   - 5.4 Fuzzy Project Scheduling
   - 5.5 Reactive Project Scheduling
6. Classification of reactive reactive.
   - 6.1 Schedule Repair
   - 6.2 Rescheduling
   - 6.3 Contingent Scheduling
   - 6.4 Critical Chain/Buffer Management (CC/BM)
   - 6.5 Activity Crashing
   - 6.6 Sensitivity Analysis

Table 2.9: Classification of approaches to manage project uncertainty..

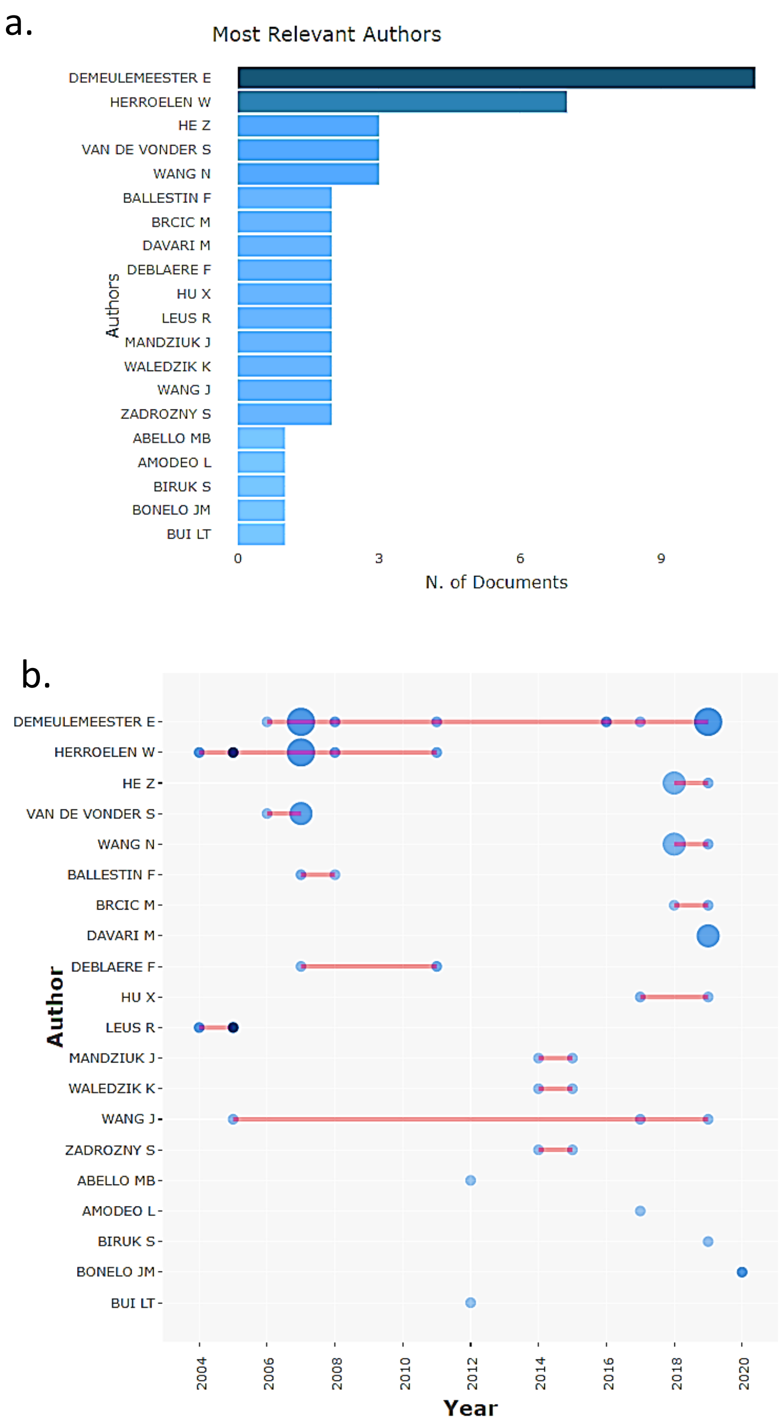


Figure 2.10: Frequency distribution of reactive project scheduling in these leading journals.

From the review of the documents and the above classification, we observe a multidimensional problem: uncertainty management and types of uncertainty. When establishing a reactive strategy, we can select schedule repair, rescheduling, contingent scheduling, critical chain/buffer management (CC/BM), activity crashing, and sensitivity analysis. In Figure 2.11, we observe the relationship between reactive and proactive scheduling, associated with reactive scheduling techniques. Now, by Table 2.10 we can add the solution type dimension.

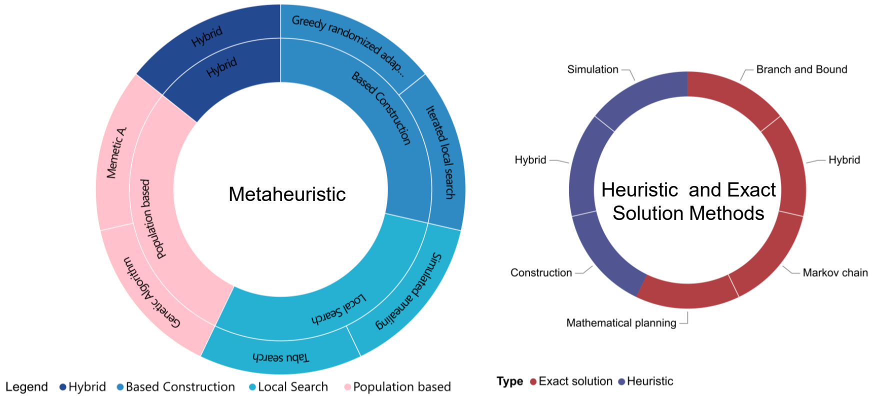


Figure 2.11: Metaheuristic, heuristic and exact strategies found by SLR

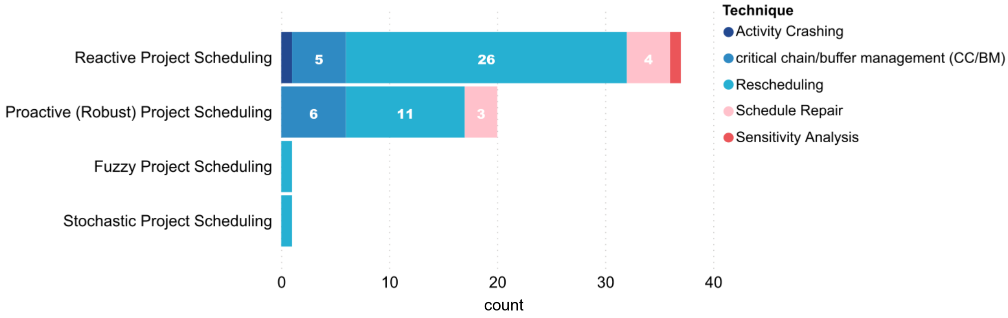


Figure 2.12: Frequency distribution of reactive project scheduling in these leading journals.

2.5 Chapter Summary

In this chapter, we have presented a taxonomy and systematic review of the literature for the field of resource-constrained multiple-mode project scheduling and reactive scheduling to handle events that affect project schedules. In this section, we summarize the overall review and provide future research directions based on the research gaps in the literature.

2.5.1 Taxonomy Summary and Research Directions

In line with the results of our taxonomy review, we identified three lines of future MRCPSP research studies. As very few of the models reviewed are aimed at solving the

Year	Article	5.1.	5.2.	5.3	5.4.4	5.5.	6.1.	6.2.	6.3.	6.4.	6.5.	6.6.	4.1.3.	4.1.1.	4.1.2.2.	4.14.1.	4.3.1.1	4.3.1.1.2.2.3	4.3.3.	4.4.3.1.	4.4.3.5.	4.4.3.2.	4.4.3.6.4.	4.4.1.3.	4.4.1.4.	4.4.1.5	4.4.4.2	4.4.5	4.4.2.1	4.4.2.3	4.4.6.
2019	Adamu, Akinwumi, and Okagbue, 2019					✓		✓									✓														
2019	Biruk and Rzepecki, 2019	✓				✓		✓										✓													
2019	Brčić, Katić, and Hlupić, 2019	✓				✓		✓																	✓						
2019	Davari and Demeulemeester, 2019a	✓				✓		✓							✓																
2019	Davari and Demeulemeester, 2019b	✓				✓		✓								✓															
2019	Hu, Wang, and Leng, 2019	✓				✓	✓										✓														
2019	Ma et al., 2019	✓				✓																									
2019	Wang et al., 2019	✓				✓	✓																							✓	
2018	Brčić and Mlinarić, 2018	✓				✓												✓													
2018	Chakrabortty, Sarker, and Essam, 2018					✓		✓							✓																
2018	Ning et al., 2018	✓								✓																					✓
2018	Song et al., 2018	✓				✓		✓										✓													
2018	Zheng et al., 2018					✓		✓																							✓
2017	Elloumi, Fortemps, and Loukil, 2017					✓		✓														✓									
2017	Hu et al., 2017	✓				✓				✓								✓													
2016	Chakrabortty, Sarker, and Essam, 2016					✓		✓							✓																
2016	Lamas and Demeulemeester, 2016	✓													✓																
2016	Suwa and Morita, 2016					✓		✓																							✓
2015	Szczesny and König, 2015					✓						✓							✓												
2014	Khan, Martini, and Staehle, 2014				✓	✓		✓																							
2014	Wang, Zhan, and Nie, 2014					✓					✓																				
2012	Bagheri, Jolai, and Aryanezhad, 2012	✓				✓		✓										✓													
2012	Wang et al., 2012	✓				✓				✓																					
2012	Yuan, Polychronakis, et al., 2012	✓				✓				✓								✓													
2011	Deblaere, Demeulemeester, and Herroelen, 2011a					✓		✓																							✓
2011	Lambrechts, Demeulemeester, and Herroelen, 2011					✓																									✓
2011	Demeulemeester and Herroelen, 2011	✓				✓		✓											✓												
2010	Gürel, Körpeoğlu, and Aktürk, 2010					✓		✓						✓																	
2010	Herroelen and Demeulemeester, 2010	✓				✓				✓																					
2010	Kuster, Jannach, and Friedrich, 2010					✓		✓														✓									
2008	Ballestin and Trautmann, 2008			✓																						✓					
2008	Lambrechts, Demeulemeester, and Herroelen, 2008	✓				✓				✓																				✓	
2008	Vonder, Demeulemeester, and Herroelen, 2008			✓																										✓	
2008	Yang and Geunes, 2008					✓		✓																							
2007	Deblaere et al., 2007			✓		✓									✓																
2007	Valls et al., 2007	✓				✓		✓															✓								
2007	Vonder, Demeulemeester, and Herroelen, 2007	✓				✓		✓											✓												
2007	Vonder et al., 2007b	✓				✓		✓											✓												
2005	Wang, 2005					✓		✓																							✓
2005	Zhu, Bard, and Yu, 2005					✓									✓																
2002	Calhoun et al., 2002					✓		✓																						✓	

Table 2.10: Attribute classification (5 and 6)

MRCPSP using several objective functions, it would be desirable that exact, heuristic, or metaheuristic solution methods include multiobjective programming in their execution. Secondly, most previous work assumes that activities cannot be stopped or interrupted at any time, which is not necessarily true. It is possible that, due to external conditions, the activities are interrupted and generate penalties for exceeding the time of completion. Research is needed to develop new methods that allow the inclusion of these constraints during the mathematical formulation and posterior computational solution of the MRCPSP. Last but not least, future studies must be focused on developing ways to consider updates to the project's parameters as the project progresses, especially when a large number of jobs (or sequences) are available. As of now, decisions are made with the initial data and it is not possible to include updates to the project's schedule or activities. This produces erroneous results because the project does not conform to the physical constraints. Overall, these lines of research are oriented to the conception of more realistic, reliable, and robust models.

2.5.2 Summary of SLR for reactive scheduling and research directions

Forty-seven articles published in 36 academic journals up to 2020 were classified and considered. The trend analysis showed that there has been a growth in the number of publications in recent years; however, these publications were distributed in a wide range of journals. It is worth noting that scientific production fell in the years from 2009 to 2013 with a very low average percentage of citations per year in this period. Our study revealed that among the different research methods based on procedures for solving reactive scheduling problems has been growing, which could be attributed to the need to provide solutions to this type of problem under uncertain events.

In this classification, we find publications according to certain methods to deal with uncertainty and their possible strategies to solve it under the reactive approach. Among the methods to deal with uncertain events we have: reactive, proactive, fuzzy, and stochastic project scheduling. Since our research is based on reactive scheduling, the solution strategies that dominate are metaheuristics, heuristics, and exact procedures. There is a strong relationship between reactive scheduling and proactive scheduling, this is given by providing schedules as protected as possible against interruptions and then, deploying procedures during the execution to revise or optimize the schedule when necessary. Forty-seven articles published in 36 academic journals up to 2020 were classified and considered. The trend analysis showed that there has been a growth in the number of publications in recent years; however, these publications were distributed in a wide range of journals. It is worth noting that scientific production fell in the years from 2009 to 2013 with a very low average percentage of citations per year in this period. Our study revealed that among the different research methods based on procedures for solving reactive scheduling problems has been growing, which could

be attributed to the need to provide solutions to this type of problem under uncertain events.

2.5.3 Joint Research Direction

From the taxonomy and SLR summarized above, it can be concluded that more research is needed to develop better and efficient strategies, methods, approaches, models to solve MRCPSP when suffering from uncertain events and apply reactive scheduling immediately. Some of the possible research directions identified for future studies are:

1. Combine exact, heuristic, and metaheuristic strategies to improve solutions of MRCPSP under uncertainty on a reactive plane.
2. Create an integrated reactive and proactive scheduling model under various objective functions.
3. Implement metaheuristic strategies to solve the multi-objective problem focused on reactive scheduling.

Chapter 3

A MEMETIC ALGORITHM FOR SOLVING THE MRCPSP

3.1 Introduction

The field of project scheduling has had a notable development in the last decades mainly because of the current competitive environment to offer quality products on time and within the budget, and the important role that it plays in the decision-making construction, transportation, distribution and communication fields. One of the most important components of project scheduling is its management, which includes basic administrative functions of planning, programming, and control. One source of motivation research is the multi-mode project scheduling problem (PSP) with resources-constrained, where activities can be executed in different ways or modes and the main goal is to minimize the duration of the project or makespan. In the literature, this is known as the multi-mode resource-constrained project scheduling problem (MRCPSP).

Generally speaking, the MRCPSP is composed of three subproblems:

P1. Finding feasible modes per non-renewable resource, provided there are more than two non-renewable resources. This is a NP-Complete problem (Kolisch, 2013).

P2. Finding a feasible solution of the RCPSP associated with any mode vector, which corresponds to a NP-Hard problem. (Blazewicz, Lenstra, and Kan, 1983).

P3. Finding the optimal solution within the set of feasible solutions, which resembles a NP-Hard problem (De Reyck et al., 1998).

Exact and heuristics strategies have been designed to solve P1, P2 and P3. However, depending on the number of activities, resource limitations and precedence relationships, this problem is difficult to solve in a reasonable period of time while providing quality solutions. Several exact strategies and heuristics are distinguished in Weglarz et al., 2011 that show some advantages when solving the MRCPSP. Unlike previous strategies, the use of metaheuristics (MS) has become the preferred method for solving the MRCPSP; some of them are presented in Table 3.1. In order to critically evaluate the research around the MS to solve the MRCPSP, we performed a systematic

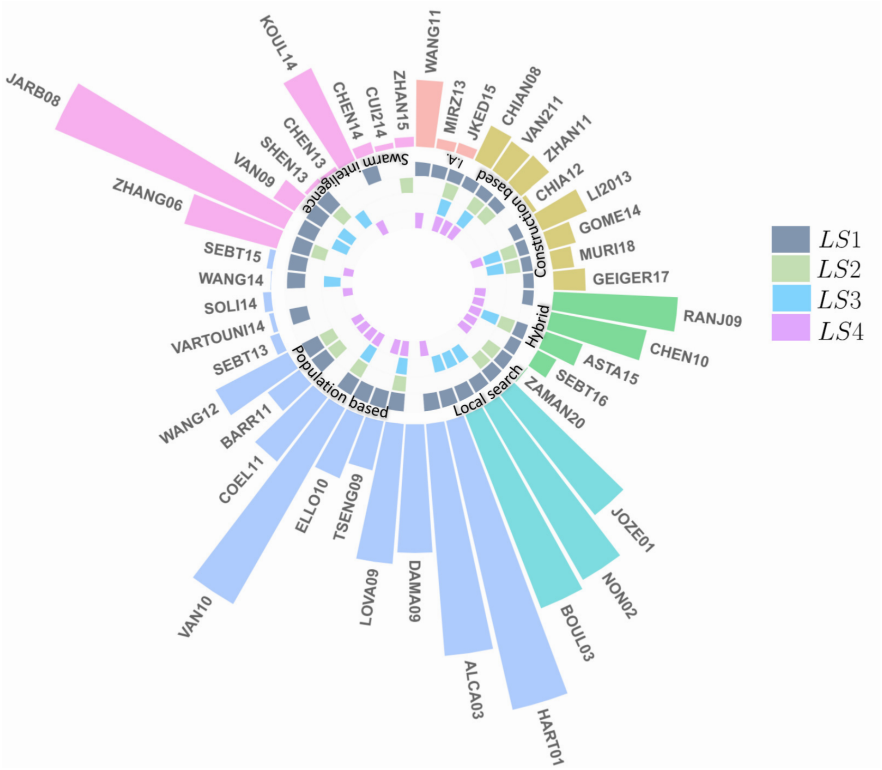


Figure 3.1: Classification of Metaheuristic Strategies for solving the MRCPSP following the criteria proposed by Van Peteghem and Vanhoucke, 2014. *LS*1: improving quality of initial population; *LS*2: improving feasibility; *LS*3: improving makespan; *LS*4: forward-backward.

review literature based on the classification criteria proposed by Van Peteghem and Vanhoucke, 2014:

1. Metaheuristic strategy: construction based, local search, population based, swarm intelligence, hybrid.
2. Schedule representation: activity-list (AL), random key (RK).
3. Mode representation: mode list (ML) and mode vector (MV).
4. Schedule generation scheme: serial (S) and parallel (P).
5. Local search procedure: improving quality of initial population (LS1), improving feasibility (LS2), improving makespan (LS3) and forward-backward (LS4).

Figure 3.1 shows a systematic analysis of MS strategies for solving the MRCPSP, considering the number of citations and the classification criteria proposed by Van Peteghem and Vanhoucke, 2014. Here we focus on solving the MRCPSP (P1-P2-P3) using a MS procedure known as Memetic Algorithm (MA). The MA, designed by Moscato et al., 1989, is defined as a MS that synergistically combines concepts from evolutionary algorithms (EA) and local search (LS). This algorithm has demonstrated to be effective for solving several problems of high computational complexity and provided exceptional

Author	Abreviation	Year	Metaheurist
Jozefowska et al., 2001	JOZE01	2001	Simulated annealing
Hartmann, 2001	HART01	2001	Genetic Algorithm
Nonobe and Ibaraki, 2002	NON02	2002	Tabu search
Alcaraz, Maroto, and Ruiz, 2003	ALCA03	2003	Genetic Algorithm
Bouleimen and Lecocq, 2003	BOUL03	2003	Simulated annealing
Zhang, Tam, and Li, 2006	ZHANG06	2006	Particle swarm optimization
Jarboui et al., 2008	JARB08	2008	Particle swarm optimization
Chiang, Huang, and Wang, 2008	CHIAN08	2008	Ant Colony Optimization
Damak et al., 2009	DAMA09	2009	Differential evolution
Lova et al., 2009	LOVA09	2009	Genetic Algorithm
Van Peteghem and Vanhoucke, 2009	VAN09	2009	Artificial immune system (AIS)
Ranjbar, De Reyck, and Kianfar, 2009	RANJ09	2009	Hybrid
Tseng and Chen, 2009	TSENG09	2009	Genetic Algorithm
Elloumi and Fortemps, 2010	ELLO10	2010	Rank-based evolutionary algorithm
Chen et al., 2010a	CHEN10	2010	Hybrid
Van Peteghem and Vanhoucke, 2010	VAN10	2010	Genetic Algorithm
Wang and Fang, 2011	WANG11	2011	Shuffled frog-leaping algorithm
Van Peteghem and Vanhoucke, 2011	VAN211	2011	Scatter search
Zhang, 2011	ZHAN11	2011	Ant Colony Optimization
Coelho and Vanhoucke, 2011	COEL11	2011	Genetic Algorithm
Barrios, Ballestin, and Valls, 2011	BARR11	2011	Genetic Algorithm
Wang and Fang, 2012	WANG12	2012	Estimation of distribution algorithm
Chiang and Huang, 2012	CHIA12	2012	Ant Colony Optimization
Sebt, Alipouri, and Alipouri, 2013	SEBT13	2013	Evolutionary programming
Mirzaei and Akbarzadeh-T, 2013	MIRZ13	2013	Multi-Agent Learning Approach
Li and Zhang, 2013	LI2013	2013	Ant Colony Optimization
Shen and Li, 2013	SHEN13	2013	Particle swarm optimization
Chen and Wang, 2013	CHEN13	2013	Particle swarm optimization
Gomes, Neves, and Souza, 2014	GOME14	2014	Greedy randomized adaptive search procedure
Koulinas, Kotsikas, and Anagnostopoulos, 2014	KOUL14	2014	Particle swarm optimization
Vartouni and Khanli, 2014	VARTOUNI14	2014	Genetic Algorithm
Soliman and Elgendi, 2014	SOLI14	2014	Estimation of distribution algorithm
Wang, Liu, and Zhou, 2014	WANG14	2014	Coevolutionary algorithms
Chen, Liang, and Padilla, 2014	CHEN14	2014	Artificial bee colony optimization
Cui and Yu, 2014	CUI214	2014	Particle swarm optimization
Asta et al., 2015	ASTA15	2015	Hybrid
Zhang, Luo, and Zhang, 2015	ZHAN15	2015	Particle swarm optimization
Jedrzejowicz and Ratajczak-Ropel, 2015	JKED15	2015	Multi-Agent Learning Approach
Sebt, Afshar, and Alipouri, 2015	SEBT15	2015	Genetic Algorithm
Sebt, Afshar, and Alipouri, 2016	SEBT16	2016	Hybrid
Muritiba, Rodrigues, and Costa, 2018	MURI18	2018	Path-relinking algorithm
Geiger, 2017	GEIGER17	2017	Variable Neighborhood search
Zaman et al., 2020	ZAMAN20	2020	Variable Neighborhood search
Chakrabortty, Abbasi, and Ryan, 2020	CHAK20	2020	Hybrid

Table 3.1: Metaheuristic strategies for solving the MRCPSP.

results for the exploration and diversification of solutions in each iteration. Furthermore, its easy implementation and adaptation to problems with linear and non-linear constraints, make MA attractive for combinatorial optimization problems.

The contributions of this paper can be summarized as follows:

1. The implementation of a variable neighborhood search (VNS) adapted to a double justification operator, and a uniform crossing operator;
2. The development of an improvement procedure that optimizes the consumption of resources and minimizes the work for the activities or their duration;
3. A robust experimental design to find suitable parameters for the MA algorithm.

The content of this document is as follows. In section 2 we conducted a systematic review literature to classify MS procedures developed between 2000 to 2020. A detailed description of the MRCPSP is given in section 3.2. In section 3.3, our approach is explained in detail. In section 3.4 the results of a computational experiment are shown. Finally, our conclusions, along with potential lines of research, are presented in section **??**.

3.2 Problem Description

Here we follow the problem description in Erik L. Demeulemeester, 2002, which simultaneously assigns the mode and the start time to each activity. The standard MRCPSP involves the selection of activities by an execution mode for each activity and the assignment of an initialization or completion time to each activity in such a way that the precedence relations are fulfilled, the resources are not exceeded and the makespan is minimized.

According to the classification in Herroelen, Demeulemeester, and De Reyck, 1999, the MRCPSP is denoted as $u, 1T|cpm, disc, \mu|C_{max}$, where u provides the number of resources; $1T$ provides the availability of renewable and non-renewable resources, both specified in a period of unit duration and a total project horizon basis; cpm regulates the final-start precedence relationships without delays as used in the basic PERT/CPM model; $disc$ corresponds to the resource requirements of the activities and are a discrete function of the duration of the activity; μ activities have multiple prespecified execution modes; and C_{max} is the objective function, which involves minimizing the duration of the project or makespan.

A project consists of a set of real activities $V = \{1, \ldots, n\}$ each of which is carried out without interruption. The dummy activities 0 and $n + 1$ are introduced to represent the start and end of the project, respectively. The set of renewable resources is denoted by R^{τ} and will be used to denote the availability (units) of the type of renewable resource $k \in R^{\tau}$. Renewable resources, available period to period, are those that can be restored at a similar or superior speed than the consumption speed. Non-renewable resources are limited to the finished duration of the project without restrictions on any time period. An example of non-renewable resource is the entire budget for the achievement of a project. The set of non-renewable resource is denoted by R^{η}, and the availability of a non-renewable resource type $\ell \in R^{\eta}$ is denoted by R^{η}_{ℓ}.

To complete the project satisfactorily, it is necessary to process (execute, sequence) each activity in one of several modes; the set of modes for an activity $i \in V$ is denoted by $M_i = \{1, 2, \dots, |M_i|\}$, where every mode $m \in M_i$ represents a different way of finishing the i-th activity and $|M_i|$ represents the total number of modes. A mode $m \in M_i$ determines the duration $d_{i,m} \geq 0$ of the activity $i \in V$, measured in number of periods or units of time, which indicates the time necessary to complete the activity, and whether it is possible to interrupt the process of that activity. For dummy activities, $d_{0,1} = d_{(n+1),1} = 0$. If the execution mode of activity $i \in V$ is $m \in M_i$, $r^{\tau}_{i,m,k} \geq 0$, where $r^{\tau}_{i,m,k} \leq R^{\tau}_{k}$ and $r^{\tau}_{0,1,k} = r^{\tau}_{n+1,1,k} = 0$ are the units of the renewable resource k required by activity i for its realization. On the other hand, $r^{\eta}_{i,m,\ell} \geq 0$ are the required units of the non-renewable resource l with $r^{\eta}_{i,m,\ell} \leq R^{\eta}_{\ell}$ and $r^{\eta}_{0,1,\ell} = r^{\eta}_{(n+1),1,\ell} = 0$.

The most widely used mathematical formulations of the MRCPSP are those presented by Pritsker, Waiters, and Wolfe, 1969; Talbot, 1982; Valdes and Goerlich, 1993; Mingozzi et al., 1998a. Most of the existing models for solving the MRCPSP are adapted versions of Talbot (1982) who introduced the notation of a 0-1 programming model with a binary decision variable x_{imt}. In this model, x_{imt}=1 if the i-th activity is performed in m mode and starts at time t, and $x_{imt} = 0$ otherwise. Here we propose, based on the start times and resource consumption of the activities, the following mathematical formulation, which constitutes a practical and simple model (Christofides, Alvarez-Valdes, and Tamarit, 1987; Neumann, Schwindt, and Zimmermann, 2012):

$$\min \quad s_{n+1} \tag{3.1}$$

$$\text{subject to} \quad s_i + d_{i,m_i} \leq s_j, \quad \forall (i,j) \in E, \tag{3.2}$$

$$\sum_{i \in \mathcal{A}(S,M,t)} r^{\tau}_{i,m_i,k} \leq R^{\tau}_{k}, \quad k \in R^{\tau}, 0 \leq t \leq \bar{d}, \tag{3.3}$$

$$\sum_{i=1}^{n} r^{\eta}_{i,m_i,\ell} \leq R^{\eta}_{\ell}, \quad \ell \in R^{\eta}, \tag{3.4}$$

$$m_i \in M_i, \quad \forall i \in V \tag{3.5}$$

$$s_i \geq 0, \quad \forall i \in V \tag{3.6}$$

$$s_0 = 0 \tag{3.7}$$

In (3), $\mathcal{A}(S, \mu, t)$, also called the active set, is the set of real activities that be sequenced in time t, $\mu = (m_i)_{i \in V}$ is the vector mode, $S = (s_i)_{i \in V}$ is the vector of start times for each activity and $\bar{d} = \sum_{j \in V} \max_{m \in M_j} d_{j,m}$ is the maximum makespan. While the objective function (1) minimizes the makespan, the constraints represented in equation (2) describes the precedence relationships between activities, equation (3) ensures that the per-period availability of the renewable resources is not violated, and equation (4) represents the restriction of non-renewable resources during the execution of the project. Restrictions (5) and (6) ensure that each activity is assigned a single mode and a single start time during the execution of the schedule. It is assumed that the start and end of dummy activities are only executed in a single zero-duration mode and do not consume resources (7).

