Various extensions in resource-constrained project scheduling with alternative subgraphs

Tom Servranckx[a] and José Coelho[a,b,c] and Mario Vanhoucke[a,d,e]

[a]Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Ghent (Belgium);[b]INESC Technology and Science, Rua Dr. Roberto Frias, 4200-465 Porto (Portugal);[c]Universidade Aberta Rua Almirante Barroso 38, 1500-235 Lisbon (Portugal);[d]Operations and Technology Management Centre, Vlerick Business School, Reep 1, 9000 Ghent (Belgium);[e]UCL School of Management, University College London, 1 Canada Square, London E14 5AA (UK)

ARTICLE HISTORY

Compiled June 2, 2021

ABSTRACT

In this research, we present several extensions for the resource-constrained project scheduling problem with alternative subgraphs (RCPSP-AS). First of all, we investigate more complex variants of the alternative project structure. More precisely, we consider nested alterative subgraphs, linked alternative branches, multiple selection, caused and closed choices, and split choices. Secondly, we introduce non-renewable resources in the RCPSP-AS in order to implicitly avoid certain combinations of alternatives given a limited availability of this resource over the complete project horizon. We formulate both the basic RCPSP-AS and its extensions as an ILP model and solve it using Gurobi. The computational experiments are conducted on a large set of artificial project instances as well as three case studies. The results show the impact of the different extensions on the project makespan and the computational complexity. We observe that combinations of the proposed extensions might imply complex alternative project structures, resulting in an increasing computational complexity or even infeasible solutions. The analysis of the three case studies shows

tom.servranckx@ugent.be
jcoelho@uab.pt
mario.vanhoucke@ugent.be

that it is hard to find feasible solutions with a small time limit or optimal solutions with a larger time limit for projects with a realistic size in terms of the number of activities or alternatives.

KEYWORDS

Project scheduling; Resource constraints; Alternative subgraphs; Integer formulation; Case studies

1. Introduction

Scheduling a pre-defined set of activities in a project given a limited availability of renewable resources and precedence constraints such that the project makespan is minimized, is a well-studied problem called the resource-constrained project scheduling problem (RCPSP). For this scheduling problem, several optimization procedures have been developed (Herroelen, De Reyck, and Demeulemeester 1998). In order to improve the practical relevance of the RCPSP with its restrictive assumptions, many extensions of the problem have been presented and investigated over the years (Hartmann and Briskorn 2010). In the current research study, we consider an extension of the RCPSP, the so-called RCPSP with alternative subgraphs (RCPSP-AS), introduced by Servranckx and Vanhoucke (2019a). In the basic RCPSP, the project structure is pre-defined and fixed and hence all activities in the project structure should be included in the final project schedule. In the RCPSP-AS, however, this assumption is no longer supported as not all activities in the project network should be scheduled. More precisely, alternative ways to execute work packages are considered in the RCPSP-AS, each containing a subset of interconnected activities in the project. Similar to the RCPSP, the project consists of a pre-defined set of activities connected in a project network. In contrast to the RCPSP, however, only a subset of the activities should be scheduled in the RCPSP-AS. As a result, the RCPSP-AS consists of two subproblems: a selection subproblem and a scheduling subproblem. First, one alternative should be selected for each WP and, subsequently, the scheduling of the selected activities boils down to the basic RCPSP. Where the basic problem was introduced by Servranckx and Vanhoucke (2019a), we extend this problem to solve more complex variants of the RCPSP-AS and apply the problem in more realistic settings. We present extensions such as the ability to select multiple alternatives, exclude or induce alternatives and group alternatives for each WP.

In this research, we present a problem formulation for the basic RCPSP-AS and its extensions and solve this problem using Gurobi. This solution approach has already been used in literature for similar scheduling problems (Kellenbrink and Helber 2015; Tao and Dong 2017, 2018) and showed promising results. In order to quantify the impact of these extensions on the computational effort and solution quality of the schedule, we design different computational experiments using a dataset of artificial project instances. Also, three case studies consisting of a combination of the proposed extensions are analysed. This analysis is used to test our solution approach for project instances with a more realistic size.

The outline of the paper can be summarised along the following lines. In Section 2, we link the proposed research efforts to the existing literature in project management and production research. Section 3 introduces the problem formulation of the basic RCPSP-AS and the different extensions proposed in this research. We also present an illustrative example to explain the terminology used in this study and apply the different concepts to illustrative examples from the production environment. We present the computational experiments and discuss the outcomes as well as introduce some case studies in section 4. In section 5, we draw general conclusions and focus on future research avenues.

2. Related work

In this section, we provide a summary of the existing literature on project and production scheduling with a certain degree of flexibility in the project or production process. More precisely, we focus on (1) the resource-constrained project scheduling problem with alternative network structures, (2) the resource-constrained project scheduling problem with resource flexibility (given its practical relevance for production processes) and (3) the flexible (machine) job shop scheduling problem. An overview of the related work is provided in Table 1.

RCPSP with alternative project structures Capek, Šucha, and Hanzálek (2012) discuss an extension of the RCPSP for production scheduling with *alternative process plans*. Alternative process plans are characterised by different resource requirements and processing times, however, even the number and type of activities as well as the precedence relations between the activities might differ in alternative process plans. Kellenbrink and Helber (2015, 2016) consider *flexible project structures* in the RCPSP (RCPSP-PS) given that the set of activities that should be scheduled is not known in advance. The authors define causing activities and caused choices in order to model dependencies between the inclusion/exclusion of certain activities. Tao and Dong (2017) present another extension of the RCPSP, called the RCPSP with *alternative activity chains* (RCPSP-AC). In this problem, the authors consider interchangeable process patterns or methods that are modeled by one or more activities connected through precedence relations, i.e. activity chains. The authors propose an integer linear program as well as a simulated annealing algorithm to solve problem instances for the RCPSP-AC. Tao et al. (2018) further investigate this scheduling problem in case that the selection of one alternative process method might trigger the selection of another process method and thus there exists a hierarchical structure of choices. In the multi-mode RCPSP (MRCPSP) with alternative project structures, activities can be implemented in one of several execution modes (with distinct resource requirements and activity durations), however, only a part of the activities should be implemented (Tao and Dong 2018). As a result, this problem consists of two subproblems: a mode selection and activity selection subproblem. Cajzek and Klanšek (2019) investigate cost optimization of project schedules under resource constraints and *alternative production processes* (APPs). Where project scheduling determines the timing of each activity considering the technological aspects of the activities and the resource availabilities, the production processes mainly focus on the technological features of the resources. The authors present a mixed-integer non-linear programming formulation for the cost optimization problem with amongst others generalized precedence relations and lag and lead times. Kosztyán (2015), Kosztyán and Szalkai (2018) and Kosztyán and Szalkai (2020) also investigate projects with uncertain rather than fixed logical project structures, called *flexible projects*. In flexible projects, the authors assume flexible dependencies, i.e. a probability that a precedence relation exists between two activities, and uncertain task completions, i.e. a probability that an activity with its corresponding precedence relations should be included in the project structure. Finally, Servranckx and Vanhoucke (2019a) present an extension of the basic RCPSP in which there exist alternative project structures, called *alternative subgraphs* (i.e. RCPSP-AS), and they identify different types of relations between those alternative subgraphs. Subsequently, Servranckx and Vanhoucke (2019b) investigate how the inherent flexibility in the RCPSP-AS can be used to construct a set of backup schedules (either similar or dissimilar) in order to deal with the uncertainty during project execution. Due to the complexity of the selection subproblem in the RCPSP-AS, Servranckx, Vanhoucke, and Vanhouwaert (2020) present a two-step procedure to reduce the number of alternatives based on the schedule diversity and choice frequency thresholds.

Resource flexibility From a production management perspective, many research efforts focus on scheduling problems with resource allocation flexibility, referred to as the *flexible resource-constrained project scheduling problem* (FRCPSP) (Kolisch et al. 2003). In this extension of the RCPSP, the resource demand profile is no longer pre-defined, but a model-endogenous decision. An important aspect in FRCPSP is resource flexibility, which implies that the resource requirement of each activity is known, but the duration and resource demand is variable. During the activity execution, a positive (variable) resource demand exists in each time period such that the overall resource demand is higher than or equal to the total resource requirement. Fündeling and Trautmann (2010) study a variant of the RCPSP with discrete resource requirements and aim to determine a feasible resource-usage profile for each activity. Furthermore, Ranjbar and Kianfar (2010) investigate the RCPSP with flexible work profiles and exact allocation of resource requirements. They use a genetic algorithm to generate all feasible work profiles and, subsequently, schedule the project such that the project makespan is minimized and the precedence and resource relations are satisfied. Tritschler, Naber, and Kolisch (2017) develop a hybrid metaheuristic for the FRCPSP that consists of a genetic algorithm extended with delayed scheduling and non-greedy resource allocation in the schedule generation scheme. Kis (2005) presents a strong model formulation for a scheduling problem in which the resource usage of an activity varies over time proportionally to its varying intensity. Naber and Kolisch (2014) investigate four mixed-integer programming formulations for the FRCPSP considering continuous resources, subdivided in three categories (principal, dependent and independent resources), and overallocation of resources. The authors show that the variable-intensity-based model dominates the other three models for both solution quality and computational times. The work of Kis (2005) is extended by Bianco, Caramia, and Giordani (2016) to study the *resource levelling problem* (RLP) with variable execution intensities and flexible durations of the activities. In the RLP, activities should be scheduled such that the resource utilization will be as stable as possible over a medium planning horizon (Demeulemeester and Herroelen 2002). This problem is relevant from both a practical and theoretical perspective in production settings since it avoids high penalty costs related to resource over- and underusage.

Another type of resource flexibility implies that resources might be multi-skilled (e.g. multi-purpose machines) and activities require one (Li and Womer 2009) or multiple (Correia, Womer, and da Gama 2012) resource units with certain skills. This extension of the RCPSP is referred to as the *project scheduling problem with flexible resources* (PSPFR). Li and Womer (2009) solve the PSPFR with a cost minimization objective considering a pre-defined project deadline using Benders decomposition. Correia, Womer, and da Gama (2012) propose a mixed-integer linear programming (MILP) formulation with the sole objective of project makespan minimization. Almeida, Correia, and Saldanha-da Gama (2016) extend the well-known parallel schedule generation scheme (PSGS) in order to tackle large problem instances of the PSPFR. Building on this research, Almeida and Saldanha-da Gama (2018) propose a biased random-key genetic algorithm for the PSPFR based on the combined application of a serial scheduling generation scheme (SGSG) and the PSGS extension proposed by Almeida, Correia, and Saldanha-da Gama (2016). Furthermore, Correia and Saldanha-da Gama (2014) study the impact of fixed and variable (related to the project makespan) costs associated with the resource usage based on an empirical study and extensive computational experiments. Finally, a general modeling framework is introduced by Correia and Saldanha-da Gama (2015) that focuses on MILP formulations and includes many problem features of project staffing and scheduling problems. The authors discuss several model enhancements, preprocessing procedures and modeling issues.

Other researchers have investigated the activity-resource flexibility of temporary resources (i.e. renting) in order to reduce the project cost. The *resource renting problem* (RRP) (Nübel 2001) is a time-constrained project scheduling problem that aims to minimize the total project cost consisting of time-independent costs, time-dependent costs (the renting costs, e.g. wages of resources) and costs of idle resources. Ballestín (2007) presents a genetic algorithm for the RRP with minimum and maximum time-lags, while Ballestín (2008) compares different metaheuristics for this scheduling problem with activity-resource flexibility. Vandenheede, Vanhoucke, and Maenhout (2016) extend the RRP with additional costs from the total adjustment cost problem (Kreter, Rieck, and Zimmermann 2014) and present a scatter search algorithm with specific building blocks that are developed for the RRP/extended. In order to improve the practical relevance of the RRP, Kerkhove, Vanhoucke, and Maenhout (2017) combine the basic problem with the ability to schedule some activities during overtime.

