BMIJ
ISSN: 2148-2586

bmij (2022) 10 (3):1095-1117
doi: https://doi.org/10.15295/bmij.v10i3.2111
Research Article

Dynamic multi-mode resource-constrained multi-project scheduling problem with weighted earliness and tardiness: a real-life boutique furniture implementation[1]

Ağırlıklı erken ve geç bitirmeli dinamik çok modlu kaynak kısıtlı çoklu proje çizelgeleme problemi: bir gerçek hayat butik mobilya uygulaması

Murat Ruhlusaraç[2]
Filiz Çalışkan[3]

[1] This study is derived from the first author's PhD thesis.

[2] Dr., Erciyes University, Kayseri, Turkiye, mruhlusarac@erciyes.edu.tr

ORCID: 0000-0003-1092-8331

[3] Prof. Dr., Erciyes University, Kayseri, Turkiye, caliskan@erciyes.edu.tr

ORCID: 0000-0002-7079-7559

Corresponding Author:

Murat Ruhlusaraç,

Erciyes University, Kayseri, Turkiye, mruhlusarac@erciyes.edu.tr

Submitted: 1/08/2022

1st Revised: 20/09/2022

Accepted: 23/09/2022

Online Published: 25/09/2022

Citation: Ruhlusaraç, M., & Çalışkan, F., Dynamic multi-mode resource-constrained multi-project scheduling problem with weighted earliness and tardiness: a real-life boutique furniture implementation, bmij (2022) 10 (3): 1095-1117, doi: https://doi.org/10.15295/bmij.v10i3.2111

Abstract

Real-life project scheduling environments are often dynamic and subject to disruption. Early or late completion of interrupted projects can create costs for the business. At the same time, there is an alternative to producing multiple projects at multiple different costs. In this study, a new mixed integer linear programming model that minimizes the sum of weighted earliness and tardiness penalties and mode selection costs is proposed to solve the real-life problem faced by a boutique furniture company. A proposed dynamic model also considers the cost of deviation from the baseline schedule in case disruption scenarios corrupt the resulting baseline schedule. The problems are solved with the CPLEX solver using the GAMS program. The results show that the interruption scenarios partially change the baseline schedule and increase the total cost. In case of more than one interruption in the same schedule, the number of late completed activities and their delay times increased.

Keywords: Project Scheduling, Dynamic Project, Multi-Mode, Multi-Project, Earliness and Tardiness

Jel Codes: M11, C61, O22

Öz

Gerçek hayat proje çizelgeleme ortamları genelde dinamiktir ve kesintiye uğrama olasılığı