A vector mode μ is a $n+2-$tuple $\mu = (1, \mu_1, \ldots, \mu_n, 1)$ which assigns a unique mode μ_j, $1 \leq \mu_j \leq |M_j|$m, to the j-th activity $(1 \leq j \leq n)$. A mode vector μ for which constraint (4) is satisfied is called resource feasible, and resource non-feasible otherwise. The excess of non-renewable resource for a vector mode μ is defined by

$$L^{\eta}(\mu) = \sum_{k \in \ R^{\eta}} |\min\{0, R^{\eta}_k - \sum_{j=1}^{n} r^{\eta}_{j,\mu(j),k}\}| \tag{3.8}$$

being μ a feasible resource vector mode if and only if $L^{\eta}(\mu) = 0$.

For illustration purposes, let us consider the instance J1037_2 from the PSPLIB library. This instance contains 10 real and two dummy activities. Each real activity has three execution modes and, for each mode, two renewable and two non-renewable resources are available. The availability of renewable resources is $R^{\tau}_1 = R^{\tau}_2 = 12$, and the availability of non-renewable resources is $R^{\eta}_1 = 37$ $R^{\eta}_2 = 60$. Fig. 3.2 depicts the network of activities in the nodes and Table 3.2 shows the durations for each execution mode and the requirements for renewable and non-renewable resources. The mode vector $\mu = (1,1,2,3,3,3,1,3,2,3,3,1)$ is resource feasible as $\sum_{i=0}^{11} r^{\eta}_{i,m_i,1} = 37$, $\sum_{i=0}^{11} r^{\eta}_{i,m_i,2} = 60$ and $L^{\eta}(\mu) = 0$. However, $\mu' = (1,1,3,3,3,3,1,3,2,3,3,1)$ has non-feasible resources as $\sum_{i=0}^{11} r^{\eta}_{i,m_i,1} = 38$, $\sum_{i=1}^{12} r^{\eta}_{i,m_i,2} = 63$ and $L^{\eta}(\mu') = 4$. Fig. 3.3 depicts a schedule with a makespan of 27 time periods.

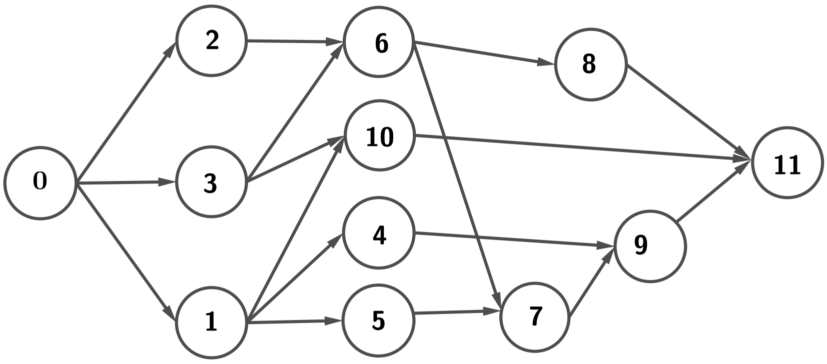


Figure 3.2: Project instance

Act i	**Mode** m_i	d_{m_i}	r^{τ}_{1,i,m_i}	r^{τ}_{2,i,m_i}	r^{η}_{1,i,m_i}	r^{η}_{2,i,m_i}
0	1	0	0	0	0	0
1	1	2	5	9	4	7
	2	5	5	8	2	7
	3	6	5	6	1	6
2	1	1	6	5	6	8
	2	8	5	4	5	5
	3	8	4	5	4	6
3	1	1	4	9	9	5
	2	8	3	4	6	1
	3	8	1	4	2	3
4	1	1	7	3	7	8
	2	2	6	3	5	8
	3	2	5	3	6	5
5	1	1	7	9	4	10
	2	4	7	7	2	10
	3	9	6	2	1	10
6	1	1	9	5	3	6
	2	1	9	4	3	7
	3	5	9	3	3	6
7	1	2	9	5	9	8
	2	2	9	6	7	8
	3	3	9	4	4	8
8	1	1	9	9	10	6
	2	5	9	8	6	6
	3	9	9	8	4	5
9	1	2	7	2	4	9
	2	9	7	2	4	6
	3	10	3	1	3	3
10	1	3	5	10	8	10
	2	6	4	10	5	7
	3	7	4	9	3	7
11	1	0	0	0	0	0
	R^{τ}_1	R^{τ}_2	R^{η}_1	R^{η}_1		
	12	12	37	60		

Table 3.2: Project information

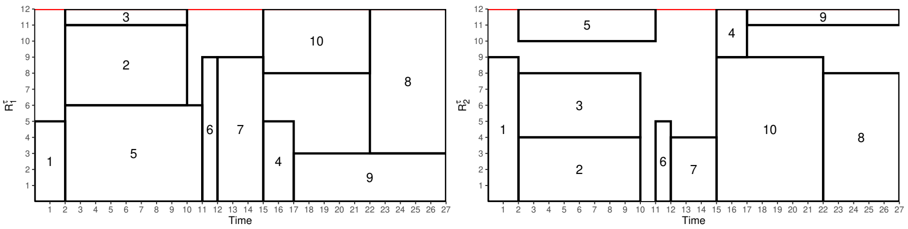


Figure 3.3: Feasible schedule MRCPSP.

3.3 Memetic Algorithm

3.3.1 Basic scheme of a MA

The MA starts by selecting an initial population POB composed of $nPob$ agents ordered according to their fitness in non-decreasing order. Each agent is denoted by $I = (\lambda, \mu)$ where μ is a mode vector and λ is a list of activities (LA). Further, a local search (LS) is applied to all members of the population in order to improve their evaluation and apply the mutation operator. Then, POB is partitioned into pairs of agents. For each pair of resulting agents, the recombination operator is applied, thus obtaining two new agents to which LS is performed to improve its evaluation. Further, the mutation operator is subsequently applied, and a new LS is performed, hence the resulting agents adhere to POB. Next, the selection operator is applied to reduce POB to its original size $nPob$ and obtain the next generation to which the same procedure is applied. This process is repeated for a pre-established number of generations, $Ngen$, or alternatively until a time limit is reached. Algorithm 1 provides a general template for a MA based on the aforementioned considerations.

3.3.2 Representation

The MA operates in the same way than a genetic algorithm (GA) with a coded representation of solutions, which is crucial for the success of a MA and the efficient use of its operators. Kolish & Hartmann (1999) distinguish five different coded representation schemes used for solving they RCPSP. Within these schemes, the LA and random key (RK) representations are worth mentioning as their use a structure based on priority rules between activities. In the LA representation, used by several authors (Bouleimen and Lecocq, 2003; Hartmann, 2001; Jozefowska et al., 2001), a schedule is represented by a precedence feasible activity list $\lambda = (0, j_1, \ldots, j_n, n+1)$ in which each activity j_i must have a higher i than each of its predecessors in $\mathcal{P}_{j_i}$, that is, $\mathcal{P}_{j_i} \subset \{j_1, j_2, j_{i-1}\}$ for $j = 1, \ldots, n$.

While in the LA representation the position of an activity is determined by the relative priority of that activity over the others, in the RK representation the sequence in which the activities are scheduled is based on the priority value attributed to each activity. Valls et al., 1999 developed the topological order representation. When activities i and j satisfy $s_i < s_j$, activity i would have a higher priority than the activity j. In this

Algorithm 1: Basic Memetic Algorithm.

```
Input  : Instance $I$ of the problem $P$, $par$ parameters of $P$
Output: $sol$
begin
    $POB \leftarrow \text{GenerateInitialPop}(par, P)$ ;
    for all $I \in POB$ do
        $I \leftarrow \text{LS}(I)$;
        $I \leftarrow \text{Mutation}(I)$;
        $POB \leftarrow POB$;
    end
    repeat
        Select two agents $I_M$, $I_F$ in $POB$;
        generate binary vector $v$;
        $I_D \leftarrow \text{Crossover}(I_M, I_F, v)$, $I_G \leftarrow \text{Crossover}(I_F, I_M, v)$;
        $I_D \leftarrow \text{Mutation}(I_D)$, $I_G \leftarrow \text{Mutation}(I_G)$;
        $I_D \leftarrow \text{LS}(I_D)$, $I_G \leftarrow \text{LS}(I_G)$;
        $newpop_1 \leftarrow$ adhere $I_D$, $I_G$;
        $POB \leftarrow \text{Compete}(POB, newpop_1)$;
        if $Converge(POB)$ then
            $POB \leftarrow \text{Restart}(POB, par)$;
        end
    until $Termination\ criterion\ (par)$;
    return GetNthBest(POB,1);
end
```

paper, however, we use the LA representation since it is equivalent to the topological order representation (Valls et al., 1999), the transformation to a sequence is efficient and, because it is a sequential process, it is easier to encode and transform it into a schedule where each individual is represented by a double list composed by a list of activities λ and a mode vector μ. Alcaraz, Maroto, and Ruiz, 2003 adds a forward/backward (f/b) gene to the LA representation, which indicates the direction in which the serial generation scheme builds the sequence. In our approach we use the two directions of the activity scheduling at random. After establishing the list of activities λ, we generate the mode vector μ and establish whether it is a feasible resource mode vector.

3.3.3 Initial Population

A fundamental part of the MA is the generation of the initial population POB of size $nPob$. Algorithm 2 is executed to create the initial population using various MS. In phase 1, Algorithm 3 is used to build $nPob$ agents. Each agent has a mode vector which is produced by the random selection of a mode construction rule; this rule selection includes the sum of durations (SOD)(Boctor, 1993), the total work content (TWC)(Van Peteghem and Vanhoucke, 2011), the opening mode-assignment (OMA)(Kolisch, 2013) and random selection (RAND). If μ is a feasible resource, an improvement step is performed in order to optimize either time or resource consumption by decreasing the work of a randomly selected activity. The work of an activity is defined as

$$work(j, \mu_j) = d_{j,\mu_j} \cdot \left(\sum_{k=1}^{|R^{\tau}|} r^{\tau}_{j,\mu_j,k} \right) \tag{3.9}$$

If μ is not a feasible resource, a LS procedure, previously used by Hartmann, 2001 in its GA, is applied to transform the mode vector into a vector of feasible modes using Algorithm 4.

To generate the LA of feasible precedence λ, we establish the modes of the activities with respect to μ and subsequently define the RCPSP associated with the vector of modes. Then, starting from an empty activity list and selecting a priority rule, a precedence feasible activity list is constructed. The priority rules considered are Latest Start Time (LST), Shortest Processing Time (SPT), Minimum Slack (MSLK), Resource Scheduling Method (RSM), Latest Finish Time (LFT), Earliest Start Time (EST), Greatest Positional Weight (GRPW), Longest Processing Time (LPT) and RAND. Next, priority rules are selected in the following order: SPT, MSLK, RSM, LFT, EST, GRPW, LPT and RAND. Each of this priority rules contributes 10% of the solutions from the initial population, making up 80% of all solutions. The last priority rule to be selected is the LST rule, which generates the remaining 20% of the solutions, to complete 100% of the solutions. The above percentages (i.e., 10% and 20%) are chosen by performing a 2-tailed Wilcoxon matched-pairs signed-ranks test for each pair of priority rules following Kolisch, 1996a. Hence, the initial population is formed with the *nPob* solutions found, thus combining quality, feasibility and diversity.

In phase 2, as agents are generated, their fitness function is calculated. Agents are subsequently attached to the population in non-decreasing order according to their fitness. Thus, the result is an initially ordered population where the first agent has the lowest fitness, and the last agent has the highest fitness.

Algorithm 2: Introduction of high quality solutions in the initial population.

```
Function: GenerateInitialPop(par Parameter, P Problem)
Output  : Poblation POB
begin
    $POB \leftarrow$ initilize;
    while $|POB| < nPob$ do
        $I \leftarrow GenerateAgent(par, P)$;
        $POB \leftarrow POB \cup \{I\}$;
    end
    return $POB$;
end
```

3.3.4 Schedule generation scheme and fitness computation

Each agent *I* is related to a certain sequence *S*, which is obtained from the adjustment of the mode vector μ. The RCPSP associated to said mode vector is searched and the generation scheme is applied in series (i.e., serial generation scheme [SGS]). SGS starts without sequencing any activities with renewable and non-renewable resources at their

Algorithm 3:

```
Function: GenerateAgent(par Parameter, P Problem)
Randomly select a rule in $\{SOD, TWC, OMA, RAND\}$;
$\mu \leftarrow rule(\cdot)$;
$\mu \leftarrow LocalSearchMode(\mu, par, P)$;
The RCPSP is built with the mode vector;
Feasible precedence list $\lambda$ is generated with one of the priority rules;
Agent $I$ is defined for $I = (\lambda, \mu)$;
Return $I = (\lambda, \mu)$
```

Algorithm 4: Feasible mode vector generator.

```
Function: LocalSearchMode(agent i,par Parameter, P Problem)
$\mu$ mode vector associated with $i$
if $L^{\eta}(\mu) = 0$ then
    the vector of modes is feasible;
    We randomly select one case;
    case 1 do
        $\mu \leftarrow ImprovementMode(\mu)$ Return $\mu$
    end
    case 2 do
        Return $\mu$
    end
else
    repeat
        Select randomly $j \in V$ with $M_j > 1$;
        select randomly $m_j \in M_j \setminus \{\mu_j\}$;
        define $\mu' = (\mu_1, \dots, m_j, \dots, \mu_n)$;
        calculate $L^{\eta}(\mu)$ y $L^{\eta}(\mu')$;
        if $L^{\eta}(\mu') \leq L^{\eta}(\mu)$ then
            $\mu = \mu'$
        else
            go to the random selection of the activity.
        end
    until $L^{\eta}(\mu) = 0$ or some stop criterion;
end
Return $\mu$
```

Algorithm 5: Mode improvement step.

```
Function: $ImprovementMode$(mode vector $\mu$)
$\mu' = copy(\mu)$
for $i = 1$ to $n$ do
    $act \in V$ is randomly selected;
    forall $j \in Suc(act)$ (set of successors) do
        forall $m \in M_{act}$ and $k \in M_j$ do
            if $work(act, m) + work(j, k) < work(act, \mu_{act}) + work(j, \mu_j)$ then
                $\mu'_{act} \leftarrow m$;
                $\mu'_j \leftarrow k$;
                if $L^\eta(\mu') = 0$ then
                    $\mu \leftarrow \mu'$
                end
            end
        end
    end
end
Return $\mu$
```

initial availability. In each iteration g, $1 \leq g \leq n$, we select the activity at position λ_g and the start time is calculated such that it complies with the constraints of renewable resources. This activity further defines a partial sequence $S_g = (s_{\lambda_1}, \dots, s_{\lambda_g})$. When all the activities have been sequenced, a complete sequence with all the start times of the activities will be available. The completion time of the sequence will be determined by the final dummy activity.

To solve the three sub-problems P1-P2-P3 that compose the MRCPSP, a (M, S) pair must be found. Here, M is a feasible mode vector and S is a feasible schedule, associated with the M mode vector, such that the makespan is minimized. Therefore, there is a schedule associated for each I when the SGS is applied. Although S is feasible in general, we allow to schedule some activity i into V considering its later start time and resource constraints only when this is not the case, i.e., $s_i > Lst_i$. Thus, we penalize those agents that do not have a feasible resource mode vector. This penalty is further used to adapt the objective function in Barrios, Ballestin, and Valls, 2011. Specifically, the fitness of an agent is defined by

$$F(I) = \begin{cases} s_{n+1}, & \text{if } L^\eta(\mu) = 0 \\ F^1(I) & \text{otherwise,} \end{cases} \tag{3.10}$$

which penalizes the excessive use of non-renewable resources.

When the mode vector is not a feasible resource, we calculate the fitness of the agent using F^1 as

$$F^1(I) = \begin{cases} s_{n+1} + F(I^*) + \sum\limits_{i \in V} (s_i - Lst_i)^+ - CP^{\min} + L^\eta(\mu) & \text{if } L^\eta(\mu^*) = 0 \\ s_{n+1} + UB(\mu) + \sum\limits_{i \in V} (s_i - Lst_i)^+ + L^\eta(\mu) & \text{otherwise,} \end{cases} \tag{3.11}$$

where $UB(\mu) = \sum_{i \in V} d_{i,\mu_i}$ is an upper bound given the mode vector of I. This function is composed by $F(I^*)$, the aptitude of the best agent of the current population with feasible mode vector, plus the increase over the project duration determined by the minimum critical path of the project $s_{n+1} - CP^{\min}$ and the excess of non-renewable resources. Since some activities may be allowed to start after their later start so that they do not violate resource restrictions, such positive difference is also considered as a penalty, that is, those activities for which $s_i > Lst_i$ are penalized. For cases in which no agent in the population has a feasible mode vector, we define the fitness as the sum of all the maximum durations of the project activities plus the excess of non-renewable resources, its date of completion in the project and the differences between s_i and Lst_i, provided that $s_i > Lst_i$. Therefore, non-feasible agents have greater fitness than doable solutions. Finally, when the population does not have any agent with a vector in a feasible way, their fitness will be greater than those having at least one agent with a vector of feasible resources.

3.3.5 Crossover

The objective of the crossover operator is to generate new agents using mainly the information extracted from the combined agents. Therefore, the selection crossover operator has a big impact on the algorithm's result when looking for quality solutions. The uniform crossover operator, which is a generalization of the two-points crossing operator for the RCPSP(Hartmann, 1999), constitutes an alternative to the crossover operator. In what follows we redefine this operator for the MRCPSP.

The uniform crossover operator starts by selecting two agents I^M and I^F from the search space $\mathcal{I}$ and a binary v vector with $n+1$ components. Then, the crossing operator $\mathcal{X}_U(I^M, I^F, v) = I^D = (\lambda^D, \mu^D)$ is applied, with $v_i \in \{0,1\}$ for $i = 1, \ldots, n+1$. Next, the list of activities λ^D of I^D is defined by

$$j_i^D = \begin{cases} j_k^M & \text{if} \quad v_i = 0 \text{ with } k = \min\{\ell \in N : j_\ell^M \notin \{j_1^D, \ldots, j_{i-1}^D\}\} \\ j_k^F & \text{if} \quad v_i = 1 \text{ with } k = \min\{\ell \in N : j_\ell^F \notin \{j_1^D, \ldots, j_{i-1}^D\}\} \end{cases}$$

when $v_{n+1} = 0$ and $i = 1, 2, \ldots, n$, and by

$$j_i^D = \begin{cases} j_k^M & \text{if} \quad v_i = 0 \text{ with } k = \max\{\ell \in N : j_\ell^M \notin \{j_1^D, \ldots, j_{i-1}^D\}\} \\ j_k^F & \text{if} \quad v_i = 1 \text{ with } k = \max\{\ell \in N : j_\ell^F \notin \{j_1^D, \ldots, j_{i-1}^D\}\} \end{cases}$$

when $v_{n+1} = 1$ and $i = n, n-1, n-2, \dots, 1$. Algorithm 6 shows this procedure. It is important to highlight that, when using this crossing operator, four different alternatives for generating agents are available, thus expanding the exploration of different regions in the search space. One way to generate a new agent is by alternating the position of I^M and I^F.

Algorithm 6: Uniform crossover operator.

```
Function: $Crossover(I^M, I^F, v)$
if $v_{n+1} = 0$ then
    for $i = 1$ to $n$ do
        if $v_i = 0$ then
            Calculate $k = \min\{\ell \in N : j^M_\ell \notin \{j^D_1, \dots, j^D_{i-1}\}\}$;
            Define $j^D_i = j^M_k$
        else
            Calculate $k = \min\{\ell \in N : j^F_\ell \notin \{j^D_1, \dots, j^D_{i-1}\}\}$;
            Define $j^D_i = j^F_k$
        end
    end
else
    for $i = n$ to $1$ do
        if $v_i = 0$ then
            Calculate $k = \max\{\ell \in N : j^M_\ell \notin \{j^D_1, \dots, j^D_{i-1}\}\}$;
            Define $j^D_i = j^M_k$
        else
            Calculate $k = \max\{\ell \in N : j^F_\ell \notin \{j^D_1, \dots, j^D_{i-1}\}\}$;
            Define $j^D_i = j^F_k$
        end
    end
end
for $i = 1$ to $n$ do
    if $v_i = 0$ then
        $\mu(j^D_i) = \mu^M(j^D_i)$
    else
        $\mu(j^D_i) = \mu^F(j^D_i)$
    end
end
Return $I^D$
```

3.3.6 Mutation

The mutation operator is applied in two stages: after applying LS to the population POB and when an agent is generated by the crossing operator (Algorithm 7). In our approach, we use an adaptation to the mutation operator used by Lova et al., 2009 for solving the MRCPSP. Once the crossover operator has been applied and the population of generated agents has replaced the original population, the mutation operator is applied again to the agent population. The mutation alters one or more genes (positions) of a selected chromosome, that is, reintroduce genetically lost material and thus have

Algorithm 7: Mutation operator.

```
Function: $Mutation(I)$
Select randomly $i$ in $\{1, \ldots, n\}$;
if $RND \leq P_{mut}$ then
    Calculate $j_\mu = \max \mathcal{P}(j_i)$ and $j_\nu = \min \mathcal{S}(j_i)$;
    select $j_r \in (j_\mu, j_\nu)$;
    $\hat{j}_r = j_i$;
    $\hat{j}_i = j_r$
else
    $\hat{j}_i = j_i$
end
if $\mu$ it is a feasible resource then
    for $i = 1$ to $n$ do
        $\hat{\mu}(j_i) = \mu(j_i)$
    end
else
    $fixrec(\mu)$
end
Return $\hat{I}$
```

some extra variability in the population. In fact, the mutation can lead to completely new gene values, which sometimes allow the MA to arrive at better solutions than previous versions of the GA. In addition, the mutation helps to prevent the stagnation of the population in any local optimum.

The mutation applies to both components of each individual representation (i.e., the LA and the modes vector). In the first case, the mutation procedure used is the insertion procedure, which works as follows. For each activity in the LA, a new position is chosen at random between the highest positions of its predecessors and the lowest position of its successors. The activity is then inserted in the new position with probability P_{mut}. Regarding the mode allocation list, the application of the mutation operator changes depending on whether or not the individual has a vector of feasible resource modes. Hence, if the agent has a feasible mode vector, that mode vector is left. Conversely, if the mode vector is not feasible, we use the function $fixrec(\mu)$. This function is responsible for leveling the consumption of non-renewable resource by changing the mode for the activity that consumes the most resource.

3.3.7 Variable Neighborhoods Search for Double Justification Multiple Mode

The local search procedure used here is the variable neighborhoods search (VNS) described in Hansen et al., 2010. The VNS consists of changing the structure of neighborhoods systematically in order to continue the search after finding a local optimum. The procedure implemented in VNS uses only two neighborhoods. In our implementation, the function $GetNeighbor$ randomly generates a new agent I' from a neighborhood

(Algorithm 8). This neighborhood is defined by two types of movements: insertion forward or $\mathcal{N}_1$, and insertion backward or $\mathcal{N}_2$, described as follows:

$\mathcal{N}_1$ movement: Given an agent $I \in \mathcal{I}$, the insertion forward defines $\mathcal{N}_1(I)$ by randomly selecting $\lambda, \mu \in \{1, 2, \ldots, n-1\}$ with $\mu < \lambda$. Then, the new agent will be generated by placing the activity i_λ in the position μ, while activities in intermediate positions move a position towards the right to guarantee practicality. In addition, there is no precedence relation $i_v \rightarrow i_\lambda$ for $v \in \{\mu, \ldots, \lambda - 1\}$.

$\mathcal{N}_2$ movement: Given an agent $I \in \mathcal{I}$, the insertion backwards defines $\mathcal{N}_2(I)$ by randomly selecting $\lambda, \mu \in \{1, 2, \ldots, n-1\}$ with $\lambda < \mu$. Then, the new agent will be generated having placed the activity i_λ in the position μ, while activities in the intermediate positions move a position to the left to guarantee the feasibility. In addition, there is no relation of precedence $i_\lambda \rightarrow i_v$ for $v \in \{\lambda + 1, \ldots, \mu\}$.

Algorithm 8: VNS algorithm.

```
Function: $VNS(I)$
$k := 1$ ;
repeat
    $I' \leftarrow GetNeighbor(I, k)$;
    $I'' \leftarrow MMDJ_{max}(I', k)$ ;
    $ChangeNeighborhood(I, I'', k)$;
until $k = 2$ or a prefixed number of times;
Return $I$
```

The $MMDJ_{max}$ operator, which is a fundamental part of Algorithm 8, is described in Algorithm 9. This implementation is based on the justification operator by Tormos and Lova, 2001, which is undoubtedly a powerful tool to obtain better quality solutions. Barrios, Ballestin, and Valls, 2011 used an operator called double multiple mode justification (MMDJmax).

We adapted these justification operators to the VNS algorithm to improve its solutions. Given an agent $I \in V$, $k = 1$ and its respective S schedule obtained when applying a SGS, activities are ordered in increasing numbers according to their starting time, in order to select those activities with the earlier starting times. Further, each activity, including its mode and start time, is eliminated from the schedule. All the remaining modes of the activity are evaluated, and new start date is determined such that the new start time is the same or less than the previous, and the end date is earlier than the previous, as long as the restrictions of renewable and non-renewable resources are not violated. In this way, each activity provides a search dimension depending on the number of modes. Then, the mode that allows the activity to be moved further forward is selected. This technique simultaneously modifies the LA and the mode vector. When $k = 2$, the procedure is similar, but activities are ordered in decreasing form depending on their completion times.

Function *ChangeNeighborhood* in Algorithm 8 compares the values of $F(I)$ with the new value $F(I'')$ obtained from the $k-$th neighborhood. If an improvement is obtained, the new data is updated and k is returned to its initial value. If not, the next

Algorithm 9: MMDJmax algorithm.

```
Function: $MMDJmax(I,k)$
for $s = 1$ until $n$ do
    $i = v(s)$;
    for $h = 1$ until $|M_i|$ do
        Build $M'$: $M'(j) = M(j)\ \forall j \neq i$, $M'(i) = h$;
        if $M'$ is feasible resource then
            Calculate $ES_{i_h}$ and $LS_{i_h}$;
            if $ES_{i_h} < LS_{i_h}$ then
                Calculate the highest $t \in [ES_{i_h}, LS_{i_h}]$ where $i$ can be sequenced in
                  $[t, t + d_{i,h}]$
            end
        end
    end
end
Return $I$
```

neighborhood is considered. The neighborhood change can be performed deterministically, stochastically or using both approaches simultaneously.

3.4 Computational Experiments

Here we show the results of our MA in detail. In section 3.4.1, we conduct an experimental design to determine the potential interaction between the parameters of our MA algorithm. Section 4.4.3 reports the results of our MA algorithm when different instances are used. Finally, in section 3.4.3, we compare the results of our MA with different procedures available in the literature for solving the MRCPSP.