Job shop scheduling Research on operation optimization with flexibility in the production system has focused on (1) single machine systems, (2) flow shops (e.g. production lines) or (3) job shops that may contain different machines. Extending the job shop scheduling problem (JSP), Kis (2003) introduces alternative routings of jobs, resulting in the JSP with alternative processes (AJSP). The *flexible job shop scheduling problem* (FJSP) is another generalization of the classical JSP, in which operations must be processed on a machine selected from a set of available machines. Li et al. (2020) propose an optimization method for flexible (machining) job shop scheduling with new job arrivals and machine breakdowns. Due to the complexity of this problem, several metaheuristic approaches have been developed such as tabu search (Mastrolilli and Gambardella 2000), modified simulated annealing method (Najid, Dauzere-Peres, and Zaidat 2002), improved genetic-simulated annealing algorithm (Dai et al. 2013) and improved particle swarm optimization (Tang et al. 2016). Furthermore, the *multi-objective flexible job shop scheduling problem* (MOFJSP) has been solved with different types of objective functions (e.g. makespan, machine workload, energy consumption, tardiness, etc.) using different approaches such as hierarchy-based methods (Jiang and Ma 2016), weighted-sum-based methods (Zhang et al. 2009) and Pareto-based methods (Lei, Zheng, and Guo 2017). Gong et al. (2018) propose a *double flexible job shop scheduling problem* (DFJSP) with worker and machine flexibility, which is more practically relevant for production settings than the FJSP. Finally, Vela et al. (2020) investigate a variant of JSP with fuzzy durations, called *fuzzy job shop scheduling problems*, by extending the problem with fuzzy due dates and the objective of due date satisfaction under uncertainty. Several metaheuristics have been proposed to solve the fuzzy job shop problem and its extensions, such as a non-sorted genetic algorithm (Vela et al. 2020) and hybrid approaches (Abdullah and Abdolrazzagh-Nezhad 2014; Palacios et al. 2016).

In this study, we mainly consider the scheduling problem with alternatives from a project perspective, but the proposed approach can also be used for production scheduling. In the existing literature, it is shown that project-oriented concepts and scheduling approaches are suitable for scheduling production systems (Carvalho, Oliveira, and Scavarda 2016, 2015; Alfieri, Tolio, and Urgo 2011; Ghiyasinasab et al. 2020). However, different levels of detail might be required for aggregate and tactical production scheduling with activities corresponding to production phases and distinct production operations, respectively (Alfieri, Tolio, and Urgo 2012).

[Table 1 about here.]

3. Problem formulation

In section 3.1, we introduce a problem formulation for the basic RCPSP-AS and illustrate the terminology by means of an illustrative example. Subsequently, we introduce each of the extensions of the RCPSP-AS in section 3.2. We present the required definitions, extend the problem formulation and illustrate the impact of the extensions using an illustrative example.

3.1. *Basic RCPSP-AS*

The basic RCPSP can be stated as follows. A set of activities $i \in N$, numbered from a dummy start node 0 to a dummy end node $n+1$, is to be scheduled without preemption on a set R^p of renewable resource types. Each renewable resource $v \in R^p$ has a constant availability a^p_v per period and each non-dummy activity $i \in N$ has a deterministic duration d_i and requires $r^p_{i,v}$ units of resource type v. We assume that $r^p_{i,v} \leq a^p_v \quad (\forall i \in N; \forall v \in R^p)$. The start and end dummy activities representing the start and completion of the project have a duration d_i and a renewable resource requirement $r^p_{i,v} (\forall v \in R^p)$ equal to zero. A project network is represented by a topological ordered activity-on-the-node (AoN) format where A is the set of pairs of activities (i, j) between which a finish-start precedence relationship with a zero time-lag exists. We assume graph $G(N, A)$ to be acyclic. A schedule S is defined by a vector of activity start times and is said to be feasible if all precedence and renewable resource constraints are satisfied. The objective of the RCPSP is to find a feasible schedule within the lowest possible project makespan.

In the RCPSP-AS, the set of non-dummy activities N is subdivided in two disjoint groups: the set of fixed activities N^f and the set of alternative activities N^a. Fixed activities should always be selected in the RCPSP-AS, while alternative activities are optional. Each alternative activity $i \in N^a$ belongs to exactly one alternative subgraph $l \in L$ with L the number of different subgraphs in the project network. A subgraph l is defined by a dummy start activity p_l and a dummy end activity t_l. The dummy start activity p_l is called the principal activity and has no predecessors in the subgraph, while the dummy end activity t_l is called the terminating activity and has no successors in the subgraph. Consequently, the set of subgraphs in the project network can be defined by their principal and terminating activities as $L = \{(p_{l_1}, t_{l_1}), ..., (p_{|L|}, t_{|L|})\}$. Each subgraph contains a set of k alternative branches which are subparts of the alternative subgraph. Each alternative branch starts with a branching activity b_k. The set of alternative branches can be defined by their branching activity as $K_{p_l} = \{b_{k_1}, ..., b_{|K_l|}\}$.

Consequently, the project network used to solve the RCPSP-AS does not differ much from the network used to solve the RCPSP, and also includes the precedence relations and renewable resource constraints. However, the network now consists of both fixed activities (set N^f) and alternative activities (set N^a). Each alternative activity $i \in N^a$ belongs to a specific alternative branch k, and the set of all alternative activities in alternative branch k is defined by N_{b_k} (which always includes its branching activity b_k) with $N_{b_k} \subset N$ and $N = N^f \cup N^a$. The construction of a project schedule for the RCPSP-AS involves the selection of exactly one alternative branch for each alternative subgraph. In table 2, we summarise the notation of the basic RCPSP-AS as introduced before.

In the remainder of this section, we present a time-indexed formulation for the RCPSP-AS, based on the formulation by Pritsker, Watters, and Wolfe (1969), using two binary decision variables $x_{i,t}$ and y_i. The variable $x_{i,t}$ is equal to 1 if activity i is started at time instance t and 0 otherwise, while the variable y_i is equal to 1 if the activity i is selected and 0 otherwise.

[Table 2 about here.]

$$\text{Min} \sum_{t=es_{n+1}}^{ls_{n+1}} t x_{n+1,t} \tag{1}$$

subject to

$$\sum_{t=es_i}^{ls_i} (t + l_{i,j}) x_{i,t} - M(1 - y_i) \leq \sum_{t=es_j}^{ls_j} t x_{j,t} + M(1 - y_j) \qquad \forall (i,j) \in A \tag{2}$$

$$\sum_{t=es_i}^{ls_i} x_{i,t} = y_i \qquad \forall i \in N \tag{3}$$

$$\sum_{k=1}^{|K_{p_l}|} y_{b_k} = 1 \qquad \forall l \in L \tag{4}$$

$$y_j \geq y_i \qquad \forall (i,j) \in A; \forall i,j \in N_{b_k}; \forall k \in K_{p_l}; \forall l \in L \tag{5}$$

$$y_i = 1 \qquad \forall i \in N^f \tag{6}$$

$$\sum_{i=1}^{n} r_{i,v}^{p} \sum_{s=max(t-d_i,es_i)}^{min(t-1,ls_i)} x_{i,s} \leq a_v^p \qquad \forall v \in R^p; t = \{1, ..., T\} \quad (7)$$

$$x_{i,t}; y_i \in 0, 1 \qquad \forall i \in N; t = \{1, ..., T\} \quad (8)$$

In the above formulation, es_i and ls_i are used to indicate the earliest and latest start for activity i given the upper bound T using the adjusted forward and backward critical path calculations discussed in Servranckx and Vanhoucke (2019a). The objective of the RCPSP-AS is to minimize the total project makespan as shown in eq. (1). Eq. (2) implies that the precedence relations of the selected activities should be considered. The parameter $l_{i,j}$ is used to transform non-default precedence relations into default finish-start precedence relations with a minimal time-lag equal to zero using a set of transformation rules (Bartusch, Möhring, and Radermacher 1988). The value of $l_{i,j}$ depends on the type of precedence relation as shown in Table 3 (Demeulemeester and Herroelen 2002). Eq. (3) guarantees that each (selected) activity can only start once at a certain time $t \in [es_i; ls_i]$. Eq. (4) ensures that only one alternative branch (i.e. $y_{b_k} = 1$) can be selected for each alternative subgraph. In case that a precedence relation exists between two activities $(i, j) \in A$ that belong to the same alternative branch k, the selection of activity i implies the selection of activity j as shown in eq. (5). The fixed activities $i \in N^f$ should always be selected, including the dummy start and end activity should also be selected (eq. (6)). The renewable resource constraints of the selected activities are satisfied in eq. (7). Eq. (8) forces the decision variables to be binary values.

[Table 3 about here.]

In certain projects (e.g. construction projects), the main objective might not be to minimize the project makespan (Eq. (1)), but rather to minimize the project cost. As a result, the above model might lead to the selection of alternatives that result in a short project duration, however, with a high project cost. As this might not be preferred for certain projects, we show how the proposed model could be extended in order to incorporate the project cost objective. The total project cost consists of the direct cost C^d, the indirect cost C^i, the tardiness cost C^t and the fixed cost C^f. The direct cost is related to the work content of each activity, while the indirect cost is mainly related to the project makespan. The tardiness cost should be paid as a penalty if the contract deadline is exceeded. The unit direct, indirect and tardiness cost is represented by c^d, c^i and c^t. In order to incorporate the minimal cost objective in the RCPSP-AS, we change the objective function Eq. (1) with Eq. (1') and add the constraints Eqs. (1a)-(1e).

$$\text{Min } C^{tot} \qquad (1')$$

subject to

$$C^{tot} = C^d + C^i + C^t + C^f \qquad (1a)$$

$$C_{max} = \sum_{t=es_{n+1}}^{ls_{n+1}} t x_{n+1,t} \qquad (1b)$$

$$C^d = \sum_{i=1}^{n} \sum_{v=1}^{|R^p|} c^d \times r_{i,v} \times d_i \times x_{i,t} \tag{1c}$$

$$C^i = c^i \times C_{max} \tag{1d}$$

$$C^t = c^t \times max\{(C_{max} - h), 0\} \tag{1e}$$

In the remainder of this study, we will consider the RCPSP-AS with the minimization of the project makespan (Eq. (1)) as introduced by Servranckx and Vanhoucke (2019a). However, an important future research direction is to investigate the impact of the aforementioned cost objective on the selection of alternatives.

Illustrative example In order to illustrate the basic concepts of the RCPSP-AS formulation, we present an illustrative example in figure 1(A). The project network consists of $|N| = 44$ activities of which 22 non-dummy activities. The non-dummy activities are indicated in figure 1 by means of the bold activity numbers in the nodes. We consider one renewable resource type p with an availability $a_v^p = 10$ units, while the renewable resource requirement $r_{i,v}^p$ and duration d_i of each activity i are shown in table 4. In contrast to the basic RCPSP, not all of the 44 activities should be scheduled in the RCPSP-AS since the activities can be subdivided between alternative activities N^a and fixed activities N^f (see table 4). In our illustrative example, there exist two alternative subgraphs with four alternative branches for each alternative subgraph. Recall that alternative branches are optional ways of executing a subset of (alternative) activities. The set of alternative branches that are interchangeable are referred to as an alternative subgraph. In this research, the choice between alternative branches is indicated by means of ')'. The alternative subgraphs are identified by their (dummy) principal and terminating activities, represented in figure 1 by $\square$. The set $L = \{(1, 18), (18, 35)\}$ implies that the first (second) alternative subgraph starts with activity $p_1 = 1$ ($p_2 = 18$) and ends with activity $t_1 = 18$ ($t_2 = 35$). The alternative branches are identified by their (dummy) branching activities, i.e. $K_1 = \{2, 6, 10, 14\}$ and $K_{18} = \{19, 23, 27, 31\}$. In this case, the first alternative subgraph consists of four alternative branches that start with the activities 2, 6, 10 and 14 and the second alternative subgraph consists of four alternative branches that start with the activities 19, 23, 27 and 31. For each alternative subgraph, one alternative branch should be selected. When the alternative branch of branching activity $b_1 = 2$ is selected, the activities in the set $N_2 = \{2, 3, 4, 5\}$ should be selected accordingly, while the other activities in the alternative subgraph should not be selected in the schedule. Together with the fixed activities, the selected alternative activities determine the alternative project structure that is used for the construction of the project schedule. In this example, there exist 16 possible project structures since we need to select one out of four alternative branches in two independent alternative subgraphs. In figure 1(A), we show a feasible project schedule that corresponds with a pre-determined alternative project structure. The selected activities (both fixed and alternative) are filled with a dark color in figure 1(A). The activity information needed to construct the schedules is provided in table 4. The decision variable s_i shows the start time of each selected activity i ($s_i > 0$).