3.4.1 Experimental Design

In order to assess whether the parameters of the MA affect its performance, computational experiments were conducted using a random sample of the set of instances of 50 activities of the MMLIB library (i.e., MMLIB50). An orthogonal matrix was used to design a factorial experiment, and the effect of five factors and their interactions on the response variable were examined. Controlled parameters of the MA included the initial population size (nPob), the crossing probability (Pcross), the mutation probability (Pmut), the number of generations (Ngen), and the use of the VNS operator (Vns). With the exception of the Vns factor that have two levels (0: No, 1: Yes), all parameters were set to have three levels (Table 3.3). For the MMLIB50 instance, the response variable is %Dev, that is, the average deviation from the critical path lower bound.

An L_{18} orthogonal design was chosen;values of $\%Dev$ obtained with our MA are shown in Table 3.4. Statistical analysis of the experiment using ANOVA revealed that factors Vns, nGen, pCross, and nPob have, overall, the higher significantly significant effects on $\%Dev$ (Table 4.4); the percentage contribution to the mean squared is 52.67%,

MA parameter	nPob	pCross	pMut	nGgen	Vns
Level 1	20	0.2	0.1	20	0 (No)
Level 2	60	0.6	0.2	50	1 (Yes)
Level 3	100	0.8	0.3	80	-

Table 3.3: Levels of the parameters of the MA used in the computational experiments.

	Factors					
Trial No.	nPob	pCross	pMut	nGen	Vns	%Dev
1	20	0.8	0.1	50	1	0.33
2	100	0.2	0.2	50	0	0.389
3	60	0.8	0.1	50	0	0.372
4	20	0.5	0.2	80	1	0.343
5	60	0.5	0.2	20	0	0.4
6	60	0.8	0.3	80	1	0.328
7	100	0.5	0.3	50	1	0.333
8	20	0.2	0.3	20	1	0.39
9	100	0.2	0.1	80	1	0.358
10	100	0.8	0.2	20	1	0.393
11	60	0.2	0.3	80	0	0.383
12	20	0.8	0.2	80	0	0.377
13	60	0.5	0.1	20	1	0.345
14	20	0.5	0.3	50	0	0.4
15	100	0.5	0.1	80	0	0.369
16	100	0.8	0.3	20	0	0.378
17	60	0.2	0.2	50	1	0.348
18	20	0.2	0.1	20	0	0.466

Table 3.4: L_{18} orthogonal arrays with the response variable.

15.12%, 14.45%, and 11.6%, respectively. Although the factor pMut is not a statistically significant contributing factor to $\%Dev$, it was not removed from the model since it can introduce new information to the agents. Now, in order to avoid incorrectly selecting the model parameters and minimize Type I or Type II errors, P-values were corrected for multiple testing using the FDR.

Table 4.5 shows $\%Dev$ for each level of factors in the experimental design (see Table 3.4 for more details), with Δ representing the maximum difference between the level responses. Further, these differences were ranked from 1 (minimum) to 5 (maximum) to reflect the maximum difference in performance. Profile plots for $\%Dev$ by all factors are shown in Fig. 3.4. These plots help to graphically identify the best levels for the experimental factors and provide an excellent guide for the adjustment of the MA. We identified that the optimum set up conditions for the proposed MA are $nPob = 100$, $pCross = 0.8$, $pMut = 0.2$ and $nGen = 80$, while applying the VNS strategy.

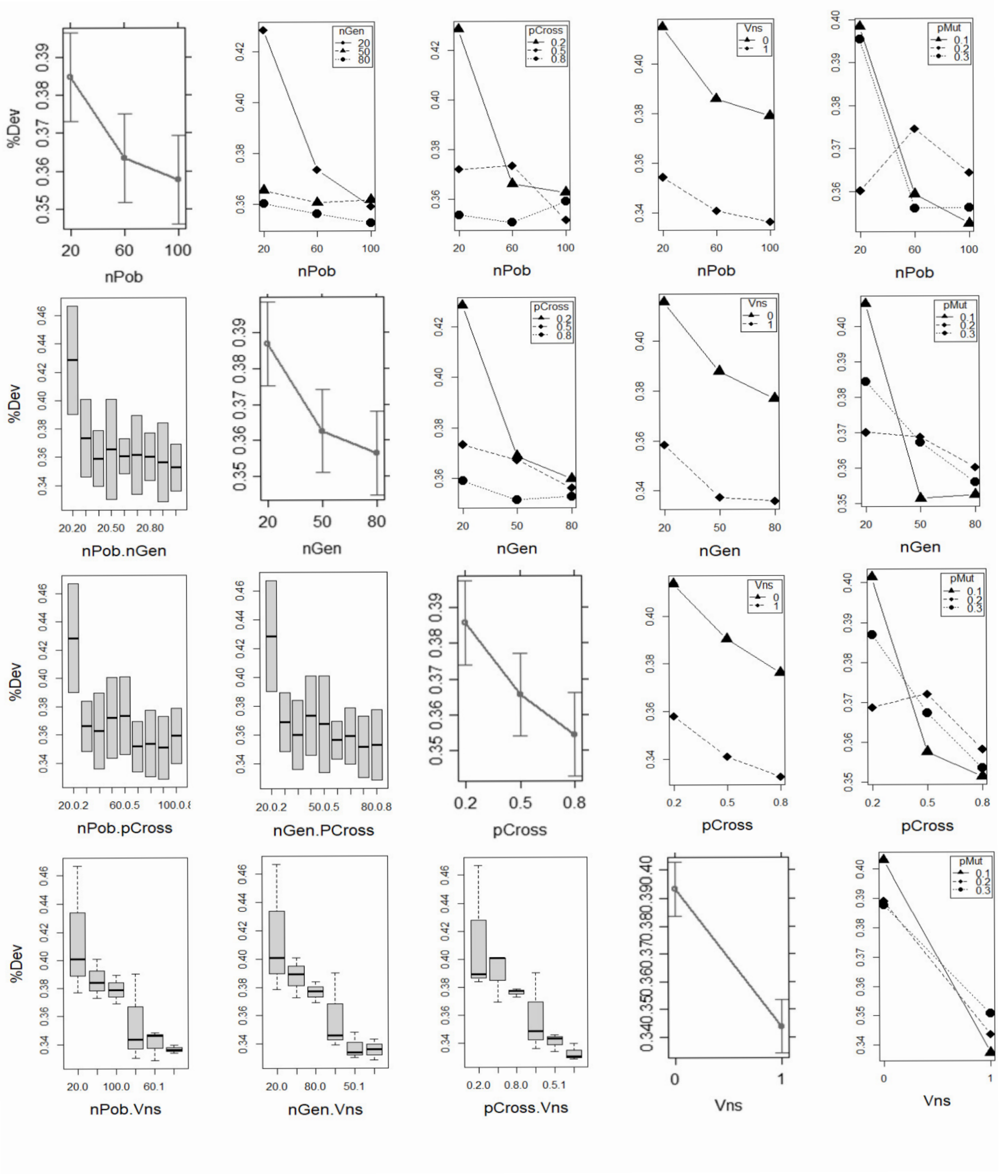


Figure 3.4: Boxplots and profile plots for %*Dev* as a function of the MA parameters.

Factor	Df.	Mean Square	F-value	C(%)	P-Value	P-Adjust
nPob	2	0.001	7.952	11.660	**0.013**	**0.016**
pCross	2	0.002	9.859	14.457	**0.007**	**0.012**
pMut	2	0.000	0.150	0.220	0.863	0.863
nGen	2	0.002	10.31	15.121	**0.006**	**0.012**
Vns	1	0.011	71.85	52.677	**0.000**	**0.000**
Residuals	8	0.000		5.865		

Residual standard error: 0.01236 on 8 degrees of freedom.
Multiple R-squared: 0.9413, Adjusted R-squared: 0.8754.
F-statistic: 14.27 on 9 and 8 DF, p-value: 0.000503

Table 3.5: ANOVA results. Here, C(%) represents the contribution of each factor and P-Adjust the associated P-value corrected for multiple testing using the False Discovery Rate (FDR) (Benjamini and Hochberg, 1995).

	Factors				
	nPob	**pCross**	**pMut**	**nGen**	**Vns**
Level 1	0.3846	0.3856	0.3700	0.3869	0.3932
Level 2	0.3633	0.3656	**0.3663**	0.3625	**0.3438**
Level 3	**0.3576**	**0.3543**	0.3692	**0.3562**	
Δ	0.0269	0.0312	0.0037	0.0306	**0.0494**
Rank	2	4	1	3	**5**

Table 3.6: Ranking of factors by level effects.

3.4.2 Results of the MA

In this section, we assess the performance of our MA algorithm under several instances available in the literature. The MA has been coded and compiled in C++ and tested on a Toshiba laptop computer with an Intel Core i5 2.3GHz processor. Computer tests were performed using the PSPLIB and MMLIB libraries created by the ProGen (Kolisch and Sprecher, 1996) and Rangen1 (Van Peteghem and Vanhoucke, 2014) generators. The PSPLIB library contains, for each subset of instances, 640 instances with two renewable and non-renewable resources, and 10, 12, 14, 16, 18, 20, and 30 activities. However, there are instances that are not feasible, i.e., not all of these instances can be resolved. Thus, all non-feasible instances will be excluded from our research.

Table 3.7 lists all instances with optimal results or at least a viable solution found for different subset instances of the PSPLIB and MMLIB libraries. For example, the MMLIB library contains three subsets of instances: the sets MMLIB50 and MMLIB100 have 540 instances with 50 and 100 activities, respectively. Each project activity has two renewable resources, two non-renewable resources, and three modes per activity. To date, 220 optimal solutions have been found for the instances in MMMLIB50 (Chakrabortty, Abbasi, and Ryan, 2020) and 236 optimal for those in MMLIB100. The MMLIB+ set, on the other hand, has a total of 3240 instances, between 2-4 renewable resources, and 2-4 non-renewable resources with 3, 6, and 9 modes per activity. A total of 373 optimal solutions have been found for this set (Chakrabortty, Abbasi, and Ryan, 2020).

Dataset	Number of instances	Number of activities	Number of renewable resources	Number of nonrenewable resources	Execution modes per activity
J10	536	10	2	2	3
J12	547	12	2	2	3
J14	551	14	2	2	3
J16	550	16	2	2	3
J18	552	18	2	2	3
J20	554	20	2	2	3
J30	552	30	2	2	3
MMLIB50	540	50	2	2	3
MMLIB100	540	100	2	2	3
MMLIB+	3240	50-100	2,4	2,4	3,6,9

Table 3.7: Parameter settings of the test instances of the PSPLIB and MMLIB libraries

Our main results are presented in Table 3.8; column $\#Act$ shows the number of activities, $\#Feas$ the number of feasible instances, and $\#Opt$ the number of optimal solutions found. We calculated the average deviation, $\%Dev$, considering 1000 and 5000 schedules with fixed parameters for the J10, J12, J14, J16, J18, J20 and J30 instances of the PSPLIB library, and the MMLIB50, MMLIB100 and MMLIB+ instances from the MMLIB library. For the J30, MMLIB50, MMLIB100 and MMLIB+ instances, we computed $\%Dev_{cpm}$, defined as the average deviation from the critical path lower bound (LB_{cpm}). The $\%Opt$ column corresponds to the percentage of optimum solutions found for the J10, J12, J14, J16, J18 and J20 instances, or equal to the LB_{cpm} for J30.For the MMLIB50, MMLIB100 and MMLIB+ instances, $\%Opt$ is obtained by comparing the solutions provided by our MA algorithm with the best-known lower bounds. In addition, we considered a time limit of 0.1 and 0.15 seconds per activity (s/act), and calculated 1000 and 5000 schedules using the criteria proposed by Van Peteghem and Vanhoucke, 2011.

Our MA algorithm behaves very well for all instances of the PSPLIB library. However, the increase of $\%Dev$ is discussed because of the influence of the MA parameters in the MMLIB instances. To investigate this pattern further, we performed a multivariate linear regression analysis with using the order strength (OS), the renewable and nonrenewable resource strength (RS^{τ} and RS^{η}), the number of modes ($|M|$), the renewable resource factor (RF^{τ}), and the number of renewable (($|R^{\tau}|$) and nonrenewable ($|R^{\eta}|$) resources as the independent the factors/variables. The response variable is the average deviation from the critical path lower bound with 5000 calculated schedules. Results are presented in Table 3.9. For the MMLIB50 and MMLIB100 set of instances, the the proportion of variance in $\%Dev$ explained by the independent variables in the model is low; these values are 51% and 52%, respectively. A different behavior occurs in the MMLIB+ instance where this percentage is 72%. We found that the OS is not

		Stopping Criterion								
		1000 Schedules		**5000 Schedules**		**0.1 s/act**		**0.15 s/act**		
Instance	*#Feas*	*#Opt*	*%Dev*	*#Opt*	*%Dev*	*#Opt*	*%Dev*	*#Opt*	*%Dev*	*%Opt*
J10	536	535	0.004	536	0	536	0	536	0	100
J12	547	537	0.085	547	0	537	0.087	547	0	100
J14	551	537	0.094	548	0.02	537	0.1	547	0.02	99
J16	550	518	0.22	543	0.05	525	0.18	541	0.06	99
J18	552	500	0.36	526	0.19	514	0.25	528	0.16	97
J20	554	496	0.41	524	0.22	508	0.33	531	0.16	96
J30*	552	-	13.76	-	12.74	-	13.66	-	12.7	54
MMLIB50*	540	-	27.30	-	26.76	-	26.82	-	26.56	40
MMLIB100*	540	-	33.23	-	29.14	-	31.61	-	30.26	44
MMLIB+*	3240	-	90.18	-	88.05	-	92.78	-	91.25	11

*average % deviation from the critical path length, $\%Dev_{cpm}$

Table 3.8: Performance of our MA for instances of the PSLIB and MMLIB libraries.

Dataset	R^2	Const.	OS	RF^τ	RS^τ	$\|M\|$	RS^η	$\|R^\eta\|$	$\|R^\tau\|$
MMLIB50	0.51	**0.83**	0.01	**0.25**	**-0.54**	-	**-0.97**	-	-
MMLIB100	0.52	**0.71**	0.06	**0.45**	**-0.64**	-	**-0.95**	-	-
MMLIB+	0.72	**1.32**	0.01	-	**-1.33**	**0.19**	**-2.12**	**0.04**	**0.03**

Table 3.9: Results of the multivariate regression on the MMLIB library.

significant for any of the instances, while RF^τ is significant and has a strong positive effect on $\%Dev$ for both MMLIB50 and MMLIB100. Interestingly, increasing RS^τ and RS^η decreases $\%Dev$ for all set of instances, that is, lower values of RS^τ and RS^η significantly improve the performance of our MA algorithm. For MMLIB+, increasing $|M|$, $|R^\tau|$ and $|R^\eta|$ increases $\%Dev$. This behavior, however, is expected as changing these parameters increases the complexity of the MRCPSP. Table 3.10 summarizes the results of our MA with respect to the OS, RS^τ , RS^η and RF^τ parameters. Overall, our results indicate that execution time of the MA is acceptable and the quality of the sequences is quite good.

3.4.3 Comparison with other Metaheuristics

In this section, we compared the results provided by our MA algorithm and other procedures available in the literature (see, for instance, Van Peteghem and Vanhoucke, 2014) when 5000 schedules are generated and the $\%Dev$ is calculated. For the J30 sets of instances of PSPLIB library, and the MMLIB50, MMLIB100, and MMLIB+ instances of MMLIB library, $\%Dev_{cpm}$ is used.

Tables 3.11 and 3.12 summarize our results. Overall, our MA outperforms all approaches in the J10, J12, J14, and J30, and competes reasonably well for the remaining

Measure	Parameter	MMLIB50		MMLIB100		MMLIB +	
		1000	5000	1000	5000	1000	5000
OS	0.25	29.93	26.73	29.2	25.84	88.00	86.62
	0.5	31.44	27.81	36.45	32.51	91.31	89.03
	0.75	31.08	27.37	34.06	29.39	100.13	97.05
RF^{τ}	0.5	23.56	21.02	22.66	19.36	-	-
	0.75	30.95	27.54	32.09	28.17	-	-
	1.00	37.82	33.13	46.09	41.27	90.18	88.05
RS^{τ}	0.25	48.39	44.28	54.15	49.98	134.73	133.43
	0.5	23.82	20.53	23.43	19.88	72.20	70.08
	0.75	20.24	17.11	22.12	17.87	65.24	62.76
	1.0	-	-	-	-	44.53	42.70
RS^{η}	0.25	62.94	57.07	65.64	58.35	145.2	142.33
	0.5	19.51	16.18	22.05	18.32	85.19	82.73
	0.75	10.01	8.67	12.02	11.07	39.93	38.11

Table 3.10: $\%Dev$ for three instances of the MMLIB library for 1000 and 5000 schedules when the proposed MA algorithm is used.

set of instances of the PSPLIB library. For the MMLIB50 and MMLIB100 sets of the MMLIB library, PA18 and VANP11 are slightly better than our MA implementation, whereas for the MMLIB+ set, CHAK20 outperforms PA18 and our proposal.

	PSPLIB Library							
Method	J10	J12	J14	J16	J18	J20	J30	Average Rank
PR (PA18)	0.01	0	0.05	0.03	0.09	0.06	12.85	1.87
MA (**This work**)	0	0	0.02	0.05	0.19	0.22	12.74	1.89
EA (VANP11)	0	0.02	0.08	0.15	0.23	0.32	13.66	2.07
GA (VANP10)	0.01	0.09	0.22	0.32	0.42	0.57	13.75	2.2
MV (CHAK20)	0.15	0.62	0.81	0.15	0.89	0.27	13.63	2.36
GA (COEL11)	0.07	0.16	0.3	0.48	0.56	0.8	14.44	2.4
GA (LOVA09)	0.06	0.17	0.32	0.44	0.63	0.87	14.77	2.47
DEA(DAMA09)	0.09	0.11	0.34	0.42	0.59	1.62	15.43	2.66
EA (ELLO10)	0.12	0.24	0.8	1.14	1.53	0.91	13.91	2.66
EOD(WANG12)	0.12	0.14	0.43	0.59	0.9	1.28	15.55	2.72
GA (HART01)	0.06	0.14	0.44	0.59	0.99	1.21	16.93	2.91
PSO (JARB08)	0.03	0.09	0.36	0.44	0.89	1.1	18.14	3.01
EA (RANJ09)	0.18	0.65	0.89	0.95	1.21	1.64	16.21	3.1
GA (TSEN09)	0.33	0.52	0.93	1.08	1.32	1.69	17.06	3.28
PSO (ZHANG06)	0.11	0.17	0.41	0.83	1.33	1.79	18.63	3.32
GA (ALCA03)	0.24	0.73	1	1.12	1.43	1.91	21.85	4.04
SA (BOUL03)	0.21	0.19	0.92	1.43	1.85	2.1	99.64	15.19

Table 3.11: Comparison of the average deviation of different algorithms for several instances. For J30, the average deviation from the critical path, $\%Dev_{cpm}$, is calculated.

In order to determine whether there are significant differences between procedures for solving the MRCPSP, we performed a Kruskal-Wallis (KW) test. For the PSPLIB

	MMLIB Library			
Method	MMLIB50	MMLIB100	MMLIB+	Average Rank
PR (PA18)	24.24	25.18	86.84	45.42
MA (**This work**)	26.76	29.14	88.05	47.98
MV (CHAK20)	35.6	65.7	48.37	49.89
EA (VANP11)	25.45	26.51	101.45	51.14
GA (LOVA09)	28.59	31.01	114.07	57.89
DEA (DAMA09)	32.46	36.87	126.69	65.34
GA (HART01)	30.61	33.98	132.01	65.53
EA (RANJ09)	32.16	37	131.06	66.74
EA (ELLO10)	32.47	40.22	130.06	67.58
EOD (WANG12)	31.95	38.55	144.84	71.78
GA (ALCA03)	43.05	52.67	177.55	91.09
GA (TSEN09)	37.92	66.04	183.02	95.06

Table 3.12: Comparison of the average deviation of different algorithms for several instances. For MMLIB50 and MMLIB100, the average deviation from the critical path was calculated.

library, we selected the four most important metaheuristic procedures that have dominated the literature in the last 10 years (i.e., PA18, CHAK20, VANP11, VANP10) and the MA proposed in this study, and tested whether any of the selected algorithms produced statistically significant differences in the percentage of deviation changes. Interestingly, the KW test provided no evidence of significant differences between the algorithms compared to solve the MRCPSP ($\chi^2_4 = 8.201$, $P = 0.084$). Similarly, we considered all the procedures described above for the MMLIB library and observed no significant statistical difference ($\chi^2_1 = 9.10$, $P = 0.611$). In summary, these results indicate that our proposal performs equally well then the most important metaheuristic procedures and can be considered a plausible and feasible alternative to solve MRCPSP in the selected instances.

3.5 Chapter Summary

The multi-mode resource-constrained project scheduling problem (MRCPSP) is a very general scheduling model. The MRCPSP covers problems where activities can be executed in several ways or modes, and are affected by parameters such as their duration, temporary relationships with other activities, and renewable and non-renewable resource requirements. The objective of the MRCPSP is to select a combination of time/resources to minimize the duration of the project and completing all activities while satisfying all resource constraints and precedence relationships. Here we describe a Memetic Algorithm to solve the MRCPSP. This algorithm uses the components of genetic algorithms and variable neighborhoods search to implement (1) an adaptation of the uniform crossover operator, and (2) a local search to assess agents' performance that appropriately guide the evolution of the algorithm and hence generate better solutions. We implement a metaheuristic strategy and compare its performance for solving different instances of the standard PSPLIB and MMLIB libraries. Overall, our Memetic Algorithm provides suitable solutions for the MRCPSP and shows outstanding performance in all tested instances.

Chapter 4

AN ADAPTATIVE BACTERIAL FORAGING OPTIMIZATION ALGORITHM FOR SOLVING THE MRCPSP WITH DISCOUNTED CASH FLOWS

4.1 Introduction

The bacterial foraging optimization (BFO) algorithm (Passino, 2002) is inspired by the group behavior of foraging bacteria such as *E.coli* and *M.xanthus*. The algorithm imitates how bacteria feed in a landscape of nutrients to perform a non-aggressive parallel optimization, perceiving chemical gradients in the environment (such as nutrients) and moving towards or away from specific signals. The BFO algorithm belongs to the natural-inspired algorithm class (Bio-Inspired Computing) (Kar, 2016), which is applied in scheduling (Zhengwei, Pang, and Wang, 2011), data clustering (Wan et al., 2012), prediction of the stock market (Majhi et al., 2009), economic load dispatch (Saber and Venayagamoorthy, 2008), dynamic economic dispatch (Vaisakh, Praveena, and Rao, 2010), forecasting using neural networks (Zang, He, and Ye, 2010) and a job-shop scheduling problem (Ge and Tan, 2012). Furthermore, Chen, Zhu, and Hu, 2011 propose a BFO algorithm based on artificial bacteria that are capable of self-adapting their exploration and exploitation behaviors in the foraging process. The proposal uses a crossing technique of a genetic algorithm (GA) to improve the evaluation of the objective function (Panda and Naik, 2012). The mutation operator is then applied to change the structure and position of the bacteria with reference to the nutrient-rich areas and to reintroduce the genetic material lost in the colony and create a variation in the bacteria.

The multi-mode resource-constrained project scheduling problem (MRCPSP) is an important practical problem in both project management and combinatorial optimization. The MRCPSP is NP-hard and, if there is more than one nonrenewable resource, then finding a feasible solution for the MRCPSP is NP-complete (Kolisch and Drexl, 1997). The classical MRCPSP consists of selecting an execution mode for each activity and the assignment of an initialization or completion time of the activities under resource and precedence restrictions with the objective of minimizing the makespan. Some variants of the MRCPSP with respect to the measure performance include maximizing the project's net present value (NPV), minimizing the resource availability cost, or minimizing project associated costs (Weglarz et al., 2011). According to Herroelen, Van Dommelen, and Demeulemeester, 1997, there are several payment models such as the Lump-sum payment (LSP), payments at activities' completion times (PAC), equal time intervals (ETI) and progress payment (PP). The existing metaheuristics to solve the MRCPSP using the NPV as the objective function are GA (Ulusoy, Sivrikaya-Şerifoğlu, and Şahin, 2001; Leyman and Vanhoucke, 2016; Aboutalebi, Najafi, and Ghorashi, 2012), simulated annealing (SA) (Dayanand and Padman, 2001; Mika, Waligóra, and Weglarz, 2005; Delgoshaei et al., 2014), memetic algorithms (MA) (Chen and Chyu, 2008), ant colony optimization (ACO) (Chen et al., 2010b) and Tabu search (TS) (Icmeli and Erenguc, 1994). However, the use of metaheuristic strategies for solving the MRCPSP based on bacteria's behavior is nonexistent.

This paper proposes an adaptative bacteria foraging optimization (ABFO) algorithm to resolve the MRCPSP with discount cash flows (MRCPSPDC), defined as the sum between discounted cash inflows and outflows. We apply the PAC payment model, where the client pays the contractor for the completion of each activity of the project. The ABFO is characterized by implementing several operators, including the chemotaxis operator based on the double-justified operator to explore and exploit feasible regions; the swarm operator that transmits the information of the best bacteria to the rest of the bacteria in the colony; and the crossover and mutation operators to the ABFO algorithm. The parameters for the ABFO implementation are obtained using a robust Taguchy experimental design. Computational experiments show that our ABFO algorithm provides very good solutions for the MRCPSP under the objective of maximizing NPV.

The remainder of the paper is organized as follows: Section 4.2 describes the MRCPSPDC formulation; section 4.3 shows our ABFO in details and in section 4.4 the results of the computational experiments are reported.

4.2 Problem description

The MRCPSPDC involves the selection of activities by an execution mode for each activity and the assignment of an initialization or completion time of the activities in such a way that the precedence relations are fulfilled without violating the deadlines, the resources are not exceeded, and the NPV of the project is maximized. The MRCPSPDC problem is denoted as $m, 1T|cpm, \delta_n, disc, \mu, c_i|npv$ when the PAC payment function is considered. Following the classification in Herroelen, Demeulemeester, and De Reyck, 1999, m provides the number of resources; $1T$ provides the availability of renewable and non-renewable resources, whose availability is specified both in a unit duration period and in a total project horizon basis; *cpm* regulates the final-start precedence relationships without delays as used in the basic PERT/CPM model; δ_n a project deadline is imposed; *disc* corresponds to the resource requirements of the activities and is a discrete function of the duration of the activity; μ activities have multiple prespecified execution modes; c_i activities have an associated arbitrary cash flow, and *npv* is the objective function, which involves maximizing the NPV of the project.

Formally, let $V = \{0, 1, 2, \dots n+1\}$ be the set of activities of a project, where $1, \dots, n$ are real activities, each of which is conducted without interruption. The dummy activities 0 and $n+1$ are introduced to represent the start and end of the project, respectively. Consider the acyclic graph $G = (V, E)$, where V denotes the set of nodes and E the set corresponds to a finish-start, zero-lag precedence relationship between pair of activities. Precedence relationships between activities are defined by $(i, j) \in E$, which denotes that activity i is an immediate predecessor of activity j, implying that activity j may not start before activity i is completed. To complete the project satisfactorily, it is necessary to process (execute, sequence) each activity in one of several modes; the set of modes for an activity $i \in V$ is denoted by M_i, where every mode $m \in M_i$ represents a different way of realizing the activity i. A mode $m \in M_i$ determines the duration $d_{i,m} \geq 0$ of the activity $i \in V$, measured in number of periods or units of time, which indicates the time necessary to complete the activity. For dummy activities, $d_{0,1} = d_{(n+1),1} = 0$. Furthermore, for the execution of each activity, renewable and non-renewable resources are needed. The set of renewable resources is denoted by R^τ and will be used to denote the availability (in units) of the type of renewable resource $k \in R^\tau$. Renewable resources, available period to period, are those that can be restored at a similar or superior speed than the consumption speed. Non-renewable resources are limited to the finished duration of the project, without restrictions on any time period. The set of non-renewable resources is denoted by R^η. The availability of a non-renewable resource type $\ell \in R^\eta$ is denoted by R^η_ℓ. In addition, we assume $\delta_n > 0$ to be the deadline for completing the project. Moreover, each activity is associated with a cash in- and outflow, such that $c_{i,in} > 0$ and $c_{i,out} < 0$, respectively, the discount rate is $\alpha > 0$ and f_i is completion time of the activity $i \in V$.

We use the following mathematical formulation, which is based on the PAC payment model (Vanhoucke, Demeulemeester, and Herroelen, 2001; Leyman and Vanhoucke, 2016):

$$\text{Maximize} \quad \sum_{i=1}^{n}(c_{i,in} + c_{i,out}) \cdot e^{-\alpha f_i} \tag{4.1}$$

$$\text{subject to} \quad f_i \leq f_j - d_{j,m_j}, \quad \forall(i,j) \in E, \tag{4.2}$$

$$\sum_{i \in \mathcal{A}(F,M,t)} r^{\tau}_{i,m_i,k} \leq R^{\tau}_{k}, \quad k \in R^{\tau}, 0 \leq t \leq \delta_{n+1}, \tag{4.3}$$

$$\sum_{i=1}^{n} r^{\eta}_{i,m_i,\ell} \leq R^{\eta}_{\ell}, \qquad \ell \in R^{\eta}, \tag{4.4}$$

$$f_{n+1} \leq \delta_{n+1} \tag{4.5}$$

$$m_i \in M_i, \qquad \forall i \in V \tag{4.6}$$

$$f_i \in int^{+}, \qquad \forall i \in V \tag{4.7}$$

In (3), $\mathcal{A}(F, M, t)$, also called the active set, is a set of real activities which will be sequenced in time t; $M = (m_i)_{i \in V}$ is the mode vector; and $F = (f_i)_{i \in V}$ is a vector of completion times for each activity. The objective function (1) maximizes the project's NPV by discounting the cash inflow and outflow to each activity's finish time. The constraints represented by equation (2) describe the precedence relationships between activities; equation (3) ensures that the availability of renewable resources is not exceeded in each period; and equation (4) ensures that non-renewable resources are available throughout the project. Restrictions in equation (5) ensure that the deadline for project delivery is met, and those in equation (6) ensure that each activity is assigned only one mode. The restriction in (7) states that the decision variables should be integers. It is assumed that dummy activities start and end are only executed in a single zero-duration mode and do not consume resources.

A vector mode μ is a $n + 2-$tuple $\mu = (1, \mu_1, \ldots, \mu_n, 1)$, which assigns to the j-th activity, $1 \leq j \leq n$, a unique mode μ_j, $1 \leq \mu_j \leq |M_j|$. Any vector μ satisfying the constraints in (4) is called resource feasible. Otherwise, it will be said resource non-feasible. We define the excess of non-renewable resource for a vector mode μ by

$$L^{\eta}(\mu) = \sum_{k \in \, R^{\eta}} |\min\{0, R^{\eta}_{k} - \sum_{j=1}^{n} r^{\eta}_{j,\mu(j),k}\}| \tag{4.8}$$

Thus, μ is a feasible resource vector mode if and only if $L^{\eta}(\mu) = 0$.

4.3 The ABFO algorithm

The ABFO algorithm is based on the chemotactic behavior of bacteria that will perceive chemical gradients in the environment (such as nutrients) and move towards or away from specific signals. According to Passino, 2002, this search is saltatory. The ABFO is a metaheuristic strategy that combines concepts from BFO and several metaheuristics such as GA, particle swarm optimization (PSO) and ant colony optimization (ACO). The different steps followed by the ABFO algorithm can be initially subdivided into four mobile behaviors: chemotaxis, swarm, reproduction, and elimination-dispersion. However, we have added two new procedures based on the mutation and a change in the way bacteria reproduce. Parameters used in our implementation of the ABFO algorithm are described in Table 4.1. Algorithm 10 provides an overview of the ABFO algorithm for maximizing the NPV of the project using the PAC model.

Table 4.1: Nomenclature for Algorithm 10.

Parameter	Description
n	Number of actual project activities
$nBact$	Initial number of bacteria in the colony
N_c	The number of chemotactic steps
N_s	The swimming length
N_{re}	The number of reproduction steps
N_{ed}	The number of elimination-dispersal steps
P_{ed}	Elimination-dispersal probability
P_{swarm}	Swarming probability
P_{cross}	Crossover probability
P_{mutE}	Spontaneous mutation probability
P_{mutR}	Reverse mutation probability
$Bact_{best}$	Best bacteria in the colony

4.3.1 Representation

Like GA, ACO and PSO, the ABFO algorithm operates on the same coded representation of solutions. Bacterium are presented as $X = (\lambda, \mu)$, where λ is a $(n+2)-$dimensional vector called activity list, and μ is a mode vector. A list of activities λ is of feasible precedence if all the predecessors of an activity are located in the list before that activity (Bouleimen and Lecocq, 2003; Hartmann, 2001; Jozefowska et al., 2001; Leyman and Vanhoucke, 2016; Van Peteghem and Vanhoucke, 2010). The μ vector is called the mode vector for the corresponding activities. For each activity $j \in V$, a mode $\mu_j \in \mathcal{M}_j$ is selected. Then

$$X = \begin{pmatrix} 0 & j_1 & \cdots & j_k & \cdots & j_n & n+1 \\ 1 & \mu_1 & \cdots & \mu_k & \cdots & \mu_n & 1 \end{pmatrix} \tag{4.9}$$

where each bacteria X is related to a given S_X schedule, which is obtained from the setting of the μ modes. Then, the single-mode resource-constrained project scheduling problem (RCPSP) associated with the mode vector is searched using a serial generation scheme is applied. The advantage of the list of activities is that it is always feasible with respect to the precedence relationships. In addition, it eliminates schedules with incompatibility problems that can never be terminated due to technological limitations. However, it will produce redundancy in the search space and will make the size of the search space at most $n!$ for the list of activities. The search space for the mode vector

Algorithm 10: Pseudocode for our ABFO implementation.

```
Input  : $n$, $nBact$, $N_c$, $N_s$, $N_{re}$, $N_{ed}$, $P_{ed}$, $P_{swarm}$ $P_{cross}$, $P_{mutE}$, $P_{mutR}$
Output: $Bact_{best}$
begin
    $Col \leftarrow InitializeColony(nBact, n)$ ;
    for $l = 1$ to $N_{ed}$ do
        for $k = 1$ to $N_{re}$ do
            for $j = 1$ to $N_c$ do
                $Col \leftarrow Chemotaxis(Col, nBact, N_s)$;
                for all $Bact \in Col$ do
                    if $NPV_{Bact} \geq NPV_{Bact_{best}}$ then
                        $Bact_{best} \leftarrow Bact$
                    end
                end
            end
            $Col \leftarrow SortByCellHealth(Col)$;
            $Swarming(Col, P_{swarm})$;
            $Crossover(Col, P_{cross})$;
            $Mutation(Col, P_{mutE}, P_{mutR})$;
            $Col \leftarrow SelectByCellHealth(Col, nBact/2)$
        end
        for all $Bact \in Col$ do
            if $RAND() \leq P_{ed}$ then
                $Bact \leftarrow CreateBactAtRandomLocation()$
            end
        end
    end
end
```

will be $\mathcal{SP} = \{(1, \mu_1, \cdots, \mu_n, 1) : \mu_j \in \mathcal{M}_j \text{ with } j \in V\}$ taking into account feasible and non-feasible modes. Although other types of representation such as random key (Kolisch and Hartmann, 1999) and topological order (Valls et al., 1999) have been used, Hartmann and Kolisch, 2000 concluded that activity list-based procedures outperform these representations. However, Debels et al., 2006 showed that the standardized random key (SRK) also leads to promising results.

4.3.2 Initial colony

The ABFO algorithm starts by building an initial colony conformed by bacteria generated randomly or by priority rules, for the selection of activities and modes. We use Algorithm 11 to generate *nBact* bacteria. Each bacteria has a mode vector which is produced by the random selection of a mode construction rule; this rule selection includes the shortest feasible mode (SFM)(Boctor, 1996), the lower total work content (LTWC)(Van Peteghem and Vanhoucke, 2011), the minimum resource consumption (MRC)(Kolisch, 2013) and random selection (RAND). Moreover, if μ is not a feasible resource, a local search (LS) procedure, previously used by Hartmann, 2001 in its GA, is applied to transform the mode vector into a vector of feasible modes using Algorithm 12.

Algorithm 11: Generate bacteria.

```
Input  : $par$ Parameter , $P$ Problem
Output: $X$ Bacteria
begin
    Randomly select a rule in $\{SFM, LTWC, MRC, RAND\}$;
    $\mu \leftarrow rule(\cdot)$;
    $\mu \leftarrow$ Feasible mode vector assignment;
    The RCPSP is built with the mode vector;
    Feasible precedence list $\lambda$ is generated with one of the priority rules;
    Bacteria $X$ is defined for $X = (\lambda, \mu)$;
end
```

To generate the list of activities of feasible precedence λ, the modes of the activities are fixed with respect to μ. Then, 10% of the *nBact* solutions are calculated with the Latest Start Time (LST), Shortest Processing Time (SPT), Minimum Slack (MSLK), Resource Scheduling Method (RSM), Latest Finish Time (LFT), and Earliest Start Time (EST) priority rules, as well as the Greatest Range Positional Weight (GRPW), Longest Processing Time (LPT), RAND (Random Priority Rules) and Modified-Cumulative Cash Flow (mCF) rules proposed by Tantisuvanichkul, 2014, to dynamically select activities with the highest value of c_i ($c_{i,in}$ or $c_{i,out}$) from a list of available activities without violating the precedence rules, critical path and other restrictions. Formally, this is equivalent to calculate

Algorithm 12: Feasible mode vector assignment.

```
Input  : Mode vector $\mu$
Output: Feasible mode vector $\mu$
begin
    if $L^{\eta}(\mu) = 0$ then
        the vector of modes is feasible;
    end
    repeat
        Select randomly $j \in V$ with $|M_j| > 1$;
        select randomly $m_j \in M_j \setminus \{\mu_j\}$;
        define $\mu' = (\mu_1, \dots, m_j, \dots, \mu_n)$ calculate $L^{\eta}(\mu)$ y $L^{\eta}(\mu')$;
        if $L^{\eta}(\mu') \leq L^{\eta}(\mu)$ then
            $\mu = \mu'$;
        else
            go to the random selection of the activity;
        end
    until $L^{\eta}(\mu) = 0$ or some stop criterium;
end
```

$$\max_{j \in D_g} c_j \cdot e^{-\alpha \cdot t_j} + \sum_{k \in Suc_j} c_k \cdot e^{-\alpha \cdot (t_k - t_j)} \quad \text{with } t_j = \begin{cases} EST_j, & \text{if } c_j \geq 0, \\ LST_j, & \text{otherwise,} \end{cases} \tag{4.10}$$

where D_g is the set of eligible activities in step g and Suc_j is set successors of $j \in V$. The initial colony is formed with the best $nCol$ obtained, thus combining quality, feasibility and diversity.

4.3.3 Schedule generator scheme

The schedule generator scheme (SGS) is a crucial part of many procedures for the standard MRCPSP. Each bacteria X is related to a certain sequence S, which is obtained from the adjustment of the mode vector μ. The RCPSP associated with said mode vector is searched and the generation scheme is applied. There are two schedule generator schemes, namely the serial (`SGS-serial`) (Kelly, 1963) and parallel (`SGS-parallel`) (Bedworth and Bailey, 1999) schemes. These schemes generate a feasible schedule by extending, in stages, a partial schedule. That is, the finish times are assigned to only a (not necessarily proper) subset of V. At each stage, the SGS builds the set of all eligible activities, also called the decision set. Only the series scheme is detailed, since it works on the temporary increase, significantly increasing the value of the NPV. In addition, the `SGS-serial` starts without sequencing any activities with renewable and non-renewable resources at their initial availability. Then, at each iteration g, $1 \leq g \leq n$, we select the activity at position g based on a priority rule; the start time is calculated such that it complies with the constraints of renewable resources. This activity defines a partial sequence $S_g = (s_{\lambda_1}, \dots, s_{\lambda_g})$. When all the activities have been sequenced, a complete sequence with all the start times of the activities will be available. The completion time of the sequence will be determined by the final dummy activity.

4.3.4 Evaluation of bacterium

An important aspect of the ABFO algorithm is the determination of the evaluation or strength of the bacteria in the colony. This is accomplished by a fitness function, which evaluates the quality of the solutions. To obtain the evaluation of a bacteria, the `SGS-serial` is applied and a feasible sequence F is obtained. If for a given bacteria the applied schedule results in the violation of the deadline or nonrenewable resource constraints, the penalty

$$NPV(X) = \begin{cases} \sum\limits_{i \in V} (c_{i,in} + c_{i,out}) \cdot e^{-\alpha(f_i + |K|)}) & \text{if } L^{\eta}(\mu) = 0 \\ \sum\limits_{i \in V} (c_{i,in} + c_{i,out}) \cdot e^{-\alpha(f_i + C_i + |K|)}) & \text{if } L^{\eta}(\mu) > 0 \end{cases} \tag{4.11}$$

is included in the objective function. Here, $K = \frac{1}{2}\max\{0, \delta_n - f_{n+1}\}$ and

$$C_i = \begin{cases} f_{n+1} + L^{\eta}(\mu) & \text{if } c_{i,in} + c_{i,out} \geq 0 \\ -f_{n+1} - L^{\eta}(\mu) & \text{if } c_{i,in} + c_{i,out} < 0 \end{cases} \tag{4.12}$$

In general, the solution strategies for the MRCPSPDC use penalties when any restriction is violated for the single-mode and multi-mode case. Mika, Waligóra, and Weglarz, 2005 shows a function to evaluate the NPV using a penalty for deadline restriction. However, it only works with positive cash flow. Leyman and Vanhoucke, 2016 use two penalty functions and apply this approach to improve the viability of the project with respect to the project deadline and the feasibility of mode selection, and generalizes the objective function presented by Leyman and Vanhoucke, 2015 for the single-mode case.

4.3.5 Chemotaxis

The chemotaxis process can be described in two operations, namely, swim and tumble. The tumble operation is based on conducting a search in different directions in which the bacteria can find environments with a greater amount of nutrients. This behavior is also called a random direction walk. Instead, the swim operation is considered a walk in the same direction to areas with better conditions. We describe the whole process of swimming and tumbling in a single procedure.

At each step of a chemotaxis process, a vector v of n components is generated; each component belongs to the set $\{0, 1\}$. The total number of "1"s are decided by nonnegative integer called the specific step size for the tumble, which is determined by the $Tum(i)$ as follows (Ge and Tan, 2012):

$$Tum(i) = \left[\!\!\left[\left(\frac{NPV_{\max} - NPV_i}{NPV_{\max} - NPV_{\min}} \right) n \right]\!\!\right] \tag{4.13}$$

In this expression, $[\![\cdot]\!]$ is the integer part function, NPV_i is the net present value of the $i-$th bacteria, and $NPV_{\max}$ and $NPV_{\min}$ correspond, respectively, to the maximum and minimum value of the NPV in the colony. As $Tum(i)$ is inversely proportional to the fitness of the $i-$th bacteria, a high value in the evaluation of the $i-$th bacteria produces a low $Tum(i)$ (Ge and Tan, 2012).

To perform the tumble on the $i-$th bacteria, the number of $Tum(i)$ positions to be selected from the position vector is randomly calculated. Further, values of the elements of vector v at these positions are set to "1". Then, for these positions, a priority value, which indicates the order to make the change in the position vector, is calculated as:

$$priori(j) = \frac{c_j \cdot e^{-\alpha \cdot t} + \sum_{k \in S_j} c_k \cdot e^{-\alpha \cdot (t_k - t_j)}}{\max_{j \in \mathcal{H}} \{ c_j \cdot e^{-\alpha \cdot t} + \sum_{k \in S_j} c_k \cdot e^{-\alpha \cdot (t_k - t_j)} \}} \tag{4.14}$$

In the expression above, the set $\mathcal{H}$ corresponds to those activities for which the vector v has a 1 in that position. We select the highest NPV given an activity and its successors. Then, for each of them, we calculate the quotient in equation (14) and determine the direction according to the highest value among the priorities. After finding the directions to make the bacteria turn, the swimming mechanism is applied. For each direction, we will make a movement regarding the duration of the activities by changing the mode as long as the new mode complies with the non-renewable resource restrictions. Algorithm 13 shows the chemotaxis procedure. The $TumbleSwim(\cdot)$ operator, which is a fundamental part of Algorithm 13, is described in Algorithm 14. This implementation is based on the double multiple mode justification by Barrios, Ballestin, and Valls, 2011. We adapted these operators to increase the value of the objective function. Hence, given the vector v, we make a movement with respect to the duration of the activities by changing the mode as long as the new mode meets the non-renewable resource restrictions. After making the movements with respect to the modes, we make a change in the start and end times of those activities for which it optimizes the objective function. Thus, activities $j \in V$ with positive cash flow will be justified on the left, or on the right otherwise.

4.3.6 Swarming

We adapt the Swarming operator by Ge and Tan, 2012 to the MRCPSPDC. Initially, an $n-$dimensional vector v with entries in the set $\{0, 1\}$ is generated. This vector defines the order in which the elements of the newly produced bacteria vector are extracted

Algorithm 13: Chemotaxis procedure.

```
Input  : $Col, n, S, N_s$
Output: $Col$
begin
    for all $Bact \in Col$ do
        for $i = 1$ to $N_s$ do
            Define $Tum(Bact)$ and $v$;
            $Bact' \leftarrow TumbleSwim(Bact, Tum(Bact), v)$;
            if $Npv(Bact') < Npv(Bact)$ then
                $i \leftarrow N_s$;
            else
                $Bact \leftarrow Bact'$;
            end
        end
    end
end
```

Algorithm 14: TumbleSwim Operator.

```
Input  : $Bact, Tum(Bact), v$
Output: $Bact$
begin
    Define $\mathcal{H} = \{i \in V : v(i) = 1\}$ with $|\mathcal{H}| = Tum(Bact)$;
    Calculate $priori(j)$ for each $j \in \mathcal{H}$.;
    for $k = 1$ to $Tum(Bact)$ do
        Select $i = \min\{j \in \mathcal{H} : priori(j) = \max_{s \in \mathcal{H}}\{priori(s)\}\}$;
        for $h = 1$ to $|M_i|$ do
            Build $M'$: $M'(j) = M(j)\ \forall j \neq i$, $M'(i) = h$;
            if $M'$ is feasible resource then
                Calculate $ES_{i_h}$ and $LS_{i_h}$;
                if $ES_{i_h} < LS_{i_h}$ then
                    Calculate the $t \in [ES_{i_h}, LS_{i_h}]$ where $i$ can be sequenced in $[t, t + d_{i,h}]$
                     and $c_i \cdot e^{-\alpha t}$ is greatest.
                end
            end
        end
        Update activity list and mode vector $M$ of $Bact$;
        Calculate $\mathcal{H} = \mathcal{H} \setminus \{i\}$ ;
    end
end
```

from the best bacteria and the selected bacteria, respectively. After one element is extracted from one bacteria and removed from the other, it is added to the newly produced bacterial vector. This step is repeated until both the best and the best vector of bacteria are empty and the resulting bacteria contains all the elements involved. A new bacteria is created based on the detection and communication of the quorum and is accepted in the population if its aptitude is higher than the original. Figure 4.1 gives an example of the Swarming operator. It should be noted that, when the Swarming operator is used, it is possible to produce premature convergence to a local optimum. To avoid this, a probability P_{swarm} must be considered.

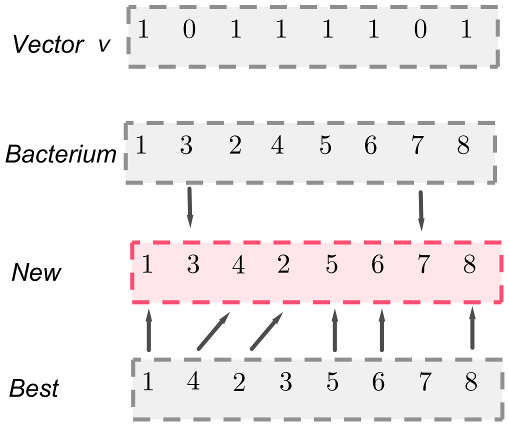


Figure 4.1: Example of the Swarming operator.

4.3.7 Crossover

After performing the Tumbling operator and swimming in the chemotactic step, it is necessary to order the bacteria according to their health in descending form. To simulate the behavior of some bacteria inheriting information to their descendants, we use a uniform crossing operator that generalizes the crossing operators of one and two points. Algorithm 15 shows the pseudocode for the uniform crossing operator used in our approach. When using this crossing operator, the colony is divided into two groups to select the parental bacteria; these groups are subsequently crossed to produce the descendant bacteria.

4.3.8 Mutation

The mutation operator has a spontaneous mutation and a reverse mutation phase (Amador-Fontalvo, Paternina-Arboleda, and Montoya-Torres, 2014), and runs as long as the bacteria need it, depending on the colony. A bacterium that has no affinity with colony members will be more likely to mutate than one that has good affinity.

In the spontaneous phase, the mutation operator alters one or more positions of a selected bacteria, that is, reintroduces lost material and thus have some extra variability in the colony. In fact, spontaneous mutation can lead to totally new bacteria,

Algorithm 15: Uniform crossover operator.

```
Input  : $X^M$, $X^F$ and $v$.
Output: $X^D$
begin
    if $v_{n+1} = 0$ then
        for $i = 1$ to $n$ do
            if $v_i = 0$ then
                Calculate $k = \min\{\ell \in N : j_\ell^M \notin \{j_1^D, \ldots, j_{i-1}^D\}\}$;
                Define $j_i^D = j_k^M$;
            else
                Calculate $k = \min\{\ell \in N : j_\ell^F \notin \{j_1^D, \ldots, j_{i-1}^D\}\}$;
                Define $j_i^D = j_k^F$;
            end
        end
    else
        for $i = n$ to $1$ do
            if $v_i = 0$ then
                Calculate $k = \max\{\ell \in N : j_\ell^M \notin \{j_1^D, \ldots, j_{i-1}^D\}\}$;
                Define $j_i^D = j_k^M$;
            else
                Calculate $k = \max\{\ell \in N : j_\ell^F \notin \{j_1^D, \ldots, j_{i-1}^D\}\}$;
                Define $j_i^D = j_k^F$;
            end
        end
    end
    for $i = 1$ to $n$ do
        if $v_i = 0$ then
            $\mu(j_i^D) = \mu^M(j_i^D)$;
        else
            $\mu(j_i^D) = \mu^F(j_i^D)$;
        end
    end
end
```

which sometimes allows ABFO to reach better solutions. In addition, the mutation helps preventing population stagnation at any local optimum. The mutation applies to both the λ activity list and the μ mode vector. In the first case, the spontaneous mutation procedure used is that of insertion which works as follows: for each activity in the list of activities a new position is chosen at random between the highest positions of its predecessors and the lowest position of its successors. The activity is then inserted in the new position with a probability P_{mutE}. As for the mode assignment vector, the application of mutation operator changes depending on whether or not the bacterium has a feasible resource mode vector. If the bacterium has a feasible mode vector, that mode vector is left. Otherwise, it seeks to improve the consumption of non-renewable resources.

The second phase is the reverse mutation. This process is used only for a part of the bacteria that performed spontaneous mutation, which are selected with probability P_{mutR} and still have little affinity with the colony. If these bacteria have moved to unfeasible regions, they should return to the position they had before the spontaneous mutation took place. We consider $P_{mutE} = P_{mutR} = P_{mut}$.

4.3.9 Selection

At the end of the mutation of the bacteria, the number of bacteria has doubled because the uniform crossing operator produces two daughter bacteria whenever the parents cross. It is necessary to select $nBact/2$ bacteria with the highest level of health determined by $NPV(\cdot)$, while the other $nBact/2$ bacteria are eliminated from the colony.

4.3.10 Elimination and Dispersal

The process of elimination-dispersion is responsible for updating the colony, eliminating some bacteria with probability P_{ed} and then introducing bacteria in different positions. In reality, the elimination-dispersion operator helps preventing the premature convergence of the colony, thus introducing new information to it.

4.4 Computational Experiments

In this section, we present the results of computational experiments. In section 4.4.1 we present the set of instances resolved using the ABFO algorithm, the algorithm parameters and the stop criteria. In section 4.4.2, we perform a Taguchi experimental design to quantify the influence of each parameter of our ABFO algorithm on its performance, while in section 4.4.3 we present the results of the algorithm evaluated in the sets of instances. In section 4.4.4 we compare the results of our ABFO with the genetic algorithm (GA) presented in Leyman and Vanhoucke, 2016. Finally, we compare the contribution of ABFO operators applied to the set of MMLIB50 instances in section 4.4.5.

4.4.1 Set of instances and parameter settings

The ABFO algorithm has been coded and compiled in C++. In order to test our ABFO algorithm, computational tests were performed on a Toshiba laptop computer with an Intel Core 2.3 GHz processor. Experiments were similar to those presented in Leyman and Vanhoucke, 2016 for the PAC payment function using the sets of instances of the PSPLIB (Kolisch and Sprecher, 1996) and MMLIB (Van Peteghem and Vanhoucke, 2014) libraries. The PSPLIB library contains seven sets of instances (J10, J12, J14, J16, J18, J20 and J30); each instance is comprised of 640 schedules with two renewable resources, two non-renewable resources, and 10, 12, 14, 16, 18, 20 and 30 activities, respectively. However, there are instances that are not feasible, i.e., not all of these instances can be resolved. Thus, all non-feasible instances were excluded from our research. Although the MMLIB library is composed by three instances (i.e., MMLIB50, MMLIB100 and MMLIB+), the sets MMLIB50 and MMLIB100 have 540 instances each with 50 and 100 activities, respectively. Furthermore, each activity has two renewable resources, two non-renewable resources and three modes. The MMLIB+ set, on the other hand, has a total of 3240 instances, between 2-4 renewable resources, 2-4 non-renewable resources with 3, 6 and 9 modes per activity. For experimental purposes, we assumed project deadline values $D_{Incr} \in \{makespan \times (1+K) \backslash K = 5\%, 10\%, 15\%, 20\%\}$, where $makespan$ is the minimum duration of the project found to date. For the cash flow, we used the data in Leyman and Vanhoucke, 2016 such that for each project there is a percentage of negative cash flow $(\%Neg)$, which takes values in $\{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$. The ABFO algorithm was executed using the optimal values established by our Taguchi design, and 5000 schedules as the stopping criterion following Lova et al., 2009.

4.4.2 Taguchi experimental design

The Taguchi experimental design is a standardized form of design of experiments (DOE) developed by Dr. Genichi Taguchi in the 1940s to improve the quality of manufactured goods in an effort to simplifying and standardizing the application of the DOE technique. Two of the major tools used in the Taguchi's method include the orthogonal array (OA) and the signal-to-noise ratio (SNR). This experimental design applies fractional factorial test designs, called OAs, that serve to reduce the number of experiments to be performed. The selection of a suitable OA depends on the number of control factors and their levels. Using an OA design can estimate the effect of multiple process variables which are simultaneously affecting on the performance characteristic, while minimizing the number of test runs. Whether a particular process variable has an effect on the response of interest is determined using an Analysis of Variance (ANOVA). The OA is denoted as $L_N(S^k)$, where N is the number of factor combinations in the experiment, S in the number of levels for each factors, and k represents the number of factors in the experiment. Following Taguchi's method, the minimum number of runs required to conduct the experiment can be calculated as $N_{\text{Taguchi}} = 1 + \sum_{i=1}^{k}(S^k - 1) = 15$.