[Figure 1 about here.]

[Table 4 about here.]

3.2. *Extensions of RCPSP-AS*

In this section, we introduce different extensions of the basic RCPSP-AS as discussed in section 3.1. In sections 3.2.1-3.2.6, we discuss extensions that allow us to model more complex alternative project structures. In contrast to the previous sections, non-renewable resource constraints are discussed in section 3.2.7. This extension is used to model situations in which certain alternatives cannot be selected simultaneously as this would result in an overconsumption of a non-renewable resource, e.g. a limited project budget. In order to model the different extensions, we need additional notations that are presented in table 5.

[Table 5 about here.]

3.2.1. Nested alternative subgraphs

In the basic formulation of the RCPSP-AS, exactly one alternative branch must be selected in each alternative subgraph. Adding nested alternative subgraphs extends this assumption and implies that at most one alternative must be selected for each alternative subgraph. More specifically, nesting implies that an alternative branch $k' \in K_{p_{l'}}$ in an alternative subgraph $l' \in L$ is selected only when another alternative branch $k \in K_{p_l}$ in alternative subgraph $l \in L$ ($k \neq k'$ and $l \neq l'$) is also selected. Eq. (4) should be replaced by eq. (9) since exactly one alternative branch k' should be selected in alternative subgraph l' when $y_{p_{l'}} = 1$, while no alternative branch k' should be selected when $y_{p_{l'}} = 0$.

$$\sum_{k'=1}^{|K_{p_{l'}}|} y_{b_{k'}} = y_{p_{l'}} \qquad \forall l' \in L \quad (9)$$

In figure 1(B), the second alternative subgraph with principal activity $p_2 = 18$ and terminating activity $t_2 = 35$ is nested in the alternative branch with branching activity $b_1 = 2$. As a result, selecting exactly one alternative branch from the set $K_{18} = \{19, 23, 27, 31\}$ is only needed when the alternative branch related to $b_1 = 2$ is selected from the set $K_1 = \{\mathbf{2}, 6, 10, 14\}$. In case that the parameters and sets for the project example in figure 1(B) needed to change compared to the basic project example in figure 1, they are shown in table 6. One possible project structure with the selected activities colored dark is shown in figure 1(B). In figure 1(B), we show a feasible project schedule for this example project.

[Table 6 about here.]

3.2.2. *Linked alternative branches*

In the basic RCPSP-AS, precedence relations between alternative branches can only be modeled between terminating activities and branching activities. Adding links also allows precedence relations between *any* activity of two alternative branches. Where each alternative activity thus belongs to *exactly one* alternative branch in the basic RCPSP-AS, an alternative activity can belong to *multiple* alternative branches when there exist linked alternative branches. More specifically, linking implies that there exists a precedence relation between two activities i and j with $i \in N_{b_k}$ and $j \in N_{b_{k'}}$ with $k \in K_{p_l}$ and $k' \in K_{p_{l'}}$ $(k \neq k')$. The alternative branches k and k' can belong to the same alternative subgraph $(l = l')$ or different alternative subgraphs $(l \neq l')$. The selection of activity i will result in the selection of activity j and its successors in $N_{b_{k'}}$ and thus activity j will become part of both the alternative branches k' and k. Eq. (5) should be replaced by eq. (10) to ensure that the selection of activity i can trigger the selection of activity j in another alternative branch.

$$y_j \geq y_i \qquad \forall (i,j) \in A; \forall i \in N_{b_k}; \forall j \in N_{b_{k'}}; \forall k \in K_{p_l}; \forall k' \in K_{p_{l'}}; \forall l, l' \in L \quad (10)$$

In figure 1(C), there exists a link between activities 21 and 24 that, respectively, belong to the alternative branches with branching activities $b_5 = 19$ and $b_6 = 23$. Consequently, the selection of the alternative branch with the activities $N_{19} = \{19, 20, 21, 22, 24, 26\}$ will result in the selection of the activities 24 and 26 that also belong to another alternative branch with the activities $N_{23} = \{23, \mathbf{24}, 25, \mathbf{26}\}$. In table 6, the adjusted set for the example project in figure 1(C) is presented. One possible project structure for the resulting project example is shown in figure 1(C), with the selected activities colored dark. Based on these selected activities, we present a feasible project schedule in figure 1(C).

3.2.3. *Multiple selection*

In the basic RCPSP-AS, *exactly one* alternative branch must be selected in each alternative subgraph. In this extension, however, we assume that *multiple* alternative branches should be selected in each alternative subgraph. We model the selection of multiple alternative branches by means of the parameter ρ_l, which indicates that ρ_l alternative branches $k \in K_{p_l}$ (with $1 < \rho_l < |K_{p_l}|$) should be selected in each alternative subgraph $l \in L$. Eq. (4) should be changed by eq. (11) when no nested alternative subgraphs exist.

$$\sum_{k=1}^{|K_{p_l}|} y_{b_k} = \rho_l \qquad \forall l \in L \quad (11)$$

In the presence of nested alternative subgraphs, eq. (9) should be replaced by eq. (12).

$$\sum_{k'=1}^{|K_{p_{l'}}|} y_{b_{k'}} = \rho_{l'} * y_{p_{l'}} \qquad \forall l' \in L \quad (12)$$

In figure 1(D), two alternative branches $k \in K_{p_l}$ should be selected in both alternative subgraphs l, i.e. $\rho_1 = 2$ and $\rho_2 = 2$ (see table 6). The number of alternative branches that should be selected in each alternative subgraph is represented by means of the number of ')' in figure 1(D). One possible project structure for the resulting project example is shown in figure 1(D), with the selected activities colored dark. Based on these selected activities, we present a feasible project schedule in figure 1(D).

3.2.4. Caused choices

A caused choice implies that an alternative branch *must* be selected when another alternative branch is selected. More precisely, a caused choice between alternative branches k and k' implies that alternative branch $k' \in K_{p_{l'}}$ must be selected when alternative branch $k \in K_{p_l}$ is selected ($k \neq k'$). The alternative branches k and k' can belong to the same alternative subgraph ($l = l'$) or different alternative subgraphs ($l \neq l'$). We model caused choices by means of the parameter $\theta_{b_k,b_{k'}}$. When $\theta_{b_k,b_{k'}} = 1$, this indicates that the alternative branch with branching activity b_k causes the alternative branch with branching activity $b_{k'}$ and 0 otherwise. Eq. (13) is added to ensure that branching activity $b_{k'}$ is selected when branching activity b_k is selected and $\theta_{b_k,b_{k'}} = 1$.

$$y_{b_{k'}} \geq y_{b_k} * \theta_{b_k,b_{k'}} \qquad \forall k \in K_{p_l}; k' \in K_{p_{l'}}; \forall l, l' \in L \quad (13)$$

In figure 1(E), we show a caused choice between the alternative branches with branching activities $b_3 = 10$ and $b_7 = 27$, i.e. $\theta_{10,27} = 1$ (see table 6), as represented in the project example by means of a dotted arc. One possible project structure for the resulting project example is shown in figure 1(E), with the selected activities colored dark. Based on these selected activities, we present a feasible project schedule in figure 1(E).

3.2.5. Closed choices

A closed choice implies that two alternative branches *cannot* be selected simultaneously. More precisely, a closed choice between alternative branches k and k' implies that an alternative branch $k' \in K_{p_{l'}}$ cannot be selected when alternative branch $k \in K_{p_l}$ is selected ($k \neq k'$) and visa versa. The alternative branches k and k' can belong to the same alternative subgraph ($l = l'$) or different alternative subgraphs ($l \neq l'$). We model closed choices by means of the parameter $\partial_{b_k,b_{k'}}$. When $\partial_{b_k,b_{k'}} = 1$, the alternative branches with branching activity b_k and $b_{k'}$ cannot both be selected, while no closed choices exist when $\partial_{b_k,b_{k'}} = 0$. For the closed choices, eq. (14) is added to ensure that the branching activities b_k and $b_{k'}$ are not selected at the same time when $\partial_{b_k,b_{k'}} = 1$.

$$y_{b_{k'}} * \partial_{b_k,b_{k'}} \leq (1 - y_{b_k}) \qquad \forall k \in K_{p_l}; \forall k' \in K_{p_{l'}}; \forall l, l' \in L \quad (14)$$

In figure 1(F), we show a closed choice between the alternative branches with branching activities $b_6 = 23$ and $b_7 = 27$, i.e. $\partial_{23,27} = 1$ (see table 6), as represented in the project example by means of a crossed two-way dotted arc. One possible project structure for the resulting project example is shown in figure 1(F), with the selected activities colored dark. Based on these selected activities, we present a feasible project schedule in figure 1(F).

3.2.6. Split choices

Split choices imply that alternative branches of an alternative subgraph are assigned to different groups such that at least one alternative branch should be selected from each group. More specifically, alternative branches $k \in K_{p_l}$ in each alternative subgraph $l \in L$ belong to different groups $h \in H_l$. As a result, each alternative branch k belongs to a set $K_{p_l,h}$ with $h \in H_l$ and $l \in L$. Similar to the other extensions, one or multiple (ρ_l) alternative branches must be selected, but at least one alternative branch k must be selected from each group h as shown in eq. (15). As a result, the number of groups $|H_l|$ should thus be larger than or equal to the number of selected alternative branches ρ_l (i.e. $|H_l| \geq \rho_l$) for each alternative subgraph $l \in L$ in order to obtain a feasible solution. Eq. (5) and eq. (10) should be replaced by, respectively, eq. (16) without linked alternative branches and eq. (17) with linked alternative branches. Also, eq. (11) and eq. (12) should be replaced by eq. (18) and eq. (19) for the selection of multiple alternative branches, respectively, without or with nested alternative subgraphs. Finally, eq. (13) for the caused choices and eq. (14) for the closed choices should be, respectively, replaced by eq. (20) and eq. (21).

$$\sum_{k=1}^{|K_{p_l,h}|} y_{b_k} \geq 1 \qquad \forall h \in H_l; \forall l \in L \quad (18)$$

$$y_j \geq y_i \qquad \forall (i,j) \in A; \forall i, j \in N_{b_k}; \forall k \in K_{p_l,h}; \forall h \in H_l; \forall l \in L \quad (16)$$

$$y_j \geq y_i \qquad \forall (i,j) \in A; \forall i \in N_{b_k}; \forall j \in N_{b_{k'}}; \forall k \in K_{p_l,h}; \forall k' \in K_{p_{l'},h'};$$
$$k \neq k'; \forall h \in H_l; \forall h' \in H_{l'}; \forall l, l' \in L \quad (17)$$

$$\sum_{h=1}^{|H_l|} \sum_{k=1}^{|K_{p_l,h}|} y_{b_k} = \rho_l \qquad \forall l \in L \quad (18)$$

$$\sum_{h'=1}^{|H_{l'}|} \sum_{k'=1}^{|K_{p_{l'},h'}|} y_{b_{k'}} = \rho_l * y_{p_{l'}} \qquad \forall l' \in L \quad (19)$$

$$y_{b_{k'}} \geq y_{b_k} * \theta_{b_k,b_{k'}} \qquad \forall k \in K_{p_l,h}; k' \in K_{p_{l'},h'}; \forall h \in H_l; \forall h' \in H_{l'}; \forall l, l' \in L \quad (20)$$

$$y_{b_{k'}} * \partial_{b_k,b_{k'}} \leq (1 - y_{b_k}) \qquad \forall k \in K_{p_l,h}; k' \in K_{p_{l'},h'}; \forall h \in H_l; \forall h' \in H_{l'}; \forall l, l' \in L \quad (21)$$