	Levels (coded)			Levels (uncoded)		
Factor	Level 1	Level 2	Level 3	Level 1	Level 2	Level 3
$nBact$	1	2	3	10	30	50
N_c	1	2	3	2	4	6
N_{re}	1	2	3	5	10	15
N_{ed}	1	2	3	5	10	15
P_{swarm}	1	2	3	0.1	0.5	0.9
P_{cross}	1	2	3	0.1	0.5	0.9
P_{mut}	1	2	3	0.1	0.5	0.9

Table 4.2: Parameters (factors) and their levels.

In this study, the identified factors were colony size ($nBact$), the elimination-dispersion step (N_{ed}), reproduction step (N_{re}), chemotactic step (N_c), swarming probability (P_{swarm}), mutation probability (P_{mut}) and crossover probability (P_{cross}), each of which has three possible levels (Table 4.2). The suitable OA should be greater than the number of experiments N_{Taguchi}. Thus, the $L_{27}(3^7)$ OA was selected. Table 4.3 shows the combinations of ABFO algorithm parameters for the Taguchi experimental design.

Taguchi's method was used to determine the optimal parameters for our ABFO algorithm. Experiments were performed in the MMLIB50 set of instances of the MMLIB library. In instances where feasible solutions ($Feas$) were determined (i.e., solutions that do not violate resource and deadline restrictions), the response variable was the average NPV $(AvgNPV)$ calculated as

$$AvgNPV = \frac{1}{|Feas|} \sum_{x \in Feas} NPV(x) \tag{4.15}$$

For each instance of the MMLIB50 set, the experiment was conducted by randomly selecting a negative percentage cash flow in $\{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$ and a deadline project increase in $\{5\%, 10\%, 15\%, 20\%\}$. Table 4.4 shows the ANOVA results; column C(%) is the percentage contribution of each factor to the total variation. We found that factors $nBact$ and N_{ed} have the higher significant effects on $AvgNPV$; the percentage contribution to the mean squared is 37% and $\approx$ 25%, respectively. Although the other factors are not statistically significant, they were kept as they may introduce new information or variations to the colony. Now, in order to avoid incorrectly selecting the model parameters and minimize Type I or Type II errors, the resulting P-values were corrected for multiple testing using the False Discovery Rate (Benjamini and Hochberg, 1995); corrected P-values are reported in Table 4.4 the P-Adjust column.

Table 4.5 shows the $AvgNPV$ for each level of factors in the Taguchi experimental design (Table 4.3). The Δ value represents the maximum difference between the level responses. These differences are then ranked from 1 to 7, being 7 the rank for the maximum value of $AvgNPV$. Thus, the optimum set up parameters that maximize

	Factors							
Experiment	$nBact$	N_c	N_{re}	N_{ed}	P_{swarm}	P_{cross}	P_{mut}	$AvgNPV$
1	10	6	10	10	0.1	0.9	0.9	-82192.94
2	10	6	10	15	0.5	0.1	0.5	-92610.33
3	10	4	15	5	0.5	0.9	0.1	-97486.04
4	10	2	5	10	0.5	0.5	0.9	-89516.52
5	10	2	5	5	0.1	0.1	0.1	-123536.66
6	10	2	5	15	0.9	0.9	0.5	-88533.90
7	10	4	15	15	0.1	0.5	0.5	-88789.31
8	10	6	10	5	0.9	0.5	0.1	-85040.58
9	10	4	15	10	0.9	0.1	0.9	-94020.51
10	30	2	15	10	0.1	0.9	0.5	-49722.65
11	30	4	10	10	0.5	0.5	0.5	-61358.33
12	30	2	15	5	0.9	0.5	0.9	-76807.57
13	30	2	15	15	0.5	0.1	0.1	-65278.19
14	30	4	10	5	0.1	0.1	0.9	-104694.10
15	30	6	5	15	0.1	0.5	0.1	-78679.51
16	30	6	5	10	0.9	0.1	0.5	-84569.60
17	30	4	10	15	0.9	0.9	0.1	-80373.05
18	30	6	5	5	0.5	0.9	0.9	-116035.01
19	50	2	10	15	0.1	0.5	0.9	-36236.18
20	50	6	15	10	0.5	0.5	0.1	-73876.07
21	50	2	10	10	0.9	0.1	0.1	-69788.72
22	50	6	15	5	0.1	0.1	0.5	-87026.80
23	50	4	5	10	0.1	0.9	0.1	-68129.99
24	50	4	5	5	0.9	0.5	0.5	-79919.31
25	50	6	15	15	0.9	0.9	0.9	-37661.26
26	50	2	10	5	0.5	0.9	0.5	-64536.93
27	50	4	5	15	0.5	0.1	0.9	-48600.91

Table 4.3: Orthogonal matrix design L_{27} for the Taguchi experiment.

Factor	**Df.**	**Mean Square**	**F-value**	**C(%)**	**P-Value**	**P-Adjust**
$nBact$	2	2122261837	10.802	37.007	**0.002**	**0.014**
N_c	2	169850867	0.864	2.962	0.446	0.624
N_{re}	2	399890192	2.035	6.973	0.173	0.404
N_{ed}	2	1427041817	7.263	24.884	**0.009**	**0.029**
P_{swarm}	2	13882248	0.071	0.242	0.932	0.932
P_{cross}	2	323919444	1.649	5.648	0.233	0.407
P_{mut}	2	99024863	0.504	1.727	0.616	0.719
Residuals	12	196478188		20.557		

Residual standard error: 14020 on 12 degrees of freedom.
Multiple R-squared: 0.7944, Adjusted R-squared: 0.5546.
F-statistic: 3.313 on 14 and 12 DF, p-value: 0.02212

Table 4.4: Results of the ANOVA.

	Factors						
	$nBact$	N_c	N_{re}	N_{ed}	P_{swarm}	P_{cross}	P_{mut}
Level 1	-93525	-73773	-86391	-92787	-79889	-85569	-82465
Level 2	-79724	-80374	-75203	-74797	-78810	-74469	-77451
Level 3	-62864	-81965	-74518	-68529	-77412	-76074	-76196
Δ	30661.18	8192	11872	24257	2477	11100	6269
Rank	7	3	5	6	1	4	2

Table 4.5: Ranking of factors by level effects.

the NPV when using our ABFO algorithm are $nBact = 50$, $N_c = 2$, $N_{re} = 15$, $N_{ed} = 15$, $P_{swarm} = 0.9$, $P_{cross} = 0.5$ and $P_{mut} = 0.9$.

4.4.3 Results of the ABFO

In this stage we evaluate the performance of our ABFO algorithm under the set instances and parameter settings for the MRCPSPDC. Table 4.6 shows the results of the MRCPSPDC under the PAC model for the set of instances in the MMLIB library. The performance comparison criteria are based on the $AvgNPV$ and the percentage of feasible solutions $(\%FEAS)$.

First, we selected the MMLIB50, MMLIB100 and MMLIB+ libraries. Secondly, we progressively increased the levels of $\%Deadline$ (columns of the array), which resulted in a significant increase of $AvgNPV$ and the percentage of feasible solutions $(\%FEAS)$. Finally, we progressively increased the $\%Neg$ (rows of the array) decreasing the value of $AvgNPV$ as there are more activities with negative cash flow. However, it should be noted that the number of feasible solutions increases. For instance, in the MMLIB100 library with a $\%Deadline$ of 10% and a $\%Neg$ of 60%, the AvgNPV is -671.68 with a $\%FEAS$ of 71.3%.

Figure 4.3 depicts the computation time of our ABFO to find feasible solutions for each set of instances. For the instances of the PSPLIB library, the computational time increases as the number of activities increases. However, the average computation time time is considerably low. For the MMLIB50, MMLIB100 and MMLIB+ instances, increasing the number of modes produces a significant increase in computation time.

4.4.4 Comparison with GA

Here we compare the results of our proposed ABFO algorithm with those of the GA by Leyman and Vanhoucke, 2016. Data for the GA were obtained from `http://www.projectmanagement.ugent.be`. The performance of the algorithms is compared on the basis of 5000 schedules generated as the stopping criteria. Tables 4.7 and 4.8 summarize the $AvgNPV$ when $\%Deadline$ and $\%Neg$ vary. The Mean row corresponds to the $AvgNPV$ fixing $\%Deadline$ and varying Neg. Although ABFO does not dominate

Table 4.6: Average NPV obtained by the ABFO algorithm for the MMLIB library and percentage of feasible solutions found.

		%Deadline			
	%Neg.	5%	10%	15%	20%
MMLIB50	0	11063.18	10981.72	11008.29	11030.17
		(60.37)	(90.56)	(96.85)	(99.81)
	20	5811.49	5787.93	5810.47	5826.71
		(61.30)	(90.93)	(97.78)	(99.44)
	40	1029.72	1026.02	1048.68	1068.70
		(60.56)	(91.11)	(98.15)	(100.00)
	60	-1135.39	-1095.17	-1057.08	-1021.80
		(62.96)	(91.67)	(97.78)	(99.63)
	80	-4059.89	-3974.51	-3904.71	-3835.91
		(68.52)	(91.85)	(97.41)	(99.63)
	100	-10501.82	-10362.69	-10221.25	-10064.90
		(75.93)	(96.48)	(98.70)	(100.00)
MMLIB100	0	20884.26	20388.61	20386.17	20387.45
		(46.85)	(71.48)	(88.33)	(95.19)
	20	12760.47	12406.55	12344.44	12367.75
		(45.19)	(70.00)	(89.07)	(96.30)
	40	4074.07	3941.03	3937.45	3965.40
		(47.78)	(71.67)	(87.96)	(96.30)
	60	-698.56	-671.69	-641.67	-600.40
		(49.07)	(71.30)	(87.96)	(96.30)
	80	-7911.51	-7635.61	-7501.70	-7377.78
		(51.85)	(76.30)	(90.56)	(97.22)
	100	-19529.06	-19047.71	-18759.77	-18456.16
		(58.52)	(78.52)	(91.67)	(97.78)
MMLIB+	0	12960.73	10981.65	10796.50	10802.35
		(19.73)	(58.08)	(80.48)	(91.72)
	20	7377.31	6214.98	6062.11	6069.55)
		(18.19)	(57.91)	(81.99)	(92.81)
	40	1922.54	1446.85	1429.45	1450.54
		(20.67)	(59.76)	(81.36)	(92.96)
	60	-735.30	-693.72	-626.05	-560.89
		(21.83)	(61.38)	(82.94)	(92.71)
	80	-4504.76	-3752.80	-3522.94	-3408.19
		(25.37)	(64.97)	(85.11)	(93.75)
	100	-11485.75	-9954.19	-9558.94	-9313.23
		(34.57)	(70.65)	(86.76)	(94.74)

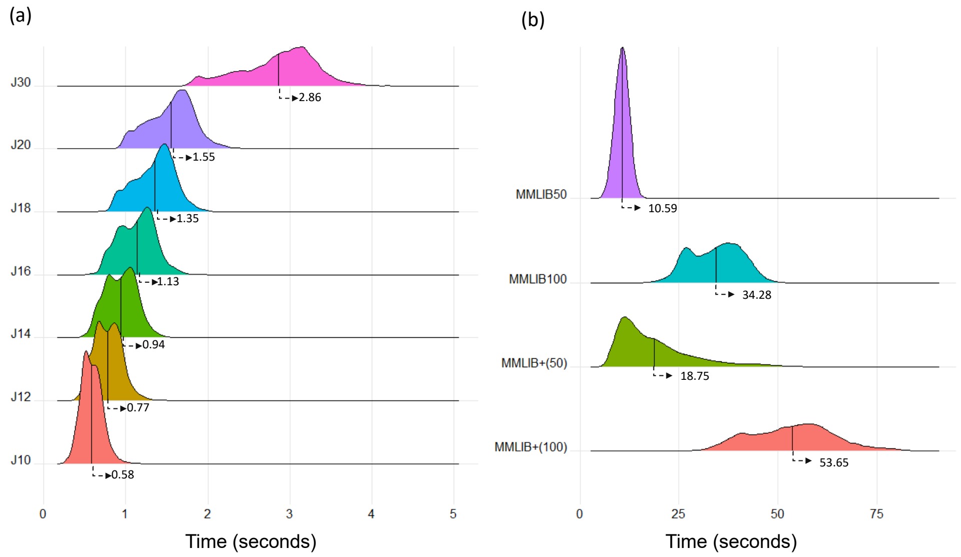


Figure 4.2: Comparison of the computation times for the **(a)** PSPLIB and **(b)** MMLIB libraries with 5000 generated schedules as the stopping condition. Arrows show average computation time.

GA in all sets of instances, the improvement in the percentage of feasible solutions is remarkable.

4.4.5 Comparison of ABFO operators.

To illustrate the technique presented in this work, we applied the ABFO algorithm to the set of instances of the MMLIB50 with 50 nonfiction activities that require two renewable and two nonrenewable resources for their execution. Each activity has three execution modes. Specifically, we calculated the $AvgNPV$ for each subset of instances varying $\%Deadline$.

Figure 4.3 shows the $AvgNPV$ as a function of the number of sequences when different ABFO operators are used. It is clear that separate operators do not provide remarkable results. The crossover operator, at the beginning of the ABFO, generates diversity and improves the NPV, but solutions are also suitable for a local optimal. Interestingly, the mutation operator improves the NPV significantly. On the other hand, the chemotactic step, unlike the other operators, produces seasonality on the NPV as the number of schedules increases. It should be noted that the Swarming operator converges fast, which is a situation that can be controlled with the elimination-dispersion step.

Table 4.7: Average NPV (top) between ABFO and GA for instances in the PSPLIB library when 5000 schedules are used as the stopping criterion . %*FEAS* is shown in ().

		%Deadline							
	%Neg.	5%		10%		15%		20%	
		ABFO	GA	ABFO	GA	ABFO	GA	ABFO	GA
J10	0	2435.84	2430.42	2442.06	2434.05	2443.90	2435.99	2444.38	2437.04
		(94.95)	(96.93)	(99.46)	(99.82)	(100.00)	(100.00)	(100.00)	(100.00)
	20	1259.45	1238.12	1274.10	1267.60	1284.37	1274.89	1294.39	1281.61
		(91.70)	(96.93)	(99.64)	(99.64)	(100.00)	(100.00)	(100.00)	(100.00)
	40	504.66	503.94	521.94	515.46	535.53	524.37	548.31	532.99
		(91.52)	(96.57)	(99.64)	(99.82)	(100.00)	(100.00)	(100.00)	(100.00)
	60	505.08	502.52	521.63	513.46	535.56	522.29	548.26	530.76
		(93.14)	(87.36)	(99.82)	(96.39)	(100.00)	(99.46)	(100.00)	(99.10)
	80	75.76	35.65	94.50	84.84	109.32	94.14	124.61	103.23
		(94.40)	(89.89)	(99.10)	(98.38)	(100.00)	(100.00)	(100.00)	(99.82)
	100	-2365.01	-2360.16	-2316.92	-2334.18	-2272.05	-2311.34	-2223.52	-2286.78
		(95.49)	(90.43)	(99.64)	(98.92)	(100.00)	(99.82)	(100.00)	(100.00)
	Mean	389.39	399.13	423.12	415.02	439.43	423.39	456.07	433.14
		(95.76)	(98.41)	(99.78)	(99.81)	(100.00)	(100.00)	(100.00)	(100.00)
J12	0	2943.07	2937.30	2949.00	2940.83	2950.90	2942.31	2951.11	2943.48
		(95.80)	(98.90)	(99.82)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)
	20	1733.65	1731.44	1741.91	1737.94	1747.31	1742.68	1751.84	1746.78
		(96.16)	(98.54)	(99.27)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)
	40	22.00	19.21	46.59	35.24	71.23	49.89	97.51	64.05
		(96.34)	(98.17)	(99.09)	(100.00)	(99.82)	(100.00)	(100.00)	(100.00)
	60	21.82	17.75	46.70	33.30	71.55	47.16	98.00	61.17
		(96.16)	(94.52)	(99.09)	(98.35)	(100.00)	(100.00)	(100.00)	(99.82)
	80	-410.28	-415.52	-383.51	-398.86	-357.17	-384.04	-330.18	-369.73
		(95.43)	(98.35)	(98.90)	(100.00)	(99.82)	(100.00)	(100.00)	(100.00)
	100	-2848.26	-2851.10	-2792.22	-2815.68	-2730.69	-2784.52	-2665.07	-2753.01
		(97.81)	(98.17)	(99.82)	(99.82)	(100.00)	(100.00)	(100.00)	(100.00)
	Mean	230.69	245.32	267.38	257.01	292.19	268.91	317.2	282.19
		(96.28)	(97.78)	(99.32)	(99.70)	(99.94)	(100.00)	(100.00)	(100.00)
J14	0	3091.35	3089.76	3101.04	3095.11	3104.35	3097.26	3105.00	3098.71
		(94.37)	(98.37)	(99.82)	(100.00)	(99.82)	(100.00)	(100.00)	(100.00)
	20	1884.18	1881.59	1893.07	1889.45	1898.42	1894.54	1901.69	1898.08
		(94.37)	(98.19)	(99.64)	(100.00)	(99.82)	(100.00)	(100.00)	(100.00)
	40	170.89	169.58	194.51	185.82	216.69	200.52	238.56	214.63
		(94.56)	(97.64)	(99.27)	(100.00)	(99.64)	(100.00)	(100.00)	(100.00)
	60	171.95	165.67	194.31	182.96	217.36	196.97	238.55	210.91
		(94.19)	(94.74)	(98.91)	(99.82)	(99.64)	(99.82)	(100.00)	(100.00)
	80	-532.39	-539.82	-502.04	-518.71	-471.22	-501.27	-440.04	-484.06
		(96.01)	(94.19)	(99.27)	(99.46)	(99.82)	(100.00)	(100.00)	(100.00)
	100	-2979.71	-2988.29	-2919.18	-2951.06	-2849.76	-2914.89	-2773.49	-2878.88
		(96.73)	(96.19)	(99.64)	(99.64)	(100.00)	(100.00)	(100.00)	(100.00)
	Mean	281.25	315.27	326.16	316.70	352.64	328.90	378.38	343.23
		(95.03)	(96.55)	(99.42)	(99.82)	(99.79)	(99.97)	(100.00)	(100.00)
J16	0	3303.73	3303.45	3311.95	3309.99	3315.24	3312.37	3316.84	3313.55
		(95.82)	(98.18)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)
	20	2090.30	2093.36	2101.13	2100.82	2106.21	2105.70	2109.13	2109.22
		(94.00)	(98.36)	(99.27)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)
	40	-18.36	-18.27	12.08	0.91	40.82	18.96	69.19	36.46
		(95.09)	(94.91)	(99.27)	(99.45)	(99.82)	(100.00)	(100.00)	(100.00)
	60	-17.82	-20.20	12.36	-1.73	40.94	16.10	68.39	32.54
		(93.64)	(83.27)	(99.64)	(93.82)	(100.00)	(98.36)	(100.00)	(99.45)
	80	-727.61	-734.91	-688.78	-710.75	-651.25	-688.96	-612.66	-667.96
		(94.00)	(95.09)	(99.64)	(99.09)	(100.00)	(99.82)	(100.00)	(100.00)
	100	-3171.58	-3183.93	-3102.02	-3142.69	-3020.57	-3100.98	-2934.64	-3060.59
		(96.36)	(95.27)	(100.00)	(99.64)	(100.00)	(100.00)	(100.00)	(100.00)
	Mean	220.7	271.83	273.64	265.97	305.23	278.21	336.04	294.11
		(94.81)	(94.18)	(99.63)	(98.67)	(99.96)	(99.70)	(100.00)	(99.91)
J18	0	3579.79	3580.53	3585.97	3583.61	3589.78	3586.10	3590.69	3587.36
		(93.84)	(97.83)	(98.55)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)
	20	1904.74	1907.41	1918.79	1918.96	1930.07	1926.81	1937.97	1934.02
		(92.75)	(97.28)	(99.09)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)
	40	-209.26	-211.02	-174.00	-185.54	-139.98	-162.90	-105.51	-140.82
		(93.84)	(95.65)	(98.91)	(99.09)	(100.00)	(100.00)	(100.00)	(100.00)
	60	-207.70	-214.79	-174.66	-191.48	-139.89	-168.62	-106.39	-146.47
		(92.93)	(89.13)	(99.28)	(94.75)	(100.00)	(96.92)	(100.00)	(98.55)
	80	-919.74	-925.89	-876.96	-898.30	-833.63	-872.23	-789.77	-846.72
		(95.29)	(91.85)	(99.64)	(99.46)	(100.00)	(100.00)	(100.00)	(100.00)
	100	-3426.82	-3436.51	-3345.05	-3386.52	-3254.38	-3341.03	-3159.03	-3293.33
		(96.20)	(93.48)	(99.82)	(99.64)	(100.00)	(99.82)	(100.00)	(100.00)
	Mean	87.24	159.62	155.66	146.66	192	164.12	227.99	183.14
		(94.14)	(94.2)	(99.21)	(98.82)	((100.00)	(99.46)	(100.00)	(99.76)

		%Deadline							
	%Neg.	5%		10%		15%		20%	
		ABFO	GA	ABFO	GA	ABFO	GA	ABFO	GA
J20	0	4037.93	4040.64	4046.59	4042.21	4049.56	4045.32	4050.85	4046.70
		(94.95)	(96.93)	(99.46)	(99.82)	(100.00)	(100.00)	(100.00)	(100.00)
	20	2363.33	2365.48	2375.30	2374.09	2382.99	2381.38	2388.34	2387.35
		(91.70)	(96.93)	(99.64)	(99.64)	(100.00)	(100.00)	(100.00)	(100.00)
	40	215.33	217.28	245.05	239.23	267.54	257.02	290.62	274.61
		(91.52)	(96.57)	(99.64)	(99.82)	(100.00)	(100.00)	(100.00)	(100.00)
	60	-296.81	-304.94	-255.68	-276.43	-218.09	-250.20	-181.94	-226.95
		(93.14)	(87.36)	(99.82)	(96.39)	(100.00)	(99.46)	(100.00)	(99.10)
	80	-1014.42	-1018.71	-963.50	-987.03	-917.83	-959.24	-871.37	-930.68
		(94.40)	(89.89)	(99.10)	(98.38)	(100.00)	(100.00)	(100.00)	(99.82)
	100	-3866.50	-3882.87	-3769.65	-3819.57	-3671.12	-3766.57	-3556.35	-3709.72
		(95.49)	(90.43)	(99.64)	(98.92)	(100.00)	(99.82)	(100.00)	(100.00)
	Mean	214.8	309.22	279.68	273.81	315.51	286.32	353.36	308.06
		(93.53)	(93.02)	(99.54)	(98.83)	(100.00)	(99.88)	(100.00)	(99.82)
J30	0	6173.75	6125.11	6107.69	6113.25	6100.99	6117.84	6104.77	6119.92
		(84.42)	(90.40)	(99.28)	(99.82)	(100.00)	(100.00)	(99.82)	(100.00)
	20	2790.08	2786.63	2772.47	2797.91	2778.92	2810.95	2788.66	2823.08
		(81.70)	(91.49)	(98.55)	(99.64)	(100.00)	(99.82)	(100.00)	(100.00)
	40	-994.09	-945.12	-946.05	-894.05	-909.70	-850.19	-875.93	-808.17
		(85.87)	(91.12)	(98.91)	(100.00)	(100.00)	(100.00)	(100.00)	(100.00)
	60	-2100.12	-2075.38	-2030.26	-1997.90	-1976.82	-1933.74	-1925.36	-1874.80
		(87.86)	(78.99)	(99.64)	(95.47)	(100.00)	(98.91)	(100.00)	(100.00)
	80	-2820.72	-2791.24	-2746.76	-2711.66	-2691.67	-2647.61	-2635.55	-2582.33
		(89.31)	(80.07)	(98.73)	(95.65)	(100.00)	(99.46)	(100.00)	(100.00)
	100	-5858.71	-5826.44	-5766.29	-5703.67	-5697.38	-5607.49	-5617.21	-5508.83
		(92.21)	(81.52)	(99.46)	(96.74)	(100.00)	(99.28)	(100.00)	(100.00)
	Mean	-744.97	-272.39	-519.17	-344.44	-414.91	-341.33	-359.74	-305.19
		(86.89)	(85.60)	(99.09)	(97.89)	(100.00)	(99.58)	(99.96)	(100.00)

Table 4.8: Average NPV (top) between ABFO and GA for instances in the MMLIB library when 5000 schedules are used as the stopping criterion . %*FEAS* is shown in ().