In figure 1(G), we assume that there exist two groups $h \in H_l$ in both alternative subgraphs $l \in L$: $H_1 = \{1, 2\}$ and $H_2 = \{1, 2\}$. This is represented by means of an interrupted ')' in order to show the alternative branches that belong to the same group. In the alternative subgraph with principal activity $p_1 = 1$, the first group consists of the alternative branches with branching activity $b_1 = 2$ and $b_2 = 6$ (i.e. $K_{1,1} = \{2, 6\}$) and the second group consists of the alternative branches with branching activity $b_3 = 10$ and $b_4 = 14$ (i.e. $K_{1,2} = \{10, 14\}$). In the alternative subgraph with principal activity $p_2 = 18$, the first group consists of the alternative branches with branching activity $b_5 = 19$ and $b_6 = 23$ (i.e. $K_{18,1} = \{19, 23\}$) and the second group consists of the alternative branches with branching activity $b_7 = 27$ and $b_8 = 31$ (i.e. $K_{18,2} = \{27, 31\}$). We refer to table 6 for the adjusted sets in figure 1(G). One possible project structure for the resulting project example is shown in figure 1(G), with the selected activities colored dark. Based on these selected activities, we present a feasible project schedule in figure 1(G).

3.2.7. Non-renewable resources

Non-renewable resources (e.g. the project budget) are limited over the complete project horizon and they are not renewed after each time period. As a result, certain combinations of alternative branches will result in an infeasible solution as more non-renewable resources are required than there are available (Boctor 1993, 1996). Each non-dummy activity $i \in N$ requires $r^q_{i,w}$ units of resource type $w \in R^q$. Each non-renewable resource $w \in R^q$ has a limited availability a^q_w. Eq.(22) restricts the use of the non-renewable resources over the complete time horizon.

$$\sum_{i=1}^{n} r^q_{i,w} \sum_{t=es_i}^{ls_i} x_{i,t} \leq a^q_w \qquad \forall w \in R^q \quad (22)$$

TO DELETE Total model:

$$\textbf{Min } f_1 = \sum_{t=es_{n+1}}^{ls_{n+1}} tx_{n+1,t} \quad (1)$$

$$\textbf{Min } f_2 = C^{tot} \quad (1')$$

subject to

$$C^{tot} = C^d + C^i + C^t + C^f \quad (1a)$$

$$C_{max} = \sum_{t=es_{n+1}}^{ls_{n+1}} tx_{n+1,t} \quad (1b)$$

$$C^d = \sum_{i=1}^{n} \sum_{v=1}^{|R^p|} c^d \times r_{i,v} \times d_i \times x_{i,t} \quad (1c)$$

$$C^i = c^i \times C_{max} \quad (1d)$$

$$C^t = c^t \times max\{(C_{max} - h), 0\} \quad (1e)$$

$$\sum_{t=es_i}^{ls_i} (t + l_{i,j})x_{i,t} - M(1 - y_i) \leq \sum_{t=es_j}^{ls_j} tx_{j,t} + M(1 - y_j) \qquad \forall (i,j) \in A \quad (2)$$

$$\sum_{t=es_i}^{ls_i} x_{i,t} = y_i \qquad \forall i \in N \quad (3)$$

$$\sum_{h'=1}^{|H_{l'}|} \sum_{k'=1}^{|K_{p_{l'},h'}|} y_{b_{k'}} = \rho_l * y_{p_{l'}} \qquad \forall l' \in L \quad (19)$$

$$y_j \geq y_i \qquad \forall (i,j) \in A; \forall i \in N_{b_k}; \forall j \in N_{b_{k'}}; \forall k \in K_{p_l,h}; \forall k' \in K_{p_{l'},h'}; k \neq k'; \forall h \in H_l; \forall h' \in H_{l'}; \forall l, l' \in L \quad (17)$$

$$y_i = 1 \qquad \forall i \in N^f \quad (6)$$

$$\sum_{i=1}^{n} r_{i,v}^p \sum_{s=max(t-d_i, es_i)}^{min(t-1, ls_i)} x_{i,s} \leq a_v^p \qquad \forall v \in R^p; t = \{1, ..., T\} \quad (7)$$

$$y_{b_{k'}} \geq y_{b_k} * \theta_{b_k, b_{k'}} \qquad \forall k \in K_{p_l,h}; k' \in K_{p_{l'},h'}; \forall h \in H_l; \forall h' \in H_{l'}; \forall l, l' \in L \quad (20)$$

$$y_{b_{k'}} * \partial_{b_k, b_{k'}} \leq (1 - y_{b_k}) \qquad \forall k \in K_{p_l,h}; k' \in K_{p_{l'},h'}; \forall h \in H_l; \forall h' \in H_{l'}; \forall l, l' \in L \quad (21)$$

$$x_{i,t}; y_i \in 0, 1 \qquad \forall i \in N; t = \{1, ..., T\} \quad (8)$$

3.3. *Examples: production alternatives*

In the remainder of this section, we provide three examples obtained from the existing literature on production process optimization. These examples illustrate the practical relevance of the proposed model and concepts for production environments in a wide variety of industries such as cane sugar, hydrogen and aircraft component production. In each example, there exists a base case or reference choice that is traditionally used in the production process, but this base case is challenged by different innovative processes in order to optimise a wide variety of objectives such as resource savings, emission reduction and production costs.

Example 1: Basic case An intermediate stage in the cane sugar production process is the by-product valorization or transforming the waste in the cane sugar production into a valuable resource. Contreras et al. (2009) identified different alternatives for the by-product valorization that can be classified as two stages (waste water and filter cake emission) with four alternatives for each stage (similar to Figure 1(A)). For stage 1, the alternatives are to emit the wast water (option 1, base case), use the waste water as fertilizer (option 2), clean the waste water using anaerobic digestion (option 3) or use the waste water as fresh water after distillation (option 4). Independent of stage 1, the alternatives for stage 2 are to emit filter cake to the soil (option 1, base case), use the filter cake as fertilizer (option 2), use the filter cake as fertilizer after anaerobic digestion (option 3) or combine the filter cake and waste water as fertilizer after anaerobic digestion (option 4). The authors show that the selection of the alternatives has a major impact on the resource savings.

Example 2: Nested case Alternative processes for hydrogen production are constantly developed and analysed in order to minimize the resource requirements, emissions and costs (Dufour et al. 2012). These authors compare traditional alternatives, such as methane steam reforming and electrolysis, with innovative alternatives: water photosplitting, solar thermochemical cycles and automaintained methane decomposition. Where the first choice is to select one of these five alternative processes supported by completely different technologies, four of these five processes – except methane steam reforming – result in a follow-up choice or nested alternatives (similar to Figure 1(B)). For example, the hydrogen production via electrolysis can be executed using electricity from the grid (option 1), wind turbines (option 2) and photovoltaic panels (option 3).

Example 3: Caused and closed case The high costs associated with the production of composite aircraft components have resulted in an increased interest in alternative production methods (Witik et al. 2012). The reference process makes use of unidirectional carbon fibre prepregs (option 1), however, the industry has developed alternative processing technologies such as out-of-autoclave prepregs (option 2) and non-crimp fabrics (option 3). In a next stage, autoclave curing (option 1) is the reference process for treating these materials, but it can be compared against curing in both thermal ovens (option 2) or microwave ovens (option 3). However, the selected option in the first stage implies or excludes a choice in the second stage (similar to Figure 1(E-F)). On the one hand, the autoclave curing technique (option 1 in stage 2) should be selected when the unidirectional carbon fibre prepregs (option 1 in stage 1) are used (i.e. caused choice). On the other hand, the autoclave curing technique (option 1 in stage 2) cannot be selected when the out-of-autoclave prepregs (option 2 in stage 1) or the non-crimp fabrics (option 3 in stage 1) are used (i.e. closed choice).

Based on the above illustrative examples of the RCPSP-AS and its extensions as well as the practical cases from the production literature, we can summarise the main contributions of the proposed problem to project and production management practitioners:

(1) The proposed problem evaluates the impact of alternative execution methods and/or different risk mitigation strategies on the project or production schedule with renewable and non-renewable resources. Furthermore, managers are required to (proactively) define these risk

mitigation strategies upfront in the RCPSP-AS rather to (reactively) identify these strategies when a critical, uncertain event actually occurs.

(2) The explicit definition of the different types of relations between the alternative execution methods allows managers to consider the long-term impact of choices rather than focussing solely on the short-term benefits of certain execution methods.

(3) The ability to incorporate alternative execution methods for parts of the project or production process will stimulate managers to consider innovative technologies and alternative production processes in the decision-making process. The traditional execution methods can be compared with innovative execution methods, without any obligation to commit to these innovative technologies. Also, the traditional methods for some work packages can be combined with innovative technologies for other work packages. In other words, it is possible to model the interaction between different (base and innovative) technologies in the project or production process and motivate managers to innovate when this improves the corresponding scheduling objective.

4. Computational experiments

In this section, we will briefly discuss the parameter settings of the data instances used in this research (section 4.1). Also, we report the results of the computational experiments that analyse the impact of the extensions of the RCPSP-AS (section 4.2). Finally, we validate these extensions by means of three case studies (section 4.3). The model is coded in C++ and solved by GUROBI 6.5.0. The computational experiments in this study were carried out on a computer with an Intel Core i5 processor 2.5 GHz and 8 Gb RAM.

4.1. *Dataset*

We have generated a dataset of project instances using the data generation procedure described in Servranckx and Vanhoucke (2019a). The number of non-dummy activities in the project instances is set to [40,100] and project instances are generated for any combination of multiple selection, split choices, caused and closed choices, and non-renewable resources. The different parameters are discussed below and the parameter settings are summarised in table 7.

[Table 7 about here.]

Degree of flexibility (%flex) This parameter determines the number of alternatives in the project structure. This parameter is expressed as a percentage of the number of alternative branches that can potentially be included in the project structure as introduced by Servranckx and Vanhoucke (2019a).

Nested alternatives (%nested) The number of nested alternative subgraphs can be computed as the %nested multiplied by the actual number of alternative branches that can potentially be nested (Servranckx and Vanhoucke 2019a).

Linked alternatives (%linked) The number of linked alternative branches can be computed as the %linked multiplied by the actual number of alternative branches that can potentially be linked (Servranckx and Vanhoucke 2019a).

Multiple selection The impact of the number of selected alternative branches on the project makespan is investigated by varying the value of the parameter ρ_l between $\{1, ..., |K_l| - 1\}$. The maximum number of alternative branches $|K_l|$ in each alternative subgraph l is equal to 5 (Servranckx and Vanhoucke 2019a) and thus the number of selected alternative branches is equal to $\{2, 3, 4\}$ in our dataset when we consider multiple selection.

Split choices We require at least two alternative branches in each group to avoid a trivial selection subproblem and, therefore, there will be a maximum of two groups in our dataset since the project instances have a maximum of 5 alternative branches per alternative subgraph (Servranckx and Vanhoucke 2019a).

Caused choices (%caused) This parameter expresses the relative number of alternative branches that are caused.