		%Deadline							
	%Neg.	5%		10%		15%		20%	
		ABFO	GA	ABFO	GA	ABFO	GA	ABFO	GA
MMLIB50	0	11027.15	11008.97	10996.76	11007.88	11014.20	11021.31	11030.17	11022.71
		(71.48)	(84.26)	(95.00)	(98.15)	(98.52)	(99.81)	(99.81)	(100.00)
	20	5810.49	5795.87	5798.72	5798.67	5815.84	5815.39	5828.80	5825.73
		(70.19)	(83.70)	(95.00)	(97.22)	(98.89)	(99.81)	(99.81)	(100.00)
	40	1027.78	1017.02	1030.29	1044.05	1050.07	1071.81	1068.70	1096.19
		(69.63)	(81.30)	(95.19)	(97.41)	(98.89)	(99.81)	(100.00)	(100.00)
	60	-1135.39	-1120.45	-1098.35	-1073.95	-1058.91	-1017.49	-1022.14	-960.79
		(74.63)	(67.22)	(95.74)	(89.26)	(98.89)	(97.22)	(99.81)	(99.63)
	80	-4054.01	-4041.39	-3982.96	-3942.37	-3909.06	-3844.64	-3836.76	-3747.00
		(78.52)	(67.78)	(95.56)	(89.81)	(98.70)	(96.67)	(99.81)	(99.63)
	100	-10515.11	-10507.88	-10376.61	-10304.12	-10224.03	-10131.62	-10064.90	-9944.98
		(84.26)	(72.59)	(98.33)	(90.00)	(99.26)	(98.15)	(100.00)	(99.07)
	Mean	-458.5	839.20	199.31	655.95	400.72	545.35	493.80	568.48
		(74.78)	(76.14)	(95.80)	(93.64)	(98.85)	(98.58)	(99.87)	(99.72)
MMLIB100	0	20585.71	20542.29	20336.36	20367.33	20376.90	20418.60	20420.95	20422.45
		(57.78)	(71.85)	(83.89)	(93.52)	(95.37)	(100.00)	(98.89)	(100.00)
	20	12540.41	12449.99	12328.69	12320.26	12345.01	12369.36	12395.03	12396.55
		(55.74)	(70.93)	(83.52)	(93.70)	(95.74)	(99.63)	(99.81)	(99.81)
	40	3956.63	3910.73	3927.86	3900.43	3938.55	3962.74	3967.83	4004.94
		(57.59)	(68.15)	(82.96)	(94.07)	(95.74)	(99.26)	(99.07)	(100.00)
	60	-715.07	-739.05	-691.56	-691.27	-653.10	-617.56	-604.92	-533.77
		(60.37)	(50.00)	(82.96)	(77.04)	(95.19)	(90.74)	(99.44)	(97.41)
	80	-7812.10	-7772.00	-7641.58	-7525.79	-7530.55	-7369.58	-7400.50	-7146.15
		(64.26)	(56.11)	(84.81)	(80.19)	(96.85)	(92.04)	(99.63)	(97.96)
	100	-19374.96	-19341.18	-19058.03	-18921.92	-18802.96	-18500.46	-18500.60	-18085.08
		(70.19)	(58.89)	(89.44)	(83.33)	(98.15)	(93.89)	(100.00)	(99.26)
	Mean	-2200.15	2696.03	-1143.07	2285.26	223.18	2078.43	1211.43	1905.87
		(60.98)	(62.65)	(84.60)	(86.97)	(96.17)	(95.92)	(99.47)	(99.07)
MMLIB+	0	10670.37	12106.02	10664.51	12311.40	10918.75	12470.47	11200.47	12621.05
		(53.00)	(44.94)	(83.15)	(82.50)	(90.14)	(96.64)	(84.48)	(99.44)
	20	5943.97	6856.97	5942.72	7015.80	6182.46	7135.84	6370.01	7256.07
		(52.53)	(42.22)	(84.58)	(81.02)	(83.92)	(96.33)	(81.40)	(99.41)
	40	1322.84	1658.05	1317.11	1762.87	1497.30	1875.57	1591.94	1974.36
		(55.78)	(39.66)	(85.18)	(78.46)	(79.72)	(94.44)	(76.75)	(98.83)
	60	-773.28	-742.65	-732.61	-670.55	-682.36	-567.33	-611.24	-422.56
		(57.94)	(26.94)	(85.72)	(54.60)	(79.17)	(76.27)	(75.51)	(90.19)
	80	-3790.05	-4119.82	-3670.24	-4012.55	-3788.72	-3907.09	-3532.89	-3710.92
		(62.74)	(28.70)	(86.92)	(56.14)	(79.26)	(78.58)	(67.72)	(92.19)
	100	-10159.73	-11051.61	-9995.35	-10775.07	-10146.73	-10504.00	-9561.32	-10215.78
		(69.84)	(30.31)	(84.68)	(58.58)	(79.32)	(80.59)	(67.20)	(92.87)
	Mean	-12396	2002.44	-2613	2016.35	-437.80	1669.41	922.41	1470.68
		(58.63)	(35.46)	(85.03)	(68.54)	(81.92)	(87.14)	(75.5)	(95.49)

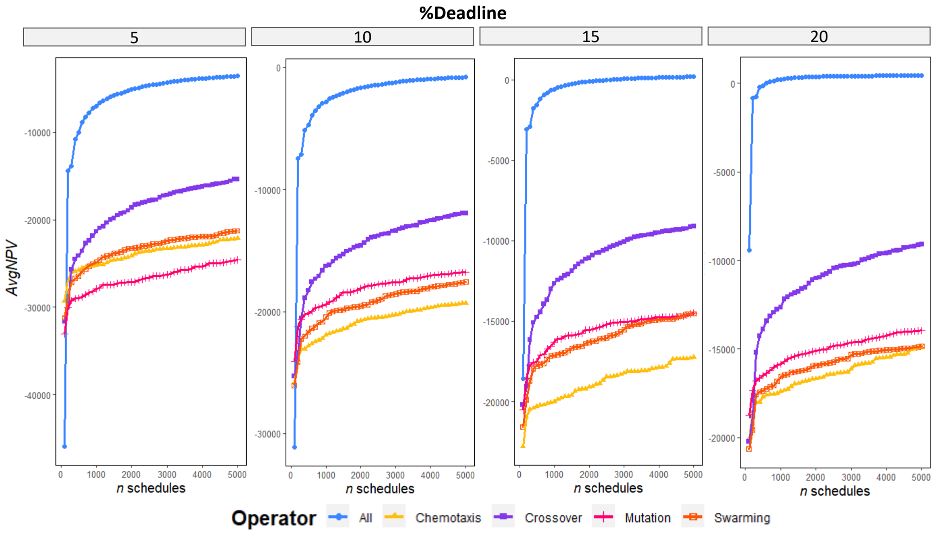


Figure 4.3: Performance of the ABFO algorithm when different operators are used for the MMLIB50 library.

4.5 Chapter Summary

In this paper, a metaheuristic solution algorithm for solving the multi-mode resource-constrained project scheduling problem (MRCPSP) with discounted cash flows (MRCPSPDC) is proposed. This problem consists of determining a schedule such that the project is completed, maximizing the project's net present value (NPV) while complying with the delivery deadline. The adaptative bacterial foraging optimization (ABFO) algorithm is a variation of the original bacterial foraging optimization (BFO), which is a nature-inspired metaheuristic optimization algorithm. We implement a version of the chemotactic operator based on a double justification of the activities given the cash flow. This metaheuristic has been tested in the PSPLIB and MMLIB benchmark datasets available in the literature with promising results. Our ABFO algorithm shows excellent performance in all tested instances and provides suitable solutions for the MRCPSP maximizing the NPV.

Chapter 5

The Reactive MRCPSP

In most real-life projects, the construction, scheduling, and execution of a schedule is a reactive process, where uncertain events force one to change and reconsider the established schedule. In this paper, we present a novel metaheuristic strategy based on bacterial foraging optimization to solve a multi-objective reactive scheduling problem. The performance of the algorithm is measured by the experimental use of a case study. The experimental result shows that the proposed algorithms can reduce the cost, make the time elapse and also improve the reliability. In most real-life projects, the construction, scheduling, and execution of a schedule is a reactive process, in which uncertain events force to change and reconsider the established schedule. In this paper, we present a novel metaheuristic strategy based on bacterial foraging optimization to solve a multi-objective reactive scheduling problem. The performance of the algorithm is measured by the experimental use of a case study. The experimental result shows that the proposed algorithms can reduce the cost, make the time elapsed and also improve the reliability.

5.1 Introduction

Uncertainty is one of the most important concerns in the field of project scheduling, since it generates instability and alterations in the execution of its activities, not achieving the objectives and if possible, canceling the project. Many research works focus on finding a schedule for a project that does not take into account uncertainty, a situation that is unrealistic in a dynamic environment. For an extensive review of the literature refer to (Erik L. Demeulemeester, 2002) and (Herroelen and Leus, 2004).

In this chapter, we develop a metaheuristic strategy to solve the multiple-mode resource-constrained project scheduling problem when considering events that cause schedule disruptions. We performed validation of our procedure with a case study examining the possible options that a client and a contractor have when an agreement must be reached to obtain the best dividends. In our case, we are concerned with optimizing two objective functions. The first one minimizes the cost of rescheduling activity and changing its execution mode. The second objective function optimizes the net present value of a project.

5.2 Problem description

It is possible that during project implementation, resource levels per period may be allowed to vary from time to time, but not less than the minimum resource requirement of any activity in the set of eligible activities. This is in order to allow for lower resource levels from period to period due to interruptions. In addition, some activities may be interrupted in their execution and may need to be postponed for completion. Before scheduling any activity, it is verified that there are no variations in the availability of resources and changes in their duration. If there are changes, the initial interruption time D_s and a final interruption time D_f must be known. This produces a window $\Delta = D_f - D_s$. This is in order to respond to the interruption and take the necessary measures. We assume that during the execution of the project, no activity can be started before its scheduled start. This in order not to start activities earlier than planned. As a consequence of this constraint, the reactive start times s_i' must always be at least as large as their initial schedule equivalents s_i'. In addition, a cost is induced by increasing the start time of activity i per unit of time through a non-negative weight w_i. Additionally, delaying activities to repair the interrupted schedule may cause changes in the modes of the activities, which causes a mode change cost c_i. If the activity does not change its mode, the mode change cost is zero. The above can be summarized:

- Before scheduling an activity i, we verify that there are no variations in resource consumption and duration. If there are changes, $r^{\tau}_{i,m,\ell} \to \hat{r}^{\tau}_{i,m,\ell}$ and/or $d_{i,m} \to \hat{d}_{i,m}$
- Given a schedule $S = (s_0, s_1, \dots, s_{n+1})$ and a vector of modes $m = (m_0, m_1, \dots, m_{n+1})$, the objective function cost of resequencing F_1 will be minimizing
$$F_1(S) = \sum_{i=1}^{n} w_i |s_i' - s_i| + \sum_{i=1}^{n} c_i \tag{5.1}$$
- Each activity is associated with a cash inflow and outflow, respectively $c_{i,in} > 0$ and $c_{i,out} < 0$. The discount rate is $\alpha > 0$ and f_i is completion time of the activity $i \in V$. Here we use following mathematical model based on the PCA payment model (Vanhoucke, Demeulemeester, and Herroelen, 2001; Leyman and Vanhoucke, 2016), where we maximize
$$F_2(S) = \sum_{i=1}^{n} (c_{i,in} + c_{i,out}) \cdot e^{-\alpha f_i} \tag{5.2}$$

subject to

$$f_i \leq f_j - d_{j,m_j}, \quad \forall (i,j) \in E, \tag{5.3}$$

$$\sum_{i \in \mathcal{A}(F,M,t)} r^{\tau}_{i,m_i,k} \leq R^{\tau}_k, \quad k \in R^{\tau}, 0 \leq t \leq \bar{d}, \tag{5.4}$$

$$\sum_{i=1}^{n} r^{\eta}_{i,m_i,\ell} \leq R^{\eta}_{\ell}, \qquad \ell \in R^{\eta}, \tag{5.5}$$

$$f_{n+1} \leq \delta_{n+1} \tag{5.6}$$

$$m_i \in M_i, \qquad i \in V \tag{5.7}$$

$$f_i \in int^{+}, \qquad i \in V \tag{5.8}$$

Here, $\mathcal{A}(F, M, t)$, also called the active set, is a set of real activities which will be sequenced in time t; $M = (m_i)_{i \in V}$ is the mode vector ; $F = (f_i)_{i \in V}$ is a vector of completion times for each activity and $\bar{d} = \sum_{j \in V} \max_{m \in M_j} d_{j,m}$.

The objective function F_1 minimizes the rescheduling cost, while F_2 maximizes the NPV of the project by discounting the cash inflows and outflows at the time of completion of each activity. The constraints represented by equation (5.3) describe the precedence relationships between activities; equation (5.4) ensures that the availability of renewable resources is not exceeded in each period, while equation (5.5) ensures that non-renewable resources are not exceeded throughout the project. The constraints in equation (5.6) ensure that the project delivery time is met, and those in equation (5.7) ensure that each activity is assigned only one mode. The constraint in (5.8) states that the decision variables must be integers. The start and finish dummy activities are assumed to run only in a single mode of zero duration and do not consume resources.

5.3 Multi-objective optimization for PSP

A multiobjective optimisation problem (MOP) can be formulated as:

$$\min_{\mathbf{x} \in D} \mathbf{F}(\mathbf{x}) \text{ with } \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})] \tag{5.9}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ is a vector of decision variables (decision vector), **F(x)** is the objetive vector, D is the set of feasible solutions, and $\mathbf{F}(D) \subset \mathbb{R}^n$. A feasible solution $\mathbf{a} \in D$ is called efficient if there is no $\mathbf{b} \in D$ such that $f_k(\mathbf{b}) \leq f_k(\mathbf{a})$ for all $k = 1 \ldots, n$ and $f_j(\mathbf{b}) < f_j(\mathbf{a})$ for some j. In other words, no solution is strictly better than **a** for at least one criterion and not worse in the remaining criteria. If **a** is efficient then $\mathbf{Y} = \mathbf{F}(\mathbf{a})$ is called non-dominated. The set of efficient solutions is D_E, the set of non-dominated vector is $\mathbf{Y}_N = \mathbf{F}(\mathbf{D_E})$

Let us define the following important concepts in multiobjective optimization.

1. *Definition 1- Pareto Dominance*
   Given the feasible solutions x and x', it is found:
   1.1 if $f_k(x) \leq f_k(x')$ for all $k = 1, 2, \dots, l$ and $f_j(x) < f_j(x')$ for any j, x will will be a solution that dominates x';
   1.2 if $f_k(x') \leq f_k(x)$ for all $k = 1, 2, \dots, l$ and $f_j(x') < f_j(x)$ for any j, x will be a solution dominated by x';
   1.3 if $f_j(x) < f_j(x')$ for any j e $f_i(x) > f_i(x')$ for any i, x and x' are stated non-dominated or indifferent.
2. *Definition 2- Pareto Optimality*
   A feasible solution x is named Pareto-optimal (or efficient) if there is no other feasible solution x' such as x' dominates x, that is, a solution x' such as $f_k(x') \leq f_k(x)$ for all $k = 1, 2, \dots, l$ and $f_j(x') < f_j(x)$ for any j.
   The set of all Pareto-optimal solutions is termed Pareto-optimal front and as a result of the defined concepts, all the solutions that belong to the Pareto-optimal front are non-dominated (indifferent).
   In all the algorithms proposed in this work the criterion of the Pareto Dominance was used, as described in this section, to assess the solutions generated along with its iterations and to determine the set of non-dominated solutions, denoted by D', to be returned by the algorithms.

5.4 Multi-objetive Bacterial Foraging Optimization

The results of single-objective project scheduling (PS) problems are usually expressed in terms of deviations from target values compared to the optimum if it exists or to the best bounds otherwise. However, multi-objective results should be given special attention as the comparison methodologies are numerous.

Since BFO algorithms can solve single-objective optimization problems, shown in chapter 4. We are inspired to solve multiobjective optimization problems with BFO algorithms because of Ballestín and Blanco, 2011 and Niu et al., 2013. However, the purpose of multiobjective optimization problems is to find all solutions that satisfy several objective functions simultaneously. Since different decision-makers have different ideas about the objective functions, it is not easy to choose a unique solution for a multiobjective optimization problem without interacting with the decision-makers. Therefore, all we can do is to show the set of Pareto-optimal solutions to the decision-makers. The main goal of multi-objective optimization problems is to obtain a non-dominated front that is close to the true Pareto front. In this paper, a total rescheduling bacterial foraging multiobjective optimization (MBFO) with integration between the health ordering approach and the Pareto dominance mechanism is proposed to solve multiobjective problems. The goal is to provide the decision-maker with a good set of efficient solutions concerning two objectives. The first objective is the maximization of the net present value of the project, and the second is a minimization of the outage measure. The details of the new BFO-based optimization algorithm are presented in the following sections.

5.4.1 Description of Multi-objective Bacterial Foraging Optimization

We will use the description of the GBFO metaheuristic presented by Passino, 2010. We define a chemotactic step to perform a turn or swim. Let j be the index for the chemotactic step. Let k be the index for the reproduction step and l be the index for the elimination-dispersion step. Let us define

$$P(j,k,l) = \{\theta^i(j,k,l) | i = 1,2,\ldots,S\}$$

represents the position of each member in the colony of the S bacteria at the $j-$th chemotactic step, $k-$th repoduction step, and $l-$th elimination-dispersion step. Let N_c be the lifetime of the bacteria, measured by the number of chemotactic steps they take during their lifetime. Let $C(i) > 0$, $i = 1,2,\ldots,S$ denotes a basic chemotactic step size that we will use to define the length of steps during runs. To represent a drop, a random direction of unit length is generated, say $\varphi(j)$; this will be used to define the direction of motion after a drop. In particular, we let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\varphi(j) \tag{5.10}$$

such that $C(i)$ is the step size taken in the random direction specified by the twist. If at $X^i(j+1,k,l)$ the cost $J(i,j+1,k,l)$ is better(worse) than $X^i(j,k,l)$, then another step size $C(i)$ is taken in this same direction, and again, if that step resulted in a position with a better cost value than in the previous step, another step is taken. This form of swimming is continued as long as it continues to reduce the cost, but only up to a maximum number of steps, N_s. This represents that the bacterium will tend to keep moving if it moves toward increasingly favorable environments.

During movements, bacteria release attractant and repellent factors to send signals to other bacteria to cluster together or move away, provided they obtain a nutrient-rich environment or avoid the noxious environment. Bacteria-to-bacteria attraction and repellent effects are denoted by:

$$\begin{aligned} J_{cc}(\theta, P(j,k,l)) = &\sum_{i=1}^{S} J_{cc}(\theta, \theta^i(j,k,l)) \\ &\sum_{i=1}^{S} \left[-d_{attract} \exp\left(-w_{attract} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2 \right) \right] \\ &+ \sum_{i=1}^{S} \left[h_{repelent} \exp\left(-w_{repelent} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2 \right) \right] \end{aligned} \tag{5.11}$$

where $J_{cc}(\theta, P(j,k,l))$ is the cost function value to be added to the actual cost function to be minimized to present a time varying cost function, S is the total number of bacteria, p is the number of parameters to be optimized which are present in each bacterium, and $d_{attract}$, w_{attact}, $h_{repelent}$ and $w_{repelent}$ are different coefficients that are to be chosen properly.

We highlight that $J_{cc}(X, P(j,k,l))$ is the value of the total time-varying cost for the bacterium X to be added to the cost function to be optimized, resulting in the initial cost with swarming effect given by:

$$J^{sw}(X,j,k,l) = J(X,j,k,l) + J_{cc}(X, P(j,k,l)) \tag{5.12}$$

After the chemotactic step, the reproduction-crossover step begins. Consider the number of replicates to be performed. For convenience, it will be assumed that S is a positive integer divisible by the value of 5.12. Let be

$$S_r = \frac{S}{2} \tag{5.13}$$

is the number of members of the population that have had sufficient nutrients to have the crossover operator applied to them.

Finally, let N_{ed} be the number of elimination-scattering events, and for each elimination-scattering event, each bacterium in the population undergoes elimination-scattering with probability P_{ed}. This helps to keep track of the sudden change in the environmental condition, which may affect the life of the bacteria, so a new set of bacteria can be introduced into the search domain.

5.4.2 MBFO algorithm

Below is a step-by-step description of the MBFO, Niu et al., 2013:

Step 1 Initialing parameters p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, $C(i)$ with $i = 1, 2, \ldots, S$, θ^i where,

p: Dimension of the search space;

S: The number of bacteria;

N_c: Chemotaxis steps;

N_s: Swim steps;

N_{re}: Reproductive steps;

N_{ed}: Elimination and dispersal steps;

P_{ed}: Probability of elimination;

$C(i)$: The run-length unit during each run or tumble;

Step 2. Elimination-dispersal loop: $l = l + 1$.

Step 3. Reproduction loop: $k = k + 1$.

Step 4. Chemotaxis loop: $j = j + 1$.

a. For $i = 1, 2, \dots, S$, take a chemotactic step for bacteria i as follows.

b. Compute two fitness functions $J_1(i, j, k, l)$ y $J^2(i, j, k, l)$

c. Let $J_{last1} = J_1(i, j, k, l)$, $J_{last2} = J_2(i, j, k, l)$ to save the value since we may find better value via a run.

d. Tumble: Generate a random vector $\Delta(i) \in R^n$ with each element $\Delta_m(i) \in [-1, 1]$, $m = 1, 2, \dots, S$.

e. Move: Update of the position with

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \cdot \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}. \tag{5.14}$$

$c(i)$ is the length of unit walk and $\Delta(i)$ is the direction angle of the j^{th} step.

f. Compute each fitness function $J_t(i, j+1, k)(t = 1, 2)$ as Equation 5.11 and 5.12 with $\theta^i(i, j+1, k, l)$.

g. Swim:

i) Let $m = 0$ (counter for swimm length)

ii) While $m < N_s$ (if have not climbed down too long)
Let $m = m + 1$.
If $J_1(i, j+1, k, l) < J_{last1}$, let $J_{last1} = J_1(i, j+1, k, l)$
If $J_2(i, j+1, k, l) < J_{last2}$, let $J_{last2} = J_2(i, j+1, k, l)$
Then another step of size $C(i)$ in this same direction will be taken as Equation 5.14 and use the new generate $\theta^i(i, j+1, k, l)$ to compute the $J_{lasti} = J_i(i, j+1, k, l)$, $(i = 1, 2)$.
Else, sea $m = N_s$. This is the end of the while statement.

h. Go to next bacterium $(i + 1)$. **If** $i \neq S$ (i.e. go to (**b**)) to process the next bacterium).

Step 5 If $j < N_c$, go to step 3. In this case continue chemotaxis since the life of the bacteria is not over.

Step 6 Reproduction:

a) For the given k and l, and for each i=1,2,...,S, let J_{health} be the health of the i^{th} bacteria. The bacteria are sorted i nthe order of ascending values.

$$\begin{aligned} J_{health1} &= \sum_{j=1}^{N_c+1} J_1(i, j, k, l) \\ J_{health2} &= \sum_{j=1}^{N_c+1} J_2(i, j, k, l) \end{aligned} \tag{5.15}$$

b) The bacteria with the highest J_{health} values also dominated die, and the other non-dominated bacteria with best J_{health} values reproduce. The number of the die individuals is no more than S_r then copy the best bacteria in order of keeping the group number unchangeable.

Step 7 If $k < N_{re}$ go to step 2. In this case, the number of specified reproduction steps is not reached and start the next generation in the chemotactic loop.

Step 8 Elimination-dispersal: For $i = 1, 2, \ldots, S$, with probability P_{ed}, eliminate and disperse each bacterium, which results in keeping the number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If $l < N_{ed}$, then go to step 2; otherwise end.

5.5 Methodology for solving reactive scheduling in MR-CPSP

Before starting the project, a complete analysis of the project network is proposed: duration of the activities, modes of execution, availability of resources (renewable and non-renewable), deadline, costs for rescheduling activities, costs for changing the mode of execution, cash flow in the activities, etc. From the results of the analysis, the client and contractor will be able to know a baseline schedule for the project, the total estimated cost of the project, the minimum amount needed per period for the activity to be executed in that period, and the completion time. For uncertainty events that produce disruptions, (Le Chang and Prokopenko, 2017) shows that a Nash equilibrium between the client and contractor when project delays occur is obtained when:

- The client grants an extension and the contractor does not take corrective action.
- The client does not grant an extension and the contractor must take corrective action.

With the first option, project abandonments can be reduced and the objective for the client is to avoid interruptions and scheduling activities in a way that minimizes the objectives 5.1 and 5.2. For the second option, where no rollover is granted, it becomes a much more difficult multi-objective problem to solve. In our case, we will only solve the first case.

Consider the following pseudocode to minimize the project completion time initially:

Step 1. Write project data: list of activities, durations, execution modes, resource consumption, costs, cash flow, precedence relationships, etc.

Step 2. We use the memetic algorithm to find a solution that minimizes the project duration.

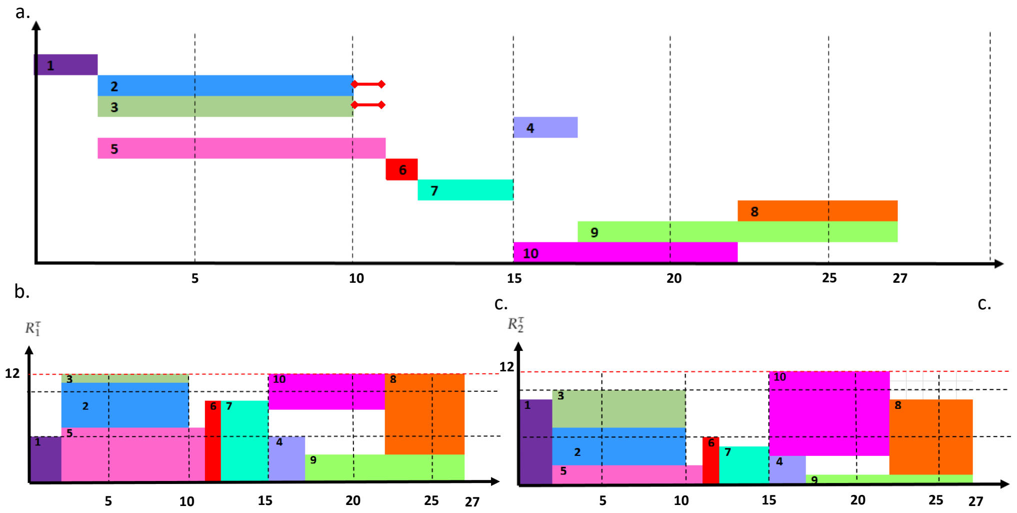


Figure 5.1: Gantt chart and consumption of resources

Activities	1	2	3	4	5	6	7	8	9	10
$c_{i,in}(c_{i,out})$	227	426	-127	323	-263	344	250	-469	-228	-25
$c_{i,m}$	1	5	1	4	1	1	4	3	1	2
w_i	2	1	2	2	5	1	7	4	8	10

Table 5.1: Rescheduling costs and cash flow

Step 3. We make the Gantt chart and set a deadline with the MA solution as the completion date.

Step 4. If interruptions are established in the project, the ABFO is executed by varying the deadline by 5, 10, 15 and 20 percent. If the possible interruptions lead to not meeting the deadline, the project is not executed. Otherwise, the contractor has a wider margin for scheduling the project, optimizing the net present value and avoiding or rescheduling interruptions.

5.6 Experiments

For the validation of the proposed solution method, one experiment was used: the project data from the PSPLIB library Table 3.2, denoted by j1037_2.mm. Figure 5.1, a) shows the Gantt chart with a duration of 27 units and part b) and c) the consumption of renewable resources.

Consider the Table 5.1, which shows the positive and negative cash flows of the activities, to calculate the NPV. In addition, we observe the w_i values that correspond to the weights for rescheduling activity i. The values $c_{i,m}$ correspond to the cost of changing mode when the interruption occurs, in order to consume less resources.

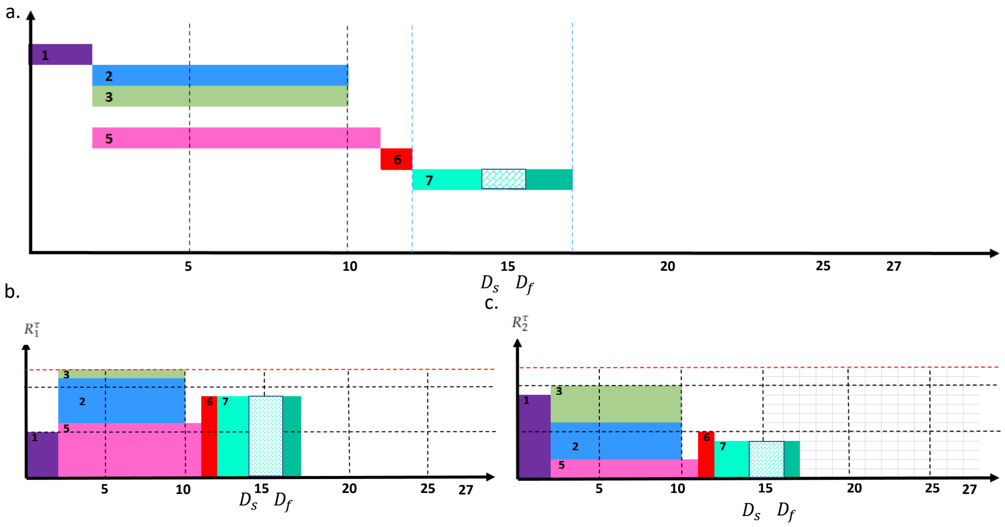


Figure 5.2: Duration disruption

5.6.1 Interruption of duration

Suppose that during the execution of an activity i with duration $d_{i,m}$, such that $m \in M_i$; an interruption occurs, altering the activity and failing to execute it during a period of time between D_s and D_f, where $s_i <= D_s$ and $D_f < f_i = s_i + d_{i,m}$. In Figure 5.2, we observe the execution of the activities partially before the interruption of activity $i = 7$, for which there is an interruption at $t = 14$, with a duration of two units, i.e., activity 7 will finish in 17-time units. Running the MBFO algorithm we calculated the NPV=499.36 and a rescheduling cost of 48 units. Running the MBFO algorithm with a 10% increase in deadline, we find a schedule with $NPV = 499.36$ and a rescheduling cost of 48 units, see Figure 5.3.

5.6.2 Disruption of resources

We will now consider an interruption of a renewable resource k^τ. For this outage we must make some assumptions:

- In a time window $[D_s, D_f]$, the resource $k \in R^\tau$ decreases in Δ^k units.
- $R_k^\tau - \Delta_k^\tau \geq \min\limits_{i \in Eleg} \{ \min\limits_{m \in M_i} \{ r_{i,m,k}^\tau \} \}$.
- When activities are interrupted, they must be carried out where they were at the time of interruption.

Suppose that after the execution of activity 6, we have an interruption of resource 1, between $D_s = 12$ and $D_f = 14$. Such an interruption drops the resource availability by 6 units. That is to say, $R_1^\tau - \Delta_1^\tau = 12 - 6 = 6 > \min\{r_{4,3,1}^\tau, r_{7,1,1}^\tau, r_{10,2,1}^\tau\}$. This

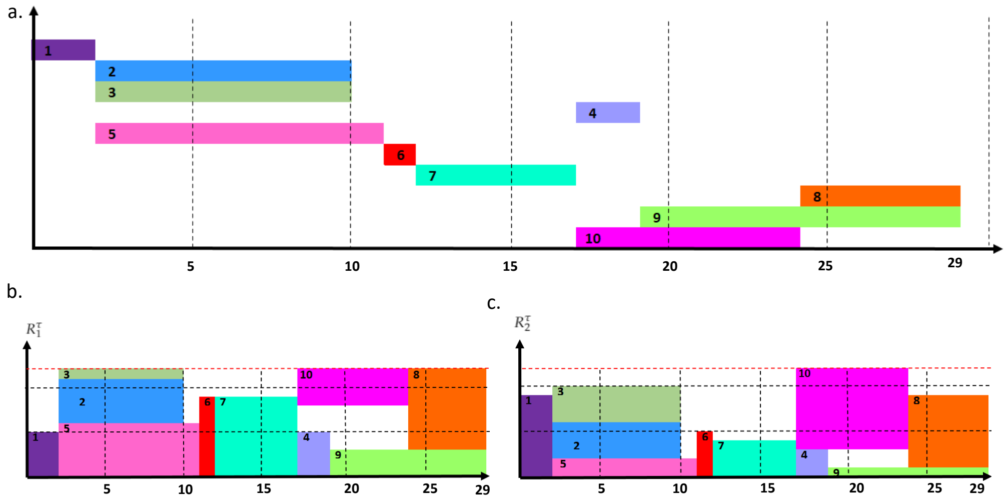


Figure 5.3: Duration disruption with 10% deadline

interruption can be seen in Figure 5.4. It can be seen in Figure 5.5 with a deadline percentage increased by 20%, the $NPV = 519.78$ and a rescheduling cost of 58 units.

5.6.3 Performance metric

In order to have quantitative performance of our algorithm, first we consider two metrics for spacing and diversity of nondominated Pareto solutions and then we report the results of our algorithm.

Spacing (sp)

Schott, 1995 proposed this metric to measure how well the solutions are distributed.This metric measures the (distance) variance of neighboring vectores in non dominated vectores and it is definded as:

$$sp = \frac{1}{n-1}\sqrt{\sum_{i=1}^{n}(\bar{d} - d_i)^2} \tag{5.16}$$

where $d_i = \min_{1 \le j \le n} \{|F_1^i - F_1^j| + |F_2^i - F_2^j|\}$, $i, j = 1\ldots, K$ and K is the number of non-dominated solutions generated by the algorithm. In the case of $\bar{d}$ is the mean of all d_j. A smaller sp value corresponds to better algorithm performance and a value of zero for this metric shows that all non-dominant solutions found are equidistant.

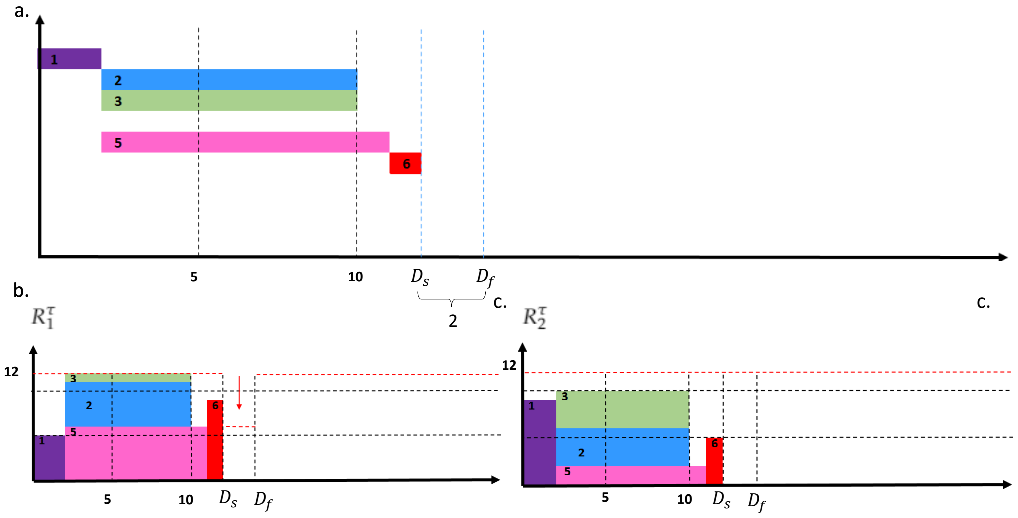


Figure 5.4: Disruption of resources

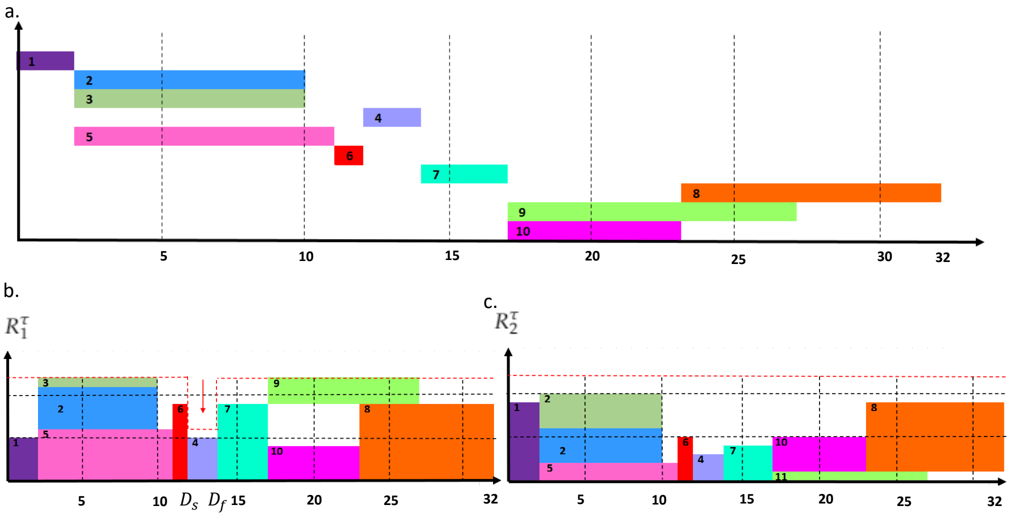


Figure 5.5: Disruption of resources with 20% deadline

Costo	NPV	Deadline	CashFlow	Feasible
85	-1.0317	0.15	0	1
172	-5.12094	0.15	0	1
192	-2319.45	0.15	0	1
149	-55.4905	0.2	0	1
252	-2319.45	0.2	0	1
138	-1263.82	0.15	0.2	1
162	-1268.91	0.15	0.2	1
162	-1268.91	0.2	0.2	1
189	-1269.04	0.2	0.2	1
88	-522.655	0.15	0.4	1
112	-530.092	0.15	0.4	1
138	-522.655	0.2	0.4	1
162	-530.092	0.2	0.4	1
885	-530.092	0.15	0.6	1
78	-522.655	0.2	0.6	1
102	-530.092	0.2	0.6	1
120	-355.588	0.2	0.6	1
192	-501.974	0.2	0.6	1
211	-506.033	0.2	0.6	1
231	-510.071	0.2	0.6	1
888	-77.6447	0.15	0.8	1
912	-85.0818	0.15	0.8	1
521	-77.6447	0.2	0.8	1
545	-85.0818	0.2	0.8	1
122	2468.12	0.15	1	1
135	2288.99	0.15	1	1
244	2228.54	0.2	1	1

Table 5.2: Pareto front with % deadline and % CashFlow for feasible solutions

Maximum dispersion (md)

According to Zitzler and Thiele, 1998, the maximum dispersion is used to measure the diversity of the non-dominated front obtained. This metric is defined as

$$md = \sqrt{|F_1^{\max} - F_1^{\min}| + |F_2^{\max} - F_2^{\min}|} \tag{5.17}$$

where $F_i^{\max}, F_i^{\min}$ represent the maximum respectively minimum value for the objective functions.

In the Table 5.2, 5.3 we show the algorithm measure considering the previously described metrics. For which, the measure sp is maximum 31.01 and for md we have a maximum value of 49.85, which is a quite high value considering the number of values on the Pareto frontier.

%Neg	No. of non-dominated solutions	sp	md
0	5	6.39	49.853
20	4	0.438	7.49
40	4	0.24	9.65
60	5	37.01	17.53
80	2	0	5.6
100	3	6.12	19.015

Table 5.3: Result of number of solutions not mastered and performance measures.

5.7 Chapter Summary

Project delays, cancellations or interruptions are the worst-case scenarios a project manager can handle. In this chapter we develop a new bacterial foraging optimization (BFO) approach to multiobjective optimization, called multiobjective bacterial foraging optimization (MBFO). The objectives in multiobjective bacterial foraging optimization are to minimize the cost of rescheduling and maximize the net present value. The main goal of MBFO is to integrate with reactive scheduling to find a non-dominated upper front, trying to close to the true Pareto front. Changes in the $deadlineand$ cashflow are guaranteed to find feasible solutions. We performed our experiment with a very popular project from the PSPLIB library to refine our algorithm. Our results show that resource interruptions yield higher solution complexity for projects that consider interruptions. It is worth noting that reactive scheduling problems are an important field for research and development of new methods. and development of new methods.

Chapter 6

Discussions and conclusions

The last chapter of this thesis illustrates an argumentation of the results obtained in the course of the research and the subsequent analysis performed; the narrative is based on the original aims and objectives of the study and considers the implications, limitations, and contribution to knowledge. Recommendations for future research, the importance of reactive scheduling in projects as an approach to deal with uncertainty, and the problem of scheduling projects with limited multiple-mode resources are demonstrated and explained. for this, it is necessary to analyze the behavior and characteristics of the projects to be scheduled. Reactive scheduling refers to a whole plan that goes from the beginning of the scheduling of an initial baseline schedule to dealing with different interruptions that cause analysis and control of the project, to then provide a final schedule to execute it.

The present research works with the multiple-mode resource-constrained project scheduling problem taking into account uncertain events once the scheduling of activities is initiated. For this purpose, a taxonomy and a literature review were developed to describe and reduce the complexity of identifying studies related to the research. This review groups and classifies previous works. On the other hand, a memetic algorithm was designed, coded, and compared to solve the MRCPSP with an objective function to minimize the project duration. For the MRCPSP with an objective function to maximize the net present value, an algorithm based on bacterial foraging was presented and with this same idea, a multiobjective BFO was developed to minimize the cost of rescheduling and to maximize the net present value.

For the MRCPSP problem with makespan minimization, the memetic algorithm, combines evolutionary algorithms, local search (LS), and variable neighborhoods search (VNS), which is a generalization of the uniform crossover operator, and a mutation operator used to minimize the consumption of non-renewable resources. We developed an operator that combines the exploration of VNS by changing the list of activities with time improvement provided by multi-mode justification. Separately, these approaches have contributed to high-quality solutions to large problems. The MRCPSPDC which considers the project's NPV and uses a PAC payment model as the objective function. The interaction between the execution time and cash inflows or outflows is dynamic. Finding a balance between time/cost has been a priority for project managers. For this, it is necessary to have tools that help managers and engineers to make decisions. Here we proposed an adaptative bacterial foraging optimization (ABFO) algorithm for such purpose. In this algorithm, we implemented the chemotaxis, swarming, crossover and mutation operators that explore and exploit the solution space to consider the positive and negative cash flow of the activities. Finally, the MBFO algorithm is a novel algorithm that uses the chemotaxis and swarming operators to find a feasible schedule for randomly presented interruptions. By integrating these three concepts, the methodology for reactive scheduling can be built.

For each algorithm, we performed a robust design using the PSPLIB and MMLIB test libraries. For the MA case, as the ABFO in terms of quality and capacity is quite good, being able to achieve excellent results on ensembles with a larger number of activities. The results can be further improved as more time and computational capacity becomes available. The MBFO was run with an experiment, where different metrics were applied to compare the non-dominated solutions and we report their results.

It would be pertinent to create a neural network for project scheduling problems with risk assessment components. For the case of the evaluated algorithms, increase the number of objective functions and have different evaluation criteria. For this, it is necessary to implement several strategies based on metaheuristics, hyper-heuristics, and matheuristic, considering several uncertain events, to adjust the problem more to reality.

Future research will focus on MRCPSP with reactive scheduling applied to different scenarios. For example, the assignment of vessels to berths in ports, the competitive game between client and contractor in project execution, supply chain planning, and the integration of different tools to commercial software. Generally, these scenarios have a larger number of activities and resources, which can lead to an increase in the complexity of the problem. Therefore, more research is needed in the application of MRCPSP and its various solution strategies.

Bibliography

Abdolshah, Mohammad (2014). “A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions”. In:

Aboutalebi, RS, AA Najafi, and B Ghorashi (2012). “Solving multi-mode resource-constrained project scheduling problem using two multi objective evolutionary algorithms”. In: *African Journal of Business Management* 6.11, p. 4057.

Adamu, Patience I., Isaac I. Akinwumi, and Hilary I. Okagbue (2019). “Reactive project scheduling: minimizing delays in the completion times of projects”. In: *Asian Journal of Civil Engineering*. ISSN: 2522011X.

Ahn, Taeho and S Selcuk Erenguc (1998). “The resource constrained project scheduling problem with multiple crashable modes: a heuristic procedure”. In: *European Journal of Operational Research* 107.2, pp. 250–259.

Alcaraz, Javier, Concepcion Maroto, and Ruben Ruiz (2003). “Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms”. In: *Journal of the Operational Research Society* 54.6, pp. 614–626. ISSN: 0160-5682.

Amador-Fontalvo, Jaime E, Carlos D Paternina-Arboleda, and Jairo R Montoya-Torres (2014). “Solving the heterogeneous vehicle routing problem with time windows and multiple products via a bacterial meta-heuristic”. In: *International Journal of Advanced Operations Management* 6.1, pp. 81–100.

Artigues, Christian, Sophie Demassey, and Emmanuel Neron (2013). *Resource-constrained project scheduling: models, algorithms, extensions and applications*. John Wiley & Sons.

Artigues, Christian, Francois Roubellat, and J. C. Billaut (1999). “Characterization of a set of schedules in a resource-constrained multi-project scheduling problem with multiple modes”. In: *International Journal of Industrial Engineering* 6, pp. 112–122.

Asta, Shahriar et al. (2015). “Combining monte-carlo and hyper-heuristic methods for the multi-mode resource-constrained multi-project scheduling problem”. In: *Information Sciences* 373, pp. 476–498.

Bagheri, M, F Jolai, and MB Aryanezhad (2012). “Stability analysis in project resource levelling problem with stochastic activity durations”. In: *World Applied Sciences Journal* 16.1, pp. 126–134.

Bailey, Kenneth D (1994). *Typologies and taxonomies: an introduction to classification techniques*. Vol. 102. Sage.

Balas, Egon (1968). *Project Scheduling with Reosurce Constraints.* Tech. rep. DTIC Document.

Ballestín, Francisco and Rosa Blanco (2011). “Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems”. In: *Computers & Operations Research* 38.1, pp. 51–62.

Ballestin, Francisco and Norbert Trautmann (2008). “An iterated-local-search heuristic for the resource-constrained weighted earliness-tardiness project scheduling problem”. In: *INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH* 46.22, 6231–6249. ISSN: 0020-7543. DOI: 10.1080/00207540701420560.

Ballestín, Francisco, Vicente Valls, and Sacramento Quintanilla (2008). “Pre-emption in resource-constrained project scheduling”. In: *European Journal of Operational Research* 189.3, pp. 1136–1152. URL: http://www.sciencedirect.com/science/article/pii/S0377221707005905.

Barrios, Agustin, Francisco Ballestin, and Vicente Valls (2011). “A double genetic algorithm for the MRCPSP/max”. In: *Computers & Operations Research* 38.1, pp. 33–43. ISSN: 0305-0548.

Bedworth, David D and James E Bailey (1999). *Integrated production control systems: management, analysis, design*. John Wiley & Sons, Inc.

Benjamini, Yoav and Yosef Hochberg (1995). “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. In: *Journal of the Royal statistical society: series B (Methodological)* 57.1, pp. 289–300.

Bianco, Lucio, Paolo Dell’Olmo, and M Grazia Speranza (1998). “Heuristics for multimode scheduling problems with dedicated resources”. In: *European Journal of Operational Research* 107.2, pp. 260–271.

Billaut, Jean Claude and François Roubellat (1996a). “A new method for workshop real time scheduling”. In: *International Journal of Production Research* 34.6, pp. 1555–1579.

– (1996b). “Characterization of a set of schedules in a multiple resource context”. In: *Journal of Decision Systems* 5.1-2, pp. 95–109.

Biruk, Sławomir and Łukasz Rzepecki (2019). “Simulation model for resource-constrained construction project”. In: *Open Engineering*. ISSN: 2391-5439. DOI: 10.1515/eng-2019-0037.

Blazewicz, Jacek (1986). *Scheduling under resource constraints: Deterministic models*. Vol. 7. JC Baltzer.

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

Boctor, Fayez F (1996). “A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes”. In: *European Journal of Operational Research* 90.2, pp. 349–361.

Boctor, Fayez Fouad (1993). “Heuristics for scheduling projects with resource restrictions and several resource-duration modes”. In: *The International Journal of Production Research* 31.11, pp. 2547–2558.

Bouleimen, KLEIN and HOUSNI Lecocq (2003). “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, pp. 268–281. ISSN: 0377-2217.

Brčić, Mario, Marija Katić, and Nikica Hlupić (2019). “Planning horizons based proactive rescheduling for stochastic resource-constrained project scheduling problems”. In: *European Journal of Operational Research*. ISSN: 03772217. DOI: 10.1016/j.ejor.2018.07.037.

Brčić, Mario and Danijel Mlinarić (2018). “Tracking predictive gantt chart for proactive rescheduling in stochastic resource constrained project scheduling”. In: *Journal of Information and Organizational Sciences*. ISSN: 18469418. DOI: 10.31341/jios.42.2.2.

Briand, Cyril, Emmanuelle Despontin, and François Roubellat (2002). “Scheduling with time lags and preferences: a heuristic”. In: *8th workshop on Project Management and Scheduling, Valencia*, pp. 3–5.

Brucker, Peter et al. (1999). “Resource-constrained project scheduling: Notation, classification, models, and methods”. In: *European Journal of Operational Research* 112.1, pp. 3 –41. ISSN: 0377-2217.

Calhoun, Kevin M et al. (2002). “Planning and re-planning in project and production scheduling”. In: *Omega* 30.3, pp. 155–170.

Chakrabortty, Ripon K, Alireza Abbasi, and Michael J Ryan (2020). “Multi-mode resource-constrained project scheduling using modified variable neighborhood search heuristic”. In: *International Transactions in Operational Research* 27.1, pp. 138–167.

Chakrabortty, Ripon K, Ruhul A Sarker, and Daryl L Essam (2016). “Multi-mode resource constrained project scheduling under resource disruptions”. In: *Computers & Chemical Engineering* 88, pp. 13–29.

Chakrabortty, Ripon K., Ruhul A. Sarker, and Daryl L. Essam (2016). “Multi-mode resource constrained project scheduling under resource disruptions”. In: *COMPUTERS & CHEMICAL ENGINEERING* 88, 13–29. ISSN: 0098-1354. DOI: {10.1016/j.compchemeng.2016.01.004}.

– (2018). “Single mode resource constrained project scheduling with unreliable resources”. In: *Operational Research*. ISSN: 18661505.

Chen, Angela HL and Chiuh-Cheng Chyu (2008). “A memetic algorithm for maximizing net present value in resource-constrained project scheduling problem”. In: *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE, pp. 2396–2403.

Chen, Angela Hsiang Ling, Yun Chia Liang, and Jose David Padilla (2014). “An Entropy-Based Upper Bound Methodology for Robust Predictive Multi-Mode RCPSP Schedules”. In: *Entropy* 16.9, pp. 5032–5067.

Chen, Hanning, Yunlong Zhu, and Kunyuan Hu (2011). “Adaptive bacterial foraging optimization”. In: *Abstract and Applied Analysis* 2011.

Chen, Reuy Maw and Chuin Mu Wang (2013). “Controlling Search Using an S Decreasing Constriction Factor for Solving Multi-mode Scheduling Problems”. In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, pp. 545–555.

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

Chen, Wei-Neng et al. (2010b). "Optimizing discounted cash flows in project scheduling - an ant colony optimization approach". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.1, pp. 64–77.

Chiang, Chuan-Wen and Yu-Qing Huang (2012). "Multi-mode Resource-Constrained Project Scheduling by Ant Colony Optimization with a Dynamic Tournament Strategy". In: *2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications*. IEEE, pp. 110–115.

Chiang, Chuan-Wen, Yu-Qing Huang, and Wen-Yen Wang (2008). "Ant colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling". In: *Journal of Intelligent & Fuzzy Systems* 19.4, 5, pp. 345–358.

Choi, Jaein, Matthew J Realff, and Jay H Lee (2004). "Dynamic programming in a heuristically confined state space: a stochastic resource-constrained project scheduling application". In: *Computers & Chemical Engineering* 28.6, pp. 1039–1058.

Christofides, Nicos, R. Alvarez-Valdes, and J.M. Tamarit (1987). "Project scheduling with resource constraints: A branch and bound approach". In: *European Journal of Operational Research* 29.3, pp. 262 –273. ISSN: 0377-2217.

Coelho, José and Mario Vanhoucke (2011). "Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers". In: *European Journal of Operational Research* 213.1, pp. 73–82. ISSN: 03772217.

Coffman, Edward Grady and Peter J Denning (1973). *Operating systems theory*. Vol. 973. Prentice-Hall Englewood Cliffs, NJ.

Cui, Jianshuang and Liruoyang Yu (2014). "An efficient discrete particle swarm optimization for solving multi-mode resource-constrained project scheduling problem". In: *Ieee*, pp. 858–862.

Damak, N et al. (2009). "Differential evolution for solving multi-mode resource-constrained project scheduling problems". In: *Computers & Operations Research* 36.9, pp. 2653–2659. ISSN: 0305-0548.

Davari, M. and E. Demeulemeester (2019a). "Important classes of reactions for the proactive and reactive resource-constrained project scheduling problem". In: 1-2. cited By 4, pp. 187–210. ISSN: 02545330.

Davari, Morteza and Erik Demeulemeester (2019b). "The proactive and reactive resource-constrained project scheduling problem". In: *Journal of Scheduling* 22.2, pp. 211–237.

Davis, Edward W (1966). "Resource allocation in project network models-A survey". In: *Journal of Industrial Engineering* 17.4, p. 177.

Dayanand, Nalini and Rema Padman (2001). "A two stage search heuristic for scheduling payments in projects". In: *Annals of Operations Research* 102.1, pp. 197–220.

De Reyck, Bert et al. (1998). "A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations". In: *European Journal of Operational Research* 111.1, pp. 152–174.

Debels, Dieter et al. (2006). "A hybrid scatter search/electromagnetism meta-heuristic for project scheduling". In: *European Journal of Operational Research* 169.2, pp. 638–653.

Deblaere, Filip, Erik Demeulemeester, and Willy Herroelen (2011a). "Exact and Heuristic Reactive Planning Procedures for Multimode Resource-Constrained Projects". In: *SSRN Electronic Journal*.

– (2011b). "Reactive scheduling in the multi-mode RCPSP". In: *Computers & Operations Research* 38.1, pp. 63–74.

Deblaere, Filip et al. (2007). "Robust resource allocation decisions in resource-constrained projects". In: *Decision Sciences* 38.1, pp. 5–37.

Delgoshaei, Aidin et al. (2014). "A backward approach for maximizing net present value of multi-mode pre-emptive resource-constrained project scheduling problem with discounted cash flows using simulated annealing algorithm". In: *International Journal of Industrial Engineering and Management* 5.3, pp. 151–158.

Demeulemeester, E. and W. Herroelen (2011). "Robust project scheduling". English. In: *Foundations and Trends in Technology, Information and Operations Management* 3.3. cited By 18, pp. 201–376. ISSN: 15719545.

Eksioglu, Burak, Arif Volkan Vural, and Arnold Reisman (2009). "The vehicle routing problem: A taxonomic review". In: *Computers & Industrial Engineering* 57.4, pp. 1472–1483.

Elloumi, Sonda and Philippe Fortemps (2010). "A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem". In: *European Journal of Operational Research* 205.1, pp. 31–41. ISSN: 03772217.

Elloumi, Sonda, Philippe Fortemps, and Taïcir Loukil (2017). "Multi-objective algorithms to multi-mode resource-constrained projects under mode change disruption". In: *Computers and Industrial Engineering*. ISSN: 03608352.

Elmaghraby, Salah Eldin (1977). *Activity networks: Project planning and control by network models*. John Wiley & Sons.

Erik L. Demeulemeester, Willy S. Herroelen (2002). *Project Scheduling - A Research Handbook*. Ed. by Frederick S. Hillier. Vol. 49. Kluwer Academic Publishers, Boston.

Farhad Habibi, Farnaz Barzinpour and Seyed Jafar Sadjadi (2018). "Resource-constrained project scheduling problem: review of past and recent developments". In: *Journal of Project Management*.

Ge, Hongwei and Guozhen Tan (2012). "A Cooperative Intelligent Approach for Job-shop Scheduling Based on Bacterial Foraging Strategy and Particle Swarm Optimization". In: *Computational Intelligence Systems in Industrial Engineering*. Springer, pp. 363–383.

Geiger, Martin Josef (2017). "A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem". In: *European Journal of Operational Research* 256.3, pp. 729–741.

Gerk, JosÉ Eduardo Vinhaes and Raad Yahya Qassim (2008). "Project acceleration via activity crashing, overlapping, and substitution". In: *IEEE Transactions on engineering management* 55.4, pp. 590–601.

Gomes, Helton Cristiano, Francisco de Assis das Neves, and Marcone Jamilson Freitas Souza (2014). “Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations”. In: *Computers & Operations Research* 44, pp. 92–104.

Gupta, Sandeep Kumar et al. (2019). “Systematic literature review of project failures: Current trends and scope for future research”. In: *Computers & Industrial Engineering* 127, pp. 274–285.

Gürel, Sinan, Ersin Körpeoğlu, and M. Selim Aktürk (2010). “An anticipative scheduling approach with controllable processing times”. In: *Computers and Operations Research*. ISSN: 03050548. DOI: 10.1016/j.cor.2009.09.001.

Hansen, Pierre et al. (2010). “Variable Neighborhood Search”. In: *Handbook of Metaheuristics*. Vol. 146. International Series in Operations Research $&$ Management Science. Springer US, pp. 61–86. ISBN: 978-1-4419-1663-1.

Hartmann, S (1999). *Project scheduling under limited resources: Models, methods, and applications*. Vol. 478. Lecture Notes in Economics and Mathematical Systems. Berlin: Springer-Verlag, pp. xii+221. ISBN: 3-540-66392-4.

Hartmann, S. and A. Drexl (1998). “Project scheduling with multiple modes: A comparison of exact algorithms”. In: *Networks* 32.4, pp. 283–297. ISSN: 0028-3045.

Hartmann, Sönke (2001). “Project Scheduling with Multiple Modes: A Genetic Algorithm”. In: *Annals of Operations Research* 102.1-4, pp. 111–135. ISSN: 02545330.

Hartmann, Sonke and Dirk Briskorn (2010). “A survey of variants and extensions of the resource-constrained project scheduling problem”. In: *European Journal of Operational Research* 207.1, pp. 1–14. ISSN: 0377-2217.