Closed choices (%closed) This parameter expresses the relative number of alternative branches that are closed.

Non-renewable resources (NRR) For each activity, we draw a random value from the uniform interval [1,10] for the non-renewable resource requirements. In order to determine the availability of the non-renewable resources, we generate randomly 100 feasible solutions for each instance and determine the total non-renewable resource requirement of the selected activities. Subsequently, we construct an list of these total resource requirements in decreasing order and use the quartile values Q1, Q2 and Q3 as respectively the low, medium and high constrainedness of the non-renewable resources.

Considering the flexibility parameters introduced by Servranckx and Vanhoucke (2019a), we include instances with 25%, 50%, 75% and 100% degree of flexibility (%flex), while the %linked, %nested and flexibility distribution are fixed in this research. The aim is to limit the size of the dataset, while retaining the overall complexity of the project instances. In case that the parameters are fixed, they are set based on preliminary experiments and summarised in table 7. For each unique combination of parameter settings, a single project instance is generated, resulting in a total of 6,000 (=4*[(5*5*2*3)+(3*3*5*5*2*3)]) project instances. More precisely, four settings for the parameter %flex exist and, subsequently, a distinction should be made for the single selection and multiple selection cases. In case of a single selection, no split choices exist and hence only the parameter %caused (5), %closed (5), the number of NRR (2) and the constrainedness of the NRR (3) should be considered. In case of multiple selection, the same parameter settings should be generated for the different settings of the parameters multiple selection (3) and split choices (3).

[Table 8 about here.]

4.2. *Computational results*

[Table 9 about here.]

In this section, we analyse the impact of the proposed extensions on the average project makespan and the average runtime of the solver. We distinguish between the project instances for which an optimal solution could be found and the solutions for which a feasible solution could be found (including the instances that were solved to optimality) within a time limit of 100 seconds. We also report the relative number of instances that could be solved to optimality and the relative number of instances for which a feasible solution could be found. Finally, we provide the average optimality gap for the feasible solutions.

MULTIPLE SELECTION/SPLIT CHOICES (table 8) We observe that the project makespan increases as the number of selected alternative branches in each alternative subgraph increases. This could be expected, however, we observe that the increase is larger between 1 and 2 choice(s) as well as 3 and 4 choices, while the increase is smaller between 2 and 3 choices. This shows that the initial increase of selected activities increases the project makespan sufficiently such that the idle resource units can be used in an effective way when the number of selected activities is further increased. We also observe that the project makespan increases when split choices are considered since the degrees of freedom in the selection subproblem are more limited.

%FLEX (table 8) When we consider the impact of the parameter %flex, we would expect the project makespan to decrease as the degree of flexibility increases. However, we observe that the project makespan slightly increases as %flex increases. This can be explained by the fact that the number of alternative activities and thus the size of the project instances increases as %flex increases. As a result, it is much harder to find a high-quality solution using our solver as shown by the increasing optimality gap. In case of the lowest degree of flexibility (%flex=25%), all project instances can be solved to optimality within the time limit, however, the average optimality gap increases when the %flex increases.

[Table 10 about here.]

CAUSED CHOICES (table 9) We observe that the project makespan of the feasible solutions increases as the %caused increases, while the project makespan of the optimal solutions remains almost unchanged. As a result, the increased average makespan (Feasible) can be explained by the fact that an increased number of caused choices makes it harder to obtain feasible solutions. In case that %caused is over 25%, the larger number of implied selections results in a significant decrease in the number of optimal (%Opt) and feasible (%Feas) solutions. When the parameter %caused is higher than 75%, no optimal solution were obtained and only a small number of feasible solutions (= 2%) could be found.

CLOSED CHOICES (table 9) Considering the closed choices, we observe a decrease of the project makespan of the feasible solutions, however, the project makespan of the optimal solutions again remains stable in case that the %closed increases. We notice that the number of feasible solutions (%Feas) decreases as the %closed increases, however, the number of optimal solutions (%Opt) remains almost unchanged. As a result, the decrease in project makespan for the feasible solutions can only be explained by the fact that an increasing number of closed choices makes project instances with a high project makespan infeasible, resulting in a larger portion of project instances with a lower project makespan.

NON-RENEWABLE RESOURCES (table 10) A higher number of non-renewable resources or a higher constrainedness of the non-renewable resources will result, in general, in a higher project makespan (Avg. Makespan - Opt) although very limited. This is supported by the limited impact on the number of optimal (%Opt) and feasible (%Feas) solutions. When we consider three non-renewable resources instead of one non-renewable resource, the relative number of feasible solutions is reduced with around 5 percent points similar to the decrease in relative number of feasible solutions when the non-renewable resource constrainedness is increased from low to high (for both one and three non-renewable resources).

[Table 11 about here.]

4.3. *Case studies*

In this research, we also use the model introduced in section 3 to obtain (near-)optimal solutions for three case studies that include some of the extensions discussed in section 3.2. The contribution of this analysis is twofold. First of all, we show that the proposed extensions are observed in real-life projects and, secondly, we show that it is possible to obtain high-quality solutions for project instances with a realistic size in a reasonable computational time. In contrast to the artificial project instances of section 4.2, the size of the projects is larger, i.e. a higher number of activities, and the alternative project structures are more complex, i.e. a higher number of possible combinations of alternative branches.

Case information The key properties of the three case studies are shown in table 11. The first two cases are construction projects, while the third case considers a consultancy project. The three case studies will be explained in more detail:

Case 1 A construction project consisting of 24 living units. There exist links between the reinforced concrete elements, ventilation and electricity. For the ventilation and electricity pipes, there are two options with regard to the installation timing. It is possible to install the pipes inside a layer of concrete, but it is also possible to build them upon the concrete layer. When choosing the latter option, the ventilation and electricity should be installed later in the project, which has an impact on the sanitary works. Furthermore, the plastering of the ceilings will take longer when the ceiling is covered with pipes. Finally, a suspended ceiling needs to be added to cover the pipes resulting in additional time and work. The living units are divided over three different blocks around a central hall and each block consists of four floors. In order to ensure consistency in the construction project, choices with respect to the concrete flooring, ventilation and electricity in one living unit imply choosing the same options in other living units. Furthermore, this project should be executed with a limited project budget, which is considered a non-renewable resource in this project.

Case 2 This construction projects consists of over twenty houses and multiple apartment building with several living units each. In this project, the alternative technologies for structural work packages, such as the use of prefabricated columns versus on-the-spot fabricated columns, in one living unit imply choosing similar technologies in other living. On the one hand, prefabricated columns require more preparation work (drawing, control and production), but can be implemented faster. On the other hand, on-the-spot fabricated columns require less preparation work, but take longer to be installed. Since these columns should be fabricated in a single batch to ensure uniformity, the selected alternative should be the same for all living units. Case 2 is the only project in which the selection for one alternative technology closes the selection of another technology. More precisely, the type of plumbing in one living unit excludes certain types of plumbing in other living units since one needs to decide on installing the pipes inside or on top of the concrete layer. A similar situation exists in Case 1, where the ventilation and electricity in one living unit results in the same selection for another living unit (i.e. caused choices). In this case, however, the technical specifications of certain types of plumbing (i.e. inside or outside concrete layer) make them incompatible with certain other types of plumbing (i.e. closed choices).

Case 3 In this consultancy project, the alternative branches correspond with different team compositions (i.e. experience and expertise of the consultants) and their corresponding work methods. We distinguish between junior programmers, senior programmers and expert programmers or any combination of these three types of consultants. For example, certain activities will be completed faster in team compositions with experts, however, this will require the inclusion of additional activities in the project structure of such an alternative (e.g. mentoring activities). Furthermore, the sequence of activities will differ between team compositions with relatively more junior or senior programmers. In case that three alternative branches exist, any combination of two teams should be selected to complete the task (i.e. multiple selection). However, sometimes these teams are assigned to groups in order to ensure that at least one team of each group is represented (i.e. split choices).

[Table 12 about here.]

[Table 13 about here.]

Computational results Rather than to consider all extensions at once in our experiments, we create different scenarios in which we gradually activate the different extensions in the cases as shown in table 11. In scenario 1, only one extension is considered in each case (i.e. nested alternative subgraphs for all cases), while two extensions are considered in scenario 2. In scenario 3, three extensions are considered for each case, and this corresponds with the actual situation of case 2 (i.e. all extensions are included). Finally, the last extensions are included for cases 1 and 3 in scenario 4. In table 12, we present the numerical results for the three case studies in the four scenarios. Since the project makespan will increase in case that the extensions are included, independent of the performance of the solver, we report the optimality gap rather than the project makespan. The relative gap between the best-known solution and the best-known bound on the objective is shown for three time limits (100 s, 1,000 s and 3,600 s). These time limits are set (relatively) high compared to section 4.2 since more than 1,000 (case 1), 10,000 (case 2) and 10,000,000 (case 3) possible project structures exist in the case studies. We observe that the optimality gap increases as the scenarios include more extensions, independent of the time limit. For the most complex scenarios, we observe that no feasible solution could be found for small time limits due to the complex nature of the case studies. When increasing the time limit, one case study could be solved to optimality. For the other two case studies, the optimality gap could be reduced significantly, however, it remains relatively large at the highest time limit (3,600 s).

Rules-of-thumb Based on the case studies, we have also identified two main strategies used by the project managers to choose amongst alternatives in the project structure:

Sum of duration In each alternative subgraph, select the alternative branch(es) with the lowest sum of activity durations, independent of the relations between the alternative branches.

Work content In each alternative subgraph, select the alternative branch(es) with the lowest sum of work content (i.e. activity duration multiplied with renewable resource requirements), independent of the relations between the alternative branches.

We will use both strategies in order to obtain a solution for the three case studies in case that all extensions are included. The advantage of using rules-of-thumb is that a feasible solution can be found very fast, which was not the case for small time limits using the solver. In table 13, we show the deviation with the best found solution obtained for each of the three cases using both strategies. We observe that both strategies perform reasonably good (especially for case 1), which can be explained by the fact that the strategies consider characteristics of the alternatives (i.e. duration and resource requirements) that are relatively important for finding a good solution given the project objective. However, the solver reports better results than both strategies (especially for cases 2 and 3). This shows again the complexity of the resulting alternative project structures and the impact of the relations between the alternatives as modeled in the extensions of the RCPSP-AS. In our experiments, the sum of duration strategy slightly outperforms the work content strategy for each case. However, this cannot be considered a general rule as it is probably impacted by the resource constrainedness of the non-renewable resources in the three cases.

[Table 14 about here.]

5. Conclusions

In this research, we present several extensions of the Resource-Constraint Project Scheduling Problem with Alternative Subgraphs (RCPSP-AS) that allows us to consider more complex variants of the alternative project structure. We consider complex interdependencies between alternatives: nested alternative subgraphs and linked alternative branches. We focus on the selection of multiple alternatives, rather than a single alternative, for each work package. Also, we introduce caused and closed choices in order to model the case where one alternative implies another alternative or two alternatives cannot be selected at the same time. Finally, alternatives can be assigned to different groups within the work packages, so-called split choices. In this case, alternatives cannot be selected from one and the same group, but at least one alternative should be selected from each group. The objective of RCPSP-AS is to minimize the total makespan of the project subject to precedence relations, renewable and non-renewable resource constraints, and the constraints to ensure that the selection of activities (based on the basic problem formulation and the extensions) is valid.

First of all, we have formulated the basic RCPSP-AS as a ILP formulation and have extended this formulation to incorporate the proposed extensions of the RCPSP-AS. Secondly, we have illustrated the basic problem as well as the different extensions using an example project in order to show their impact on the scheduling process. Finally, we have conducted computational experiments based on an artificial dataset and three case studies. The problem was solved using Gurobi in order to find (near-)optimal solutions with a high computational efficiency. The contribution of this research was twofold: measuring the impact of the extensions of the RCPSP-AS on the solution quality and computational complexity, and validating the existence of these extensions in real-life cases. We observe that the runtime of the solver increases drastically as the number of (selected) activities increases, however, good (near-)optimal solution can be found even for the largest artificial project instances. Also, we notice that combinations of the different extensions result in a large number of infeasible solutions. Especially, a large number of caused and closed choices might result in complex selection subproblems making it impossible to find a feasible combination of selected activities. The case studies with a higher number of activities and alternatives as well as more complex alternative project structures indicate that it is hard to find a feasible solution given a low runtime limit or an optimal solution given a higher runtime limit.

In future research, more efficient exact methods for solving the RCPSP-AS and its extensions could be developed both for artificial instances and real-life project. Also, more extensions for RCPSP-AS could be defined based on existing academic research, such as a multi-objective variant of the RCPSP-AS or the RCPSP-AS with resource flexibility, or based on other extensions observed in practice. Finally, the proposed model only considers the time objective and does not take into account other objectives such as the cost objective. This limitation of the current research study could be resolved by extending the RCPSP-AS with a cost objective in order to select alternatives based on fixed and variable costs. A tardiness cost related to the project deadline or intermediate deadlines for each work package can be used to incorporate the time aspect into this cost objective.

Acknowledgments

This work was supported by the Nationale Bank van België (NBB) and the Bijzonder Onderzoeksfonds (BOF) under Grant BOFGOA021.

Declaration of interest

No potential conflict of interest was reported by the authors.

References

Abdullah, S., and M. Abdolrazzagh-Nezhad. 2014. "Fuzzy job-shop scheduling problems: A review." *Information Sciences* 278: 380–407.

Alfieri, A., T. Tolio, and M. Urgo. 2011. "A project scheduling approach to production planning with feeding precedence relations." *International Journal of Production Research* 49 (4): 995–1020.

Alfieri, A., T. Tolio, and M. Urgo. 2012. "A project scheduling approach to production and material requirement planning in Manufacturing-to-Order environments." *Journal of Intelligent Manufacturing* 23 (3): 575–585.

Almeida, B. F., I. Correia, and F. Saldanha-da Gama. 2016. "Priority-Based Heuristics for the Multi-Skill Resource Constrained Project Scheduling Problem." *Expert Systems with Applications* 57: 91–103.

Almeida, Correia I., B.F., and F. Saldanha-da Gama. 2018. "A biased random-key genetic algorithm for the project scheduling problem with flexible resources." *TOP* 26: 283–308.

Ballestín, F. 2007. "A Genetic Algorithm for the Resource Renting Problem with Minimum and Maximum Time Lags." *Lecture Notes in Computer Science* 4446: 25–35.

Ballestín, F. 2008. "Different Codifications and Metaheuristic Algorithms for the Resource Renting Problem with Minimum and Maximum Time Lags." In *Recent Advances in Evolutionary Computation for Combinatorial Intelligence*, edited by C. Cotta and J. van Hemert, Vol. 153 of *Studies in Computational Intelligence*, Chap. 12, 187–202. Springer-Verlag Berlin Heidelberg.

Bartusch, M., R.H. Möhring, and F.J. Radermacher. 1988. "Scheduling project networks with resource constraints and time windows." *Annals of Operations Research* 16 (1): 199–240.

Bianco, L., M. Caramia, and S. Giordani. 2016. "Resource levelling in project scheduling with generalized precedence relationships and variable execution intensities." *OR Spectrum* 38: 405–425.

Boctor, F.F. 1993. "Heuristics for scheduling projects with resource restrictions and several resource-duration modes." *International Journal of Production Research* 31: 2547–2558.

Boctor, F.F. 1996. "A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes.” *European Journal of Operational Research* 90: 349–361.

Cajzek, R., and U. Klanšek. 2019. “Cost optimization of project schedules under constrained resources and alternative production processes by mixed-integer nonlinear programming.” *Engineering, Construction and Architectural Management* 26 (10): 2474–2058.

Capek, R., P. Šucha, and Z. Hanzálek. 2012. “Production scheduling with alternative process plans.” *European Journal of Operational Research* 217 (2): 300 – 311.

Carvalho, A. N., F. Oliveira, and L. F. Scavarda. 2015. “Tactical capacity planning in a real-world ETO industry case: An action research.” *International Journal of Production Economics* 167: 187–203.

Carvalho, A. N., F. Oliveira, and L. F. Scavarda. 2016. “Tactical capacity planning in a real-world ETO industry case: A robust optimization approach.” *International Journal of Production Economics* 180: 158–171.

Contreras, A. M., E. Rosa, M. Pérez, H. Van Langenhove, and J. Dewulf. 2009. “Comparative Life Cycle Assessment of four alternatives for using by-products of cane sugar production.” *Journal of Cleaner Production* 17 (8): 772–779.

Correia, I., and F. Saldanha-da Gama. 2014. “The impact of fixed and variable costs in a multi-skill project scheduling problem: An empirical study.” *Computers & Industrial Engineering* 72: 230–238.

Correia, I., and F. Saldanha-da Gama. 2015. “A Modeling Framework for Project Staffing and Scheduling Problems.” In *Handbook on Project Management and Scheduling Vol.1*, edited by C. Schwindt and J. Zimmermann, Vol. 1 of *International Handbooks on Information Systems*, 547–564. Springer.

Correia, I., L.L. Womer, and F. Saldanha da Gama. 2012. “Project scheduling with flexible resources: formulation and inequalities.” *Journal of Scheduling* 35: 635–663.

Dai, M., D. Tang, K. Zheng, and Q. Cai. 2013. “An Improved Genetic-Simulated Annealing Algorithm Based on a Hormone Modulation Mechanism for a Flexible Flow-Shop Scheduling Problem.” *Advances in Mechanical Engineering* 2013: 1–13.

Demeulemeester, E., and W. Herroelen. 2002. *Project scheduling: A research handbook*. Kluwer Academic Publishers.

Dufour, J., D. P. Serrano, J. L. Gálvez, A. González, E. Soria, and J. L.G. Fierro. 2012. “Life cycle assessment of alternatives for hydrogen production from renewable and fossil sources.” *International Journal of Hydrogen Energy* 37 (2): 1173–1183.

Fündeling, C.-U., and N. Trautmann. 2010. “A priority-rule method for project scheduling with work-content constraints.” *European Journal of Operational Research* 203 (3): 568–574.

Ghiyasinasab, M., N. Lehoux, S. Ménard, and C. Cloutier. 2020. “Production planning and project scheduling for engineer-to-order systems-case study for engineered wood production.” *International Journal of Production Research* 1–20.

Gong, G., Q. Deng, X. Gong, W. Liu, and Q. Ren. 2018. “A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators.” *Journal of Cleaner Production* 174: 560–576.

Hartmann, S., and D. Briskorn. 2010. “A survey of variants and extensions of the resource-constrained project scheduling problem.” *European Journal of Operational Research* 207: 1–15.

Herroelen, W., B. De Reyck, and E. Demeulemeester. 1998. “Resource-constrained project scheduling: A survey of recent developments.” *Computers & Operations Research* 25: 279–302.

Jiang, X. S., and X. P. Ma. 2016. “On the multi-objective optimization method of the flexible job-shop scheduling problem based on ant colony algorithm.” *Revista de la Facultad de Ingeniería* 31 (7).

Kellenbrink, C., and S. Helber. 2015. “Scheduling resource-constrained projects with a flexible project structure.” *European Journal of Operational Research* 246 (2): 379 – 391.

Kellenbrink, C., and S. Helber. 2016. “Quality-and-profit-oriented scheduling of resource-constrained projects with flexible project structures via a genetic algorithm.” *European Journal of Industrial Engineering* 10 (5).

Kerkhove, L.-P., M. Vanhoucke, and B. Maenhout. 2017. “On the resource renting problem with overtime.” *Computers & Industrial Engineering* 111: 303–319.

Kis, T. 2003. “Job-shop scheduling with processing alternatives.” *European Journal of Operational Research* 151 (2): 307 – 332.

Kis, T. 2005. “A branch-and-cut algorithm for scheduling of projects with variable-intensity activities.” *Mathematical Programming* 103: 515–539.

Kolisch, R., K. Meyer, R. Mohr, C. Schwindt, and M. Urmann. 2003. “Ablaufplanung für die Leitstrukturoptimierung in der Pharmaforschung.” *Zeitschrift für Betriebswirtschaft* 73 (8): 825–848.

Kosztyán, Z. T. 2015. “Exact algorithm for matrix-based project planning problems.” *Expert Systems with Applications* 42 (9): 4460 – 4473.

Kosztyán, Z. T., and I. Szalkai. 2018. "Hybrid time-quality-cost trade-off problems." *Operations Research Perspectives* 5: 306 – 318.

Kosztyán, Z. T., and I. Szalkai. 2020. "Multimode resource-constrained project scheduling in flexible projects." *Journal of Global Optimization* 76 (1): 211–241.

Kreter, S., J. Rieck, and J. Zimmermann. 2014. "The total adjustment cost problem:Applications, models, and solution algorithms." *Journal of Scheduling* 17: 145–160.

Lei, D., Y. Zheng, and X. Guo. 2017. "A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption." *International Journal of Production Research* 55 (11): 3126–3140.

Li, H., and K. Womer. 2009. "Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm." *Journal of Scheduling* 12: 281.

Li, Y., Y. He, Y. Wang, F. Tao, and J. W. Sutherland. 2020. "An optimization method for energy-conscious production in flexible machining job shops with dynamic job arrivals and machine breakdowns." *Journal of Cleaner Production* 254: 120009.

Mastrolilli, M., and L. M. Gambardella. 2000. "Effective neighbourhood functions for the flexible job shop problem." *Journal of Scheduling* 3 (1): 3–20.

Naber, A., and R. Kolisch. 2014. "MIP models for resource-constrained project scheduling with flexible resource profiles." *European Journal of Operational Research* 239 (2): 335–348.

Najid, N.M., S. Dauzere-Peres, and A. Zaidat. 2002. "A modified simulated annealing method for flexible job shop scheduling problem." In *IEEE international conference on systems, man and cybernetics*, Vol. 5, 6–pp. IEEE.

Nübel, H. 2001. "The resource renting problem subject to temporal constraints." *OR Spektrum* 23: 359–381.

Palacios, J. J., J. Puente, C. R Vela, and I. Gonzalez-Rodriguez. 2016. "Benchmarks for fuzzy job shop problems." *Information Sciences* 329: 736–752.

Pritsker, A., L. Watters, and P. Wolfe. 1969. "Multi-project scheduling with limited resources: a zero-one programming approach." *Management Science* 16: 93–108.

Ranjbar, M., and F. Kianfar. 2010. "Resource-Constrained Project Scheduling Problem with Flexible Work Profiles: A Genetic Algorithm Approach." *Scientia Iranica* 17 (1): 25–35.

Servranckx, T., and M. Vanhoucke. 2019a. A tabu search procedure for the resource-constrained project scheduling problem with alternative subgraphs." *European Journal of Operational Research* 273 (3): 841 – 860.

Servranckx, T., and M. Vanhoucke. 2019b. Strategies for project scheduling with alternative subgraphs under uncertainty: similar and dissimilar sets of schedules." *European Journal of Operational Research* 279 (1): 38 – 53.

Servranckx, T., M. Vanhoucke, and G. Vanhouwaert. 2020. "Analysing the impact of alternative network structures on resource-constrained schedules : artificial and empirical experiments." *Computers & Industrial Engineering* 148: 106706.

Tang, D., M. Dai, M.A. Salido, and A. Giret. 2016. "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization." *Computers in Industry* 81: 82–95.

Tao, S., and Z.S. Dong. 2017. "Scheduling resource-constrained project problem with alternative activity chains." *Computers & Industrial Engineering* 114: 288 – 296.