Hartmann, Sonke and Rainer Kolisch (2000). “Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem”. In: *European Journal of Operational Research* 127.2, pp. 394 –407. ISSN: 0377-2217.

Herroelen, Willy (2005). “Project scheduling: Theory and practice”. In: *Production and Operations Management* 14.4, pp. 413–432.

Herroelen, Willy and Erik Demeulemeester (2010). “Forensic project management: avoidance and analysis of project schedule delays”. In: *Production and Inventory Management Journal*. ISSN: 0897-8336.

Herroelen, Willy, Erik Demeulemeester, and Bert De Reyck (1999). “A classification scheme for project scheduling”. In: *Project Scheduling*. Springer, pp. 1–26.

Herroelen, Willy, Erik Demeulemeester, and Bert Reyck (1997). “Resource-constrained project scheduling: A survey of recent developments”. English. In: *Operations Research Proceedings 1996*. Vol. 1996. Operations Research Proceedings. Springer Berlin Heidelberg, pp. 197–202. ISBN: 978-3-540-62630-5.

Herroelen, Willy, Erik Demeulemeester, and Bert De Reyck (2001). “A note on the paper: Resource-constrained project scheduling: Notation, classification, models and methods: by Brucker et al.” In: *European Journal of Operational Research* 128.3, pp. 679 –688. ISSN: 0377-2217.

Herroelen, Willy and Roel Leus (2004). "Robust and reactive project scheduling: a review and classification of procedures". In: *International Journal of Production Research* 42.8, pp. 1599–1620.

– (2005). "Project scheduling under uncertainty: Survey and research potentials". In: *European Journal of Operational Research* 165.2, pp. 289–306.

Herroelen, Willy S, Patrick Van Dommelen, and Erik L Demeulemeester (1997). "Project network models with discounted cash flows a guided tour through recent developments". In: *European Journal of Operational Research* 100.1, pp. 97–121.

Hu, Xuejun, Jianjiang Wang, and Kaijun Leng (2019). "The Interaction between Critical Chain Sequencing, Buffer Sizing, and Reactive Actions in a CC/BM Framework". In: *Asia-Pacific Journal of Operational Research*. ISSN: 02175959. DOI: 10.1142/S0217595919500106.

Hu, Xuejun et al. (2017). "Improved critical chain buffer management framework considering resource costs and schedule stability". In: *Flexible Services and Manufacturing Journal*. ISSN: 19366590.

Icmeli, Oya and S Selcuk Erenguc (1994). "A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows". In: *Computers & operations research* 21.8, pp. 841–853.

Icmeli, Oya, S Selcuk Erenguc, and Christopher J Zappe (1993). "Project scheduling problems: a survey". In: *International Journal of Operations & Production Management* 13.11, pp. 80–91.

Jarboui, Bassem et al. (2008). "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems". In: *Applied Mathematics and Computation* 195.1, pp. 299–308. ISSN: 0096-3003.

Jedrzejowicz, Piotr and Ewa Ratajczak-Ropel (2015). "Reinforcement learning strategy for solving the MRCPSP by a team of agents". In: *Intelligent Decision Technologies*. Springer, pp. 537–548.

Józefowska, Joanna and Jan Weglarz (2006). *Perspectives in modern project scheduling*. Vol. 92. Springer Science & Business Media.

Jozefowska, Joanna et al. (2001). "Simulated annealing for multi-mode resource-constrained project scheduling". In: *Annals of Operations Research* 102.1-4, pp. 137–155.

Kar, Arpan Kumar (2016). "Bio inspired computing–a review of algorithms and scope of applications". In: *Expert Systems with Applications* 59, pp. 20–32.

Ke, Hua and Baoding Liu (2005). "Project scheduling problem with stochastic activity duration times". In: *Applied Mathematics and Computation* 168.1, pp. 342–353.

Kelley Jr, James E and Morgan R Walker (1959). "Critical-path planning and scheduling". In: *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*. ACM, pp. 160–173.

Kelly, JE (1963). "The critical path method: Resource planning and scheduling". In: *Industrial Scheduling*. Ed. by J.F. Muth and G.L. Thompson. New Jersey: Prentice Hall, pp. 347–365.

Khan, Nabeel, Maria G. Martini, and Dirk Staehle (2014). "QoS-aware composite scheduling using fuzzy proactive and reactive controllers". In: *EURASIP JOURNAL ON WIRELESS COMMUNICATIONS AND NETWORKING*. ISSN: 1687-1472. DOI: {10.1186/1687-1499-2014-138}.

Kolisch, Rainer (1996a). "Efficient priority rules for the resource-constrained project scheduling problem". In: *Journal of Operations Management* 14.3, pp. 179 –192. ISSN: 0272-6963.

– (1996b). "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation". In: *European Journal of Operational Research* 90.2, pp. 320–333.

– (2013). *Project scheduling under resource constraints: efficient heuristics for several problem classes*. Springer Science & Business Media.

Kolisch, Rainer and Andreas Drexl (1997). "Local search for nonpreemptive multimode resource-constrained project scheduling". In: *IIE Transactions* 29.11, pp. 987–999. ISSN: 0740-817X.

Kolisch, Rainer and Sonke Hartmann (1999). "Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis". English. In: *Project Scheduling*. Vol. 14. International Series in Operations Research & Management Science. Springer US, pp. 147–178. ISBN: 978-1-4613-7529-6.

– (2006). "Experimental investigation of heuristics for resource-constrained project scheduling: An update". In: *European Journal of Operational Research* 174.1, pp. 23 – 37. ISSN: 0377-2217.

Kolisch, Rainer and Rema Padman (2001). "An integrated survey of deterministic project scheduling". In: *Omega* 29.3, pp. 249 –272. ISSN: 0305-0483.

Kolisch, Rainer and Arno Sprecher (1996). "PSPLIB - A project scheduling problem library". In: *European Journal of Operational Research* 96.1, pp. 205–216. ISSN: 0377-2217.

Kopka, Helmut, Patrick W Daly, and SPQ Rahtz (2004). *Guide to LATEX*. Vol. 4. Addison-Wesley Boston, MA.

Koulinas, Georgios, Lazaros Kotsikas, and Konstantinos Anagnostopoulos (2014). "A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem". In: *Information Sciences* 277, pp. 680–693. ISSN: 00200255.

Kuster, Juergen, Dietmar Jannach, and Gerhard Friedrich (2010). "Applying Local Rescheduling in response to schedule disruptions". In: *ANNALS OF OPERATIONS RESEARCH* 180.1, 265–282. ISSN: 0254-5330. DOI: {10.1007/s10479-008-0488-x}.

Kyriakidis, Thomas S., Georgios M. Kopanos, and Michael C. Georgiadis (2012). "MILP formulations for single- and multi-mode resource-constrained project scheduling problems". In: *Computers and Chemical Engineering* 36.1, pp. 369–385. ISSN: 00981354.

Lamas, Patricio and Erik Demeulemeester (2016). "A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations". In: *JOURNAL OF SCHEDULING* 19.4, 409–428. ISSN: 1094-6136. DOI: {10.1007/s10951-015-0423-3}.

Lambrechts, Olivier, Erik Demeulemeester, and Willy Herroelen (2008). "Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities". In: *Journal of scheduling* 11.2, pp. 121–136.

– (2011). "Exact and Suboptimal Reactive Strategies for Resource-Constrained Project Scheduling with Uncertain Resource Availabilities". In: *SSRN Electronic Journal*.

Lancaster, John and Mustafa Ozbayrak (2007). "Evolutionary algorithms applied to project scheduling problems - a survey of the state-of-the-art". In: *International Journal of Production Research* 45.2, pp. 425–450.

Larson, Erick W and Clifford F Gray (2015). "A Guide to the Project Management Body of Knowledge: PMBOK (®) Guide". In: Project Management Institute.

Laue, HJ (1968). "Efficient methods for the allocation of resources in project networks". In: *Unternehmensforschung* 12.1, pp. 133–143.

Lawler, Eugene L et al. (1993). "Sequencing and scheduling: Algorithms and complexity". In: *Handbooks in operations research and management science* 4, pp. 445–522.

Le Chang, Sheryl and Mikhail Prokopenko (2017). "Instability of mixed Nash equilibria in generalised Hawk-Dove game: A project conflict management scenario". In: *Games* 8.4, p. 42.

Leyman, Pieter and Mario Vanhoucke (2015). "A new scheduling technique for the resource–constrained project scheduling problem with discounted cash flows". In: *International Journal of Production Research* 53.9, pp. 2771–2786.

– (2016). "Payment models and net present value optimization for resource-constrained project scheduling". In: *Computers & Industrial Engineering* 91, pp. 139–153.

Li, Heng and Hong Zhang (2013). "Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints". In: *Automation in Construction* 35, pp. 431–438. ISSN: 09265805.

Li, Zukui and Marianthi Ierapetritou (2008). "Process scheduling under uncertainty: Review and challenges". In: *Computers & Chemical Engineering* 32.4. Festschrift devoted to Rex Reklaitis on his 65th Birthday, pp. 715 –727. ISSN: 0098-1354.

Lova, Antonio, Pilar Tormos, and Federico Barber Sanchís (2006). "Multi-mode resource constrained project scheduling: scheduling schemes, priority rules and mode selection rules." In: *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial* 10.30, pp. 69–86.

Lova, Antonio et al. (2009). "An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes". In: *International Journal of Production Economics* 117.2, pp. 302–316.

Luna, Eduardo Gómez et al. (2014). "Metodología para la revisión bibliográfica y la gestión de información de temas científicos, a través de su estructuración y sistematización". In: *DYNA: revista de la Facultad de Minas. Universidad Nacional de Colombia. Sede Medellín* 81.184, pp. 158–163.

M., Wall (1996). "A genetic algorithm for resource-constrained scheduling." PhD thesis. Massachusetts Institute of Technology.

Ma, Zhiqiang et al. (2019). "A computational experiment to explore better robustness measures for project scheduling under two types of uncertain environments". In: *Computers and Industrial Engineering*. ISSN: 03608352.

Macpherson, Allan and Oswald Jones (2010). "Strategies for the development of International Journal of Management Reviews". In: *International Journal of Management Reviews* 12.2, pp. 107–113.

Majhi, Ritanjali et al. (2009). "Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques". In: *Expert Systems with Applications* 36.6, pp. 10097–10104.

Malcolm, Donald G et al. (1959). "Application of a technique for research and development program evaluation". In: *Operations Research* 7.5, pp. 646–669.

Mika, Marek, Grzegorz Waligóra, and Jan Weglarz (2005). "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models". In: *European Journal of Operational Research* 164.3, pp. 639–668.

Miller, Roger and Donald R Lessard (2001). *The strategic management of large engineering projects: Shaping institutions, risks, and governance*. MIT press.

Mingozzi, Aristide et al. (1998a). "An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation". In: *Management Science* 44.5, pp. 714–729.

– (1998b). "An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation". In: *Management Science* 44.5, pp. 714–729.

Mirzaei, Omid and Mohammad-R Akbarzadeh-T (2013). "A novel learning algorithm based on a multi-agent structure for solving multi-mode resource-constrained project scheduling problem". In: *Journal of Convergence* 4.1, pp. 47–52.

Mori, M. (1997). "A genetic algorithm for multi-mode resource constrained project scheduling problem". In: *European Journal Of Operational Research* 100.1, pp. 134–141. ISSN: 03772217.

Moscato, Pablo et al. (1989). "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms".

Muritiba, Albert Einstein Fernandes, Carlos Diego Rodrigues, and Franc\'\iio Araùjo da Costa (2018). "A Path-Relinking algorithm for the multi-mode resource-constrained project scheduling problem". In: *Computers & Operations Research* 92, pp. 145–154.

Neumann, Klaus, Christoph Schwindt, and Júrgen Zimmermann (2012). *Project scheduling with time windows and scarce resources: temporal and resource-constrained project scheduling with regular and nonregular objective functions*. Springer Science & Business Media.

Neumann, Klaus and Ulrich Steinhardt (1979). *GERT Networks and the Time-Oriented Evaluation of Projects*. 1st ed. Lecture Notes in Economics and Mathematical Systems 172. Springer-Verlag Berlin Heidelberg.

Ning, Minjing et al. (2018). "Metaheuristic algorithms for proactive and reactive project scheduling to minimize contractor's cash flow gap under random activity duration". In: *IEEE Access*. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2828037.

Niu, Ben et al. (2013). "Multi-objective bacterial foraging optimization". In: *Neurocomputing* 116, pp. 336–345.

Nonobe, Koji and Toshihide Ibaraki (2002). "Formulation and tabu search algorithm for the resource constrained project scheduling problem". In: *Operation Research/Computer Science Interfaces Series*. Vol. 15. Operations Research/Computer Science Interfaces Series. Springer US, pp. 557–588. ISBN: 978-1-4613-5588-5.

Özdamar, Linet (1999). "A genetic algorithm approach to a general category project scheduling problem". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 29.1, pp. 44–59. ISSN: 1094-6977.

Özdamar, Linet and Gunduz Ulusoy (1995). "A survey on the resource-constrained project scheduling problem". In: *IIE transactions* 27.5, pp. 574–586.

Panda, Rutuparna and Manoj Kumar Naik (2012). "A crossover bacterial foraging optimization algorithm". In: *Applied Computational Intelligence and Soft Computing* 2012, p. 23.

Passino, Kevin M (2002). "Biomimicry of bacterial foraging for distributed optimization and control". In: *IEEE control systems* 22.3, pp. 52–67.

– (2010). "Bacterial foraging optimization". In: *International Journal of Swarm Intelligence Research (IJSIR)* 1.1, pp. 1–16.

Patterson, James H (1984). "A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem". In: *Management Science* 30.7, pp. 854–867.

Petrović, Radivoj (1968). "Optimization of resource allocation in project planning". In: *Operations Research* 16.3, pp. 559–568.

Pritsker, A., Lawrence Waiters, and Philip Wolfe (1969). "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach". In: *Management Science* 16.1, pp. 93–108.

Ranjbar, Mohammad, Bert De Reyck, and Fereydoon Kianfar (2009). "A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling". In: *European Journal of Operational Research* 193.1, pp. 35–48. ISSN: 03772217.

Ratajczak-Ropel, Ewa and Aleksander Skakovski (2018). "Multi-mode Resource-Constrained Project Scheduling". In: *Population-Based Approaches to the Resource-Constrained and Discrete-Continuous Scheduling*. Springer, pp. 69–97.

Reyck, Bert De and Willy Herroelen (1999). "The multi-mode resource-constrained project scheduling problem with generalized precedence relations". In: *European Journal of Operational Research* 119.2, pp. 538 –556. ISSN: 0377-2217.

Saber, Ahmed Y and Ganesh K Venayagamoorthy (2008). "Economic load dispatch using bacterial foraging technique with particle swarm optimization biased evolution". In: *2008 IEEE Swarm Intelligence Symposium*. IEEE, pp. 1–8.

Schott, Jason Ramon (1995). "Fault tolerant design using single and multicriteria genetic algorithm optimization". PhD thesis. Massachusetts Institute of Technology.

Sebt, M H, M R Afshar, and Y Alipouri (2016). “Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem”. In: *Engineering Optimization*, pp. 1–18.

Sebt, M. H., Y. Alipouri, and Y. Alipouri (2013). “Solving resource-constrained project scheduling problem with evolutionary programming”. In: *Journal of the Operational Research Society* 64.9, pp. 1327–1335. ISSN: 01605682.

Sebt, Mohammad Hassan, Mohammad Reza Afshar, and Yagub Alipouri (2015). “An efficient genetic algorithm for solving the multi-mode resource-constrained project scheduling problem based on random key representation”. In: *International Journal of Supply and Operations Management* 2.3, pp. 905–924.

Shen, Hong and Xiaoping Li (2013). “Cooperative discrete particle swarms for multi-mode resource-constrained projects”. In: *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2013*, pp. 31–36. ISBN: 9781467360852.

Slowinski, Roman (1989). “Multiobjective project scheduling under multiple-category resource constraints”. In: *Advances in project scheduling* 80, p. 09.

Słowiński, Roman, Boleslaw Soniewicki, and Jan Weglarz (1994). “DSS for multiobjective project scheduling”. In: *European Journal of Operational Research* 79.2, pp. 220–229.

Soliman, Omar S and Elshimaa A R Elgendi (2014). “A Hybrid Estimation of Distribution Algorithm with Random Walk local Search for Multi-mode Resource-Constrained Project Scheduling problems”. In: *arXiv preprint arXiv:1402.5645*.

Song, Wen et al. (2018). “An agent-based simulation system for multi-project scheduling under uncertainty”. In: *SIMULATION MODELLING PRACTICE AND THEORY* 86, 187–203. ISSN: 1569-190X. DOI: {10.1016/j.simpat.2018.05.009}.

Sprecher, A. (1994). *Resource-Constrained Project Scheduling: Exact Methods for the Multi-Mode Case*. Lecture Notes in Economics and Mathematics N° 409. Springer.

Sprecher, Arno and Andreas Drexl (1998). “Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm”. In: *European Journal of Operational Research* 107.2, pp. 431–450. ISSN: 0377-2217.

Suwa, Haruhiko and Daisuke Morita (2016). “Reactive project scheduling method to enhance project progress under uncertainty”. In: *Journal of Advanced Mechanical Design, Systems and Manufacturing*. ISSN: 18813054.

Szczesny, Kamil and Markus König (2015). “Reactive scheduling based on actual logistics data by applying simulation-based optimization”. In: *Visualization in Engineering* 3.1, pp. 1–16.

Talbot, F. Brian (1982). “Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case”. In: *Manage. Sci.* 28.10, pp. 1197–1210.

Tantisuvanichkul, Vacharee (2014). “Optimising net present value using priority rule-based scheduling”. PhD thesis. University of Manchester.

Tiwari, Vikram, James H Patterson, and Vincent A Mabert (2009). “Scheduling projects with heterogeneous resources to meet time and quality objectives”. In: *European Journal of Operational Research* 193.3, pp. 780–790.

Tormos, Pilar and Antonio Lova (2001). "A Competitive Heuristic Solution Technique for Resource-Constrained Project Scheduling". In: *Annals of Operations Research* 102.1-4, pp. 65–81. ISSN: 02545330.

Tranfield, David, David Denyer, and Palminder Smart (2003). "Towards a methodology for developing evidence-informed management knowledge by means of systematic review". In: *British journal of management* 14.3, pp. 207–222.

Tseng, Lin-Yu and Shih-Chieh Chen (2009). "Two-Phase Genetic Local Search Algorithm for the Multimode Resource-Constrained Project Scheduling Problem". In: *IEEE Transactions on Evolutionary Computation* 13.4, pp. 848–857. ISSN: 1089-778X.

Ulusoy, Gündüz, Funda Sivrikaya-Şerifoğlu, and Şule Şahin (2001). "Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows". In: *Annals of Operations Research* 102.1-4, pp. 237–261.

Vaisakh, K, P Praveena, and S Rama Mohana Rao (2010). "PSO-DV and bacterial foraging optimization based dynamic economic dispatch with non-smooth cost functions". English. In: *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*. Vol. 1. IEEE, pp. 131–139.

Valdes, Ramon Alvarez and Jose Tamarit Goerlich (1993). "The project scheduling polyhedron: Dimension, facets and lifting theorems". In: *European Journal of Operational Research* 67.2, pp. 204 –220. ISSN: 0377-2217. DOI: http://dx.doi.org/10.1016/0377-2217(93)90062-R. URL: http://www.sciencedirect.com/science/article/pii/037722179390062R.

Valls, V. et al. (2007). "Project Scheduling Optimization in Service Centre Management". In: *Review of Business and Economic Literature*.

Valls, Vicente et al. (1999). "Project scheduling with stochastic activity interruptions". In: *Project Scheduling*. Springer, pp. 333–353.

Van Peteghem, Vincent and Mario Vanhoucke (2009). "An artificial immune system for the multi-mode resource-constrained project scheduling problem". In: *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 85–96.

– (2010). "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem". In: *European Journal of Operational Research* 201.2, pp. 409–418.

– (2011). "Using resource scarceness characteristics to solve the multi-mode resource-constrained project scheduling problem". In: *Journal of Heuristics* 17.6, pp. 705–728.

– (2014). "An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances". In: *European Journal of Operational Research* 235.1, pp. 62–72.

Vanhoucke, Mario (2012). *Project management with dynamic scheduling*. Springer.

Vanhoucke, Mario, Erik Demeulemeester, and Willy Herroelen (2001). "On maximizing the net present value of a project under renewable resource constraints". In: *Management Science* 47.8, pp. 1113–1121.

Vartouni, Ali Moradi and Leyli Mohammad Khanli (2014). "A hybrid genetic algorithm and fuzzy set applied to multi-mode resource-constrained project scheduling problem". In: *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology* 26.3, pp. 1103–1112. ISSN: 1064-1246.

Vieira, Guilherme E, Jeffrey W Herrmann, and Edward Lin (2003). "Rescheduling manufacturing systems: a framework of strategies, policies, and methods". In: *Journal of Scheduling* 6.1, pp. 39–62.

Vonder, Stijn Van de, Erik Demeulemeester, and Willy Herroelen (2007). "A classification of predictive-reactive project scheduling procedures". In: *Journal of scheduling* 10.3, pp. 195–207.

– (2008). "Proactive heuristic procedures for robust project scheduling: An experimental analysis". In: *European Journal of Operational Research* 189.3, pp. 723–733.

Vonder, Stijn Van de et al. (2006). "Proactive-reactive project scheduling trade-offs and procedures". In: *Perspectives in modern project scheduling*. Springer, pp. 25–51.

Vonder, Stijn Van de et al. (2007a). "Heuristic procedures for reactive project scheduling". In: *Computers & Industrial Engineering* 52.1, pp. 11–28.

– (2007b). "Heuristic procedures for reactive project scheduling". In: *Computers & Industrial Engineering* 52.1, pp. 11–28.

Waligóra, Grzegorz (2014). "Discrete-continuous project scheduling with discounted cash inflows and various payment models—a review of recent results". In: *Annals of Operations Research* 213.1, pp. 319–340.

Wan, Miao et al. (2012). "Data clustering using bacterial foraging optimization". In: *Journal of Intelligent Information Systems* 38.2, pp. 321–341.

Wang, Juite (2005). "Constraint-based schedule repair for product development projects with time-limited constraints". In: *International Journal of Production Economics*. ISSN: 09255273.

Wang, L., D. Zhan, and L. Nie (2014). "Combinatorial exchange based decentralized project reactive scheduling method". Chinese. In: *Gaojishu Tongxin/Chinese High Technology Letters* 24.3, pp. 316–322. ISSN: 10020470.

Wang, Lei et al. (2012). "Robust control for decentralized multi-project reactive scheduling based on combinatorial exchange". In: *ICIC Express Letters, Part B: Applications*. ISSN: 21852766.

Wang, Ling and Chen Fang (2011). "An effective shuffled frog-leaping algorithm for multi-mode resource-constrained project scheduling problem". In: *Information Sciences* 181.20, pp. 4804–4822.

– (2012). "An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem". In: *Computers & Operations Research* 39.2, pp. 449–460.

Wang, Lixia, Jing Liu, and Mingxing Zhou (2014). "An organizational cooperative coevolutionary algorithm for multimode resource-constrained project scheduling problems". In: *Asia-Pacific Conference on Simulated Evolution and Learning*, pp. 680–690.

Wang, Weixin et al. (2019). "Proactive and Reactive Multi-Project Scheduling in Uncertain Environment". In: *IEEE Access*. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2926337.

Wauters, T. et al. (2009). "A Learning Metaheuristic for the Multi Mode Resource Constrained Project Scheduling Problem". In: *Learning and Intelligent OptimizatioN (LION 3). Trento, Italy*.

Weglarz, Jan et al. (2011). "Project scheduling with finite or infinite number of activity processing modes: A survey". In: *European Journal of Operational Research* 208.3, pp. 177 –205. ISSN: 0377-2217. URL: http://www.sciencedirect.com/science/article/pii/S037722171000264X.

Wiest, Jerome D (1964). "Some properties of schedules for large projects with limited resources". In: *Operations Research* 12.3, pp. 395–418.

– (1967). "A heuristic model for scheduling large projects with limited resources". In: *Management Science* 13.6, B–359.

Xu, Ye et al. (2014). "An effective hybrid immune algorithm for solving the distributed permutation flow-shop scheduling problem". In: *Engineering Optimization* 46.9, pp. 1269–1283.

Yang, Bibo and Joseph Geunes (2008). "Predictive–reactive scheduling on a single resource with uncertain future jobs". In: *European Journal of Operational Research* 189.3, pp. 1267–1283.

Yuan, Qiaolin, Yiannis E Polychronakis, et al. (2012). "The development of a robust resource constrained project scheduling framework". In: *International Journal of Project Organisation and Management* 4.4, pp. 339–367.

Zaman, Forhad et al. (2020). "Hybrid evolutionary algorithm for large-scale project scheduling problems". In: *COMPUTERS & INDUSTRIAL ENGINEERING* 146. ISSN: 0360-8352. DOI: {10.1016/j.cie.2020.106567}.

Zang, Tianlei, Zhengyou He, and Deyi Ye (2010). "Bacterial foraging optimization algorithm with particle swarm optimization strategy for distribution network reconfiguration". In: *International Conference in Swarm Intelligence*. Springer, pp. 365–372.

Zapata, Juan Camilo, Bri Mathias Hodge, and Gintaras V. Reklaitis (2008). "The multimode resource constrained multiproject scheduling problem: Alternative formulations". In: *AIChE Journal* 54.8, pp. 2101–2119. ISSN: 1547-5905. DOI: 10.1002/aic.11522. URL: http://dx.doi.org/10.1002/aic.11522.

Zhang, Hong (2011). "Ant colony optimization for multimode resource-constrained project scheduling". In: *Journal of Management in Engineering* 28.2, pp. 150–159.

Zhang, Hong, C. M. Tam, and Heng Li (2006). "Multimode project scheduling based on particle swarm optimization". In: *Computer-Aided Civil and Infrastructure Engineering* 21.2, pp. 93–103. ISSN: 10939687.

Zhang, Lieping, Yingxiong Luo, and Yu Zhang (2015). "Hybrid Particle Swarm and Differential Evolution Algorithm for Solving Multimode Resource-Constrained Project Scheduling Problem". In: *Journal of Control Science and Engineering* 2015. ISSN: 16875257.

Zhang, Zhe and Jiuping Xu (2016). “Bi-level multiple mode resource-constrained project scheduling problems under hybrid uncertainty”. In: *Journal of Industrial and Management Optimization* 12.2, pp. 565–593.

Zheng, Weibo et al. (2018). “Proactive and reactive resource-constrained max-NPV project scheduling with random activity duration”. In: *Journal of the Operational Research Society*. ISSN: 14769360. DOI: 10.1057/s41274-017-0198-3.

Zhengwei, G, HL Pang, and DW Wang (2011). “Adaptive bacterial foraging optimization and its application for bus scheduling”. In: *Journal of System Simulation* 23.6, pp. 1151–1160.

Zhu, Guidong, Jonathan F Bard, and Gang Yu (2005). “Disruption management for resource-constrained project scheduling”. In: *Journal of the Operational Research Society* 56.4, pp. 365–381.

– (2006). “A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem”. In: *INFORMS Journal on Computing* 18.3, pp. 377–390.

– (2007). “A two-stage stochastic programming approach for project planning with uncertain activity durations”. In: *Journal of Scheduling* 10.3, pp. 167–180. URL: http://link.springer.com/article/10.1007/s10951-007-0008-x.

Zitzler, Eckart and Lothar Thiele (1998). “Multiobjective optimization using evolutionary algorithms—a comparative case study”. In: *International conference on parallel problem solving from nature*. Springer, pp. 292–301.