Tao, S., and Z.S. Dong. 2018. "Multi-mode resource-constrained project scheduling problem with alternative project structures." *Computers & Industrial Engineering* 125: 333 – 347.

Tao, S., C. Wu, Z. Sheng, and X. Wang. 2018. "Stochastic Project Scheduling with Hierarchical Alternatives." *Applied Mathematical Modelling* 58: 181 – 202.

Tritschler, M., A. Naber, and R. Kolisch. 2017. "A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles." *European Journal of Operational Research* 262 (1): 262–273.

Vandenheede, L., M. Vanhoucke, and B. Maenhout. 2016. "A scatter search for the extended resource renting problem." *International Journal of Production Research* 54 (16): 4723–4743.

Vela, C. R., S. Afsar, J. J. Palacios, I. González-Rodríguez, and J. Puente. 2020. "Evolutionary tabu search for flexible due-date satisfaction in fuzzy job shop scheduling." *Computers & Operations Research* 119: 104931.

Witik, R. A., F. Gaille, R. Teuscher, H. Ringwald, V. Michaud, and J.-A. E. Møanson. 2012. "Economic and environmental assessment of alternative production methods for composite aircraft components." *Journal of Cleaner Production* 29-30: 91–102.

Zhang, G., X. Shao, P. Li, and L. Gao. 2009. "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem." *Computers & Industrial Engineering* 56 (4): 1309–1318.

Research field	Research paper	Research topic
RCPSP with alternative project structures	Capek, Šucha, and Hanzálek (2012)	Production scheduling with alternative process plans
	Kellenbrink and Helber (2015)	Project scheduling with flexible project structures
	Kellenbrink and Helber (2016)	Project scheduling with flexible project structures and quality-profit perspective
	Tao and Dong (2017)	Project scheduling with alternative activity chains
	Tao and Dong (2018)	Multi-mode project scheduling with alternative project structures
	Tao et al. (2018)	Stochastic project scheduling with hierarchical alternatives
	Kosztyán (2015)	Project scheduling with flexible dependencies
	Kosztyán and Szalkai (2018)	Project scheduling with flexible dependencies and time-quality-cost trade-off
	Kosztyán and Szalkai (2020)	Multi-mode project scheduling with flexible dependencies
	Cajzek and Klanšek (2019)	Project scheduling with alternative production processes
	Servranckx and Vanhoucke (2019a)	Project scheduling with alternative subgraphs
	Servranckx and Vanhoucke (2019b)	Project scheduling with alternative subgraphs under uncertainty
	Servranckx, Vanhoucke, and Vanhouwaert (2020)	Project scheduling with alternative subgraphs: schedule diversity and choice frequency
	This research	Project scheduling with alternative subgraphs and extensions
RCPSP with flexible resource profiles	Kolisch et al. (2003)	Project scheduling with resource flexibility
	Fündeling and Trautmann (2010)	Project scheduling with fixed work content constraints
	Ranjbar and Kianfar (2010)	Project scheduling with flexible resource profiles: genetic algorithm
	Tritschler, Naber, and Kolisch (2017)	Project scheduling with flexible resource profiles: hybrid metaheuristic
	Naber and Kolisch (2014)	Project scheduling with flexible resource profiles: MIP models
	Kis (2005)	Project scheduling with absolute variable-intensity activities
	Bianco, Caramia, and Giordani (2016)	Project scheduling with cumulative variable-intensity activities
PSP with flexible resources	Li and Womer (2009)	Project scheduling with multi-skilled personnel
	Correia, Womer, and da Gama (2012)	Project scheduling with flexible resources
	Almeida, Correia, and Saldanha-da Gama (2016)	Multi-skill resource-constrained project scheduling problem
	Almeida and Saldanha-da Gama (2018)	Project scheduling problem with flexible resources
	Correia and Saldanha-da Gama (2014)	Multi-skill project scheduling problem with fixed and variable costs
	Correia and Saldanha-da Gama (2015)	Project staffing and scheduling problems framework
Resource renting problem	Nübel (2001)	Resource renting problem with activity-resource flexibility
	Ballestín (2007)	Resource renting problem with minimum and maximum time-lags
	Ballestín (2008)	Resource renting problem: different metaheuristics
	Vandenheede, Vanhoucke, and Maenhout (2016)	Extended resource renting problem with total adjustement cost problem
	Kerkhove, Vanhoucke, and Maenhout (2017)	Resource renting problem with overtime
Flexible job shop scheduling	Kis (2003)	Job shop scheduling with processing alternatives
	Li et al. (2020)	Flexible machine job shop scheduling with dynamic job arrivals
	Mastrolilli and Gambardella (2000)	Flexible job shop scheduling: tabu search with effective local searches and neighbourhoods
	Najid, Dauzere-Peres, and Zaidat (2002)	Flexible job shop scheduling: modified simulated annealing
	Dai et al. (2013)	Flexible flow shop scheduling: improved genetic-simulated annealing
	Tang et al. (2016)	Flexible flow shop scheduling with energy-efficiency
	Jiang and Ma (2016)	Multi-objective flexible job shop scheduling: ant colony algorithm
	Zhang et al. (2009)	Flexible job shop scheduling to minimize completion time and workload of machines
	Lei, Zheng, and Guo (2017)	Flexible job shop scheduling with energy consumption
	Gong et al. (2018)	Flexible job shop scheduling with human factors
	Vela et al. (2020)	Fuzzy job shop scheduling with flexible due dates
	Abdullah and Abdolrazzagh-Nezhad (2014)	Fuzzy job shop scheduling: overview
	Palacios et al. (2016)	Fuzzy job shop scheduling: benchmarks

Table 1. Literature related to the RCPSP-AS in project management and production research

Symbol	Definition
Indices	
i	Activities
l	Alternative subgraphs
k	Alternative branches
v	Renewable resource type
t	Time
Decision variables	
$x_{i,t}$	Activity i is started at time instance t (1); otherwise (0)
y_i	Activity i is selected (1); otherwise (0)
Sets	
N	Set of activities $i \in N$
N^a	Set of alternative activities $N^a \subset N$
N^f	Set of fixed activities $N^f \subset N$
L	Set of alternative subgraphs l with $(p_l, t_l) \in L$ with principal activity p_l and terminal activity t_l
K_{p_l}	Set of alternative branches with branching activities $b_k \in K_{p_l}$ in alternative subgraph with principal activity p_l $(l \in L)$
N_{b_k}	Set of activities in alternative branch k with $N_{b_k} \subset N^a$
R^p	Set of renewable resources $v \in R^p$
Parameters	
p_l	Principal activity of alternative subgraph l
t_l	Terminating activity of alternative subgraph l
b_k	Branching activity of alternative branch k
$r^p_{i,v}$	Renewable resource requirement of activity i for type v
a^p_v	Availability of renewable resource v

Table 2. Summary of terminology of basic RCPSP-AS

Minimal time-lag		Maximal time-lag	
$FS_{i,j}^{min}$	$l_{i,j} = d_i + FS_{i,j}^{min}$	$FS_{i,j}^{max}$	$l_{j,i} = -d_i - FS_{i,j}^{max}$
$SS_{i,j}^{min}$	$l_{i,j} = SS_{i,j}^{min}$	$SS_{i,j}^{max}$	$l_{j,i} = -SS_{i,j}^{max}$
$FF_{i,j}^{min}$	$l_{i,j} = d_i - d_j + FF_{i,j}^{min}$	$FF_{i,j}^{max}$	$l_{j,i} = d_j - d_i - FF_{i,j}^{max}$
$SF_{i,j}^{min}$	$l_{i,j} = SF_{i,j}^{min} - d_j$	$SF_{i,j}^{max}$	$l_{j,i} = d_j - SF_{i,j}^{max}$

Table 3. Values for parameter $l_{i,j}$ in Eq. (2)

Symbol	Example (figure 1)
Activities	
N	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26, 27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44}
N^a	{2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28, 29,30,31,32,33,34}
N^f	{1,18,35,36,37,38,39,40,41,42,43,44}
d_i	{0,0,4,2,0,0,9,8,0,0,8,12,0,0,10,8,0,0,0,2,5,0,0,2,1,0,0,2,4,0,0,3,6,0,0,3, 2,0,4,5,0,1,2,0}
Resources	
R^p	{1}
$r^p_{i,v}$	{((1,3),3),((1,4),4),((1,7),2),((1,8),6),((1,11),1),((1,12),5),((1,15),7), ((1,16),8),((1,20),5),((1,21),4),((1,24),6),((1,25),9),((1,28),4),((1,29),2), ((1,32),6),((1,33),1),((1,36),3),((1,37),6),((1,39),3),((1,40),6),((1,42),7), ((1,43),3)}
a^p_v	{(1,10)}
Alternative subgraphs	
L	$\{(1, 18), (18, 35)\}$
p_l	$p_1 = 1; p_2 = 18$
t_l	$t_1 = 18; t_2 = 35$
Alternative branches	
K_{p_l}	$K_1 = \{2, 6, 10, 14\}; K_{18} = \{19, 23, 27, 31\}$
N_{b_k}	$N_2 = \{2, 3, 4, 5\}; ...; N_{30} = \{31, 32, 33, 34\}$
b_k	$b_1 = 2; b_2 = 6; b_3 = 10; b_4 = 14; b_5 = 19; b_6 = 23; b_7 = 27; b_8 = 31$

Table 4. Notations explained based on figure 1

Extension	Symbol	Definition
Multiple selection	ρ_l	Number of alternative branches $k \in K_{p_l}$ that should be selected in alternative subgraph $l \in L$
Caused choice	$\theta_{b_k,b_{k'}}$	Alternative branch k causes alternative branch k'
Closed choice	$\partial_{b_k,b_{k'}}$	Alternative branch k closes alternative branch k'
Split choices	h	Groups of alternative branches
	H_l	Set of groups in alternative subgraph l with $h \in H_l$
	$K_{p_l,h}$	Set of alternative branches in group h of alternative subgraph l with $k \in K_{p_l,h}$
Non-renewable resources	w	Non-renewable resource type
	R^q	Set of non-renewable resources $w \in R^q$
	$r^q_{i,w}$	Non-renewable resource requirement of activity i for type w
	a^q_w	Availability of non-renewable resource w

Table 5. Summary of terminology for the extensions of the RCPSP-AS

Extension	Symbol	Example
Nested alternative	N_{b_k}	$N_2 = \{2, 3, 4, 5, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34\}$
subgraphs	t_l	$t_1 = 35; t_2 = 35$
Linked alternative branches	N_{b_k}	$N_{19} = \{19, 20, 21, 22, 24, 26\}$
Multiple choices	ρ_l	$\rho_1 = 2; \rho_2 = 2$
Caused choices	$\theta_{b_k,b_{k'}}$	$\theta_{10,27} = 1$
Closed choices	$\partial_{b_k,b_{k'}}$	$\partial_{23,27} = 1$
Split choices	H_l	$H_1 = \{1, 2\}; H_2 = \{1, 2\}$
	$K_{p_l,h}$	$K_{1,1} = \{2, 6\}; K_{1,2} = \{10, 14\}; K_{18,1} = \{19, 23\}; K_{18,2} = \{27, 31\}$
Non-renewable	R^q	{1}
resources	$r^q_{i,w}$	{((1,3),3),((1,4),4),((1,7),2),((1,8),6),((1,11),1),((1,12),5),((1,15),7), ((1,16),8),((1,20),5),((1,21),4),((1,24),6),((1,25),9),((1,28),4),((1,29),2), ((1,32),6),((1,33),1),((1,36),3),((1,37),6),((1,39),3),((1,40),6),((1,42),6), ((1,43),3)}
	a^q_w	{(1,50)}

Table 6. Notations explained based on figure 1

	%flex	Flex. Distr.	SP	RC	#NRR	Degree NRR	
ASLIB-1	{0.25,0.5,0.75,1}	{E,M}	{0.25,0.5,0.75}	{0.25,0.5,0.75}	-	-	
ASLIB-2	0.75	M	0.5	0.5	-	-	{
ASLIB-3	{0.25,0.5,0.75,1}	{E,M}	{0.25,0.5,0.75}	{0.25,0.5,0.75}	-	-	{
ASLIB-4	{0.25,0.5,0.75,1}	{E,M}	{0.25,0.5,0.75}	{0.25,0.5,0.75}	-	-	
ASLIB-5	{0.25,0.5,0.75,1}	{E,M}	{0.25,0.5,0.75}	{0.25,0.5,0.75}	-	-	
ASLIB-6	{0.25,0.5,0.75,1}	{E,M}	{0.25,0.5,0.75}	{0.25,0.5,0.75}	-	-	
ASLIB-7	{0.25,0.5,0.75,1}	{E,M}	{0.25,0.5,0.75}	{0.25,0.5,0.75}	{1,3}	{L,M,H}	

Parameter symbol	**Parameter settings**	**Parameter symbol**	**Parameter settings**
%flex	0.25;0.5;0.75;1	ρ_l	1;2;3;4
%linked	0.1667	$max\{\|H_l\|\}$	2
%nested	1	%caused	0;0.25;0.5;0.75;1
$\|N_{k_l}\|$	2	%closed	0;0.25;0.5;0.75;1
		$\|R^p\|$	1
		$\|R^q\|$	1;3

Table 7. Summary of parameter definitions and settings

		Multiple choices				Split choices	
		1	2	3	4	Yes	No
%flex: 25%	Avg. Makespan - Opt	25.03	37.17	38.16	49.92	33.76	39.19
	Avg. Runtime (s) - Opt	0.96	6.01	10.93	55.64	8.64	14.25
	Avg. Makespan - Feas	25.03	37.17	38.16	49.92	33.76	39.19
	Avg. Runtime (s) - Feas	0.96	6.01	10.93	55.64	8.64	14.25
	%Opt	0.7733	0.7600	0.7600	0.1733	0.5167	0.4533
	%Feas	0.7733	0.7600	0.7600	0.1733	0.5167	0.4533
	Avg. %Optgap	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
%flex: 50%	Avg. Makespan - Opt	27.56	40.85	54.05	72.00	33.25	44.16
	Avg. Runtime (s) - Opt	8.95	53.94	69.87	68.21	25.68	57.95
	Avg. Makespan - Feas	27.56	55.11	91.87	103.99	64.43	80.65
	Avg. Runtime (s) - Feas	8.95	77.24	97.24	98.90	66.63	89.59
	%Opt	0.5867	0.3156	0.0422	0.0156	0.2367	0.1267
	%Feas	0.5867	0.6378	0.4511	0.4378	0.5267	0.5100
	Avg. %Optgap	0.0000	0.1636	0.4032	0.3579	0.2049	0.2932
%flex: 75%	Avg. Makespan - Opt	26.31	34.85	50.00	72.00	40.10	50.98
	Avg. Runtime (s) - Opt	8.06	24.14	27.13	29.50	18.50	26.84
	Avg. Makespan - Feas	26.31	75.40	121.44	130.43	81.00	105.33
	Avg. Runtime (s) - Feas	8.06	64.93	61.81	56.64	45.92	62.09
	%Opt	0.4333	0.2067	0.1911	0.1644	0.2500	0.1867
	%Feas	0.4333	0.4467	0.3644	0.2667	0.3767	0.3600
	Avg. %Optgap	0.0000	0.2677	0.3287	0.2651	0.1991	0.2917
%flex: 100%	Avg. Makespan - Opt	26.00	45.07	54.11	73.79	37.53	48.24
	Avg. Runtime (s) - Opt	6.95	46.38	51.16	72.68	26.68	48.39
	Avg. Makespan - Feas	26.00	52.97	80.49	91.02	51.24	65.47
	Avg. Runtime (s) - Feas	6.95	62.75	67.42	83.72	47.91	68.16
	%Opt	0.6000	0.4156	0.3242	0.1153	0.3267	0.2544
	%Feas	0.6000	0.5978	0.5142	0.2865	0.4667	0.4200
	Avg. %Optgap	0.0000	0.0806	0.1488	0.1260	0.0663	0.1009
AVG	Avg. Makespan - Opt	26.22	39.48	49.08	66.93	36.16	45.64
	Avg. Runtime (s) - Opt	6.23	32.62	39.77	56.61	19.88	36.86
	Avg. Makespan - Feas	26.22	55.17	82.99	93.84	57.61	72.66
	Avg. Runtime (s) - Feas	6.23	52.73	59.35	73.70	42.28	58.53
	%Opt	0.5983	0.4244	0.3294	0.1172	0.3325	0.2553
	%Feas	0.5983	0.6106	0.5224	0.2911	0.4717	0.4358
	Avg. %Optgap	0.0000	0.1280	0.2202	0.1873	0.1176	0.1715

Table 8. Impact of multiple selection and split choices for different %flex

	%caused				
	0	0.25	0.50	0.75	1
Avg. Makespan - Opt	40.76	36.59	41.06	40.53	-
Avg. Runtime (s) - Opt	32.54	21.39	20.86	11.77	-
Avg. Makespan - Feas	51.36	70.85	95.09	130.36	94.79
Avg. Runtime (s) - Feas	42.76	58.31	61.04	69.84	100.02
%Opt	0.7658	0.3750	0.1925	0.0975	0.0000
%Feas	0.9517	0.7567	0.3617	0.1608	0.0200
Avg. %Optgap	0.0775	0.1901	0.2552	0.4202	0.4221
	%closed				
	0	0.25	0.50	0.75	1
Avg. Makespan - Opt	38.77	42.10	40.57	40.93	41.01
Avg. Runtime (s) - Opt	30.70	29.07	27.99	26.09	24.67
Avg. Makespan - Feas	86.87	68.87	54.13	52.83	52.14
Avg. Runtime (s) - Feas	61.99	53.36	47.20	43.45	42.37
%Opt	0.2833	0.3075	0.2792	0.2792	0.2817
%Feas	0.5800	0.4908	0.3983	0.3900	0.3917
Avg. %Optgap	0.2553	0.1621	0.0974	0.0825	0.0742

Table 9. Impact of caused and closed choices

# NRR	1			3		
Degree NRR	Low	Medium	High	Low	Medium	High
Avg. Makespan - Opt	40.21	40.62	40.56	40.24	41.65	41.48
Avg. Runtime (s) - Opt	30.43	38.90	26.79	26.93	29.31	26.45
Avg. Makespan - Feas	71.16	68.27	69.93	66.70	59.47	59.16
Avg. Runtime (s) - Feas	54.88	53.37	51.20	51.38	49.10	47.28
%Opt	0.2970	0.2820	0.2860	0.2880	0.2980	0.2660
%Feas	0.4950	0.4700	0.4580	0.4530	0.4330	0.3920
Avg. %Optgap	0.1818	0.1743	0.1625	0.1449	0.1115	0.2552

Table 10. Impact of non-renewable resources

Basic information		Case 1	Case 2	Case 3
	$\|N\|$	204	97	176
	$\|L\|$	13	16	16
	$\|K_l\|$	2	2	{2,3}
Extensions				
Scenario 1				
	Nested alternatives	X	X	X
Scenario 2				
	Nested alternatives	X	X	X
	Linked alternatives	X		
	Multiple selection			X
	Caused choices		X	
Scenario 3				
	Nested alternatives	X	X	X
	Linked alternatives	X		
	Multiple selection			X
	Caused choices	X	X	X
	Closed choices		X	
Scenario 4				
	Nested alternatives	X		X
	Linked alternatives	X		
	Multiple selection			X
	Caused choices	X		X
	Closed choices			
	Split choices			X
	Non-renewable resources	X		

Table 11. Basic information and extensions in four scenarios for the three case studies

Scenario	**Time limit (s)**	**Case 1**	**Case 2**	**Case 3**
1	100	0.8810	0.0846	0.7636
	1,000	0.2864	0.0000	0.1190
	3,600	0.0519	0.0000	0.0387
2	100	T	0.1073	0.9108
	1,000	0.2652	0.0021	0.1448
	3,600	0.0577	0.0000	0.0410
3	100	T	0.1295	T
	1,000	0.4015	0.0055	0.1851
	3,600	0.0882	0.0000	0.0850
4	100	T	-	T
	1,000	0.4236	-	0.2380
	3,600	0.1065	-	0.0850

Table 12. Optimality gap for different runtime limits in each case (with T = time limit reached)

Strategy	**Case 1**	**Case 2**	**Case 3**
Sum of duration	0.4089	0.1831	0.2885
Work content	0.4268	0.2262	0.2857

Table 13. Performance of the strategies: sum of duration and work content

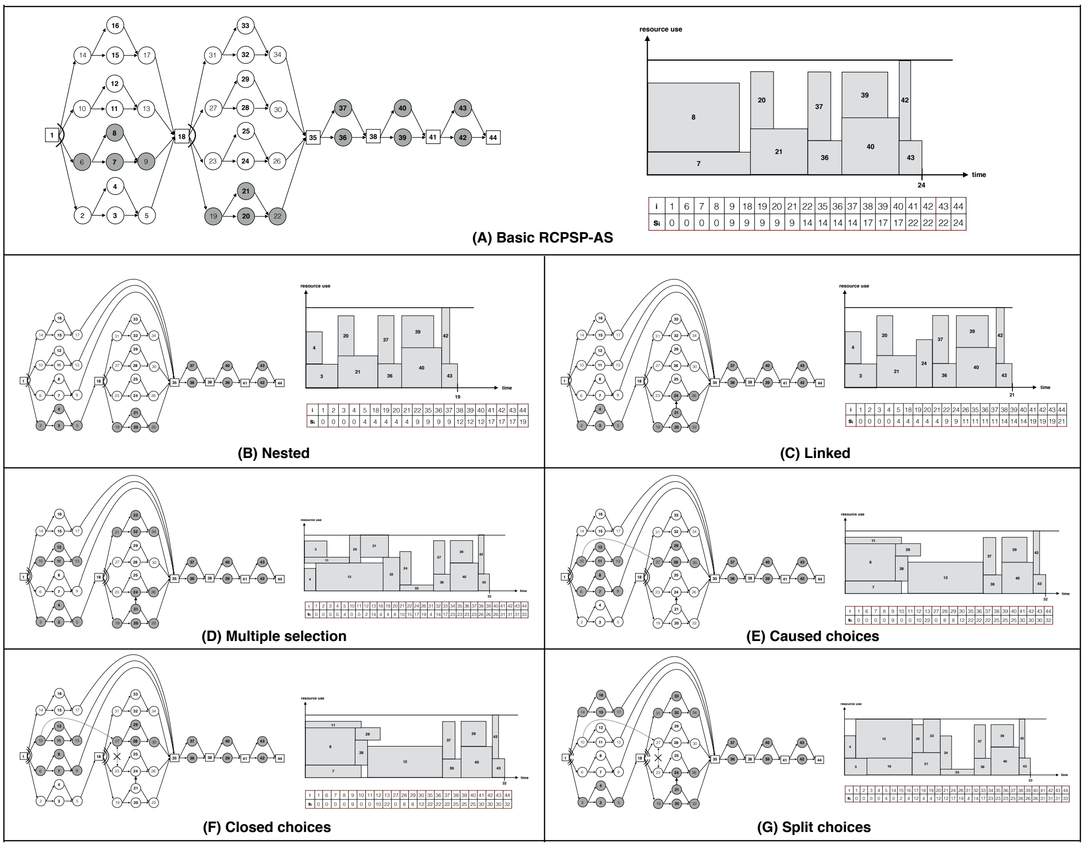

Figure 1. Illustrative project example with feasible project schedule for basic RCPSP-AS (A) and extensions (B-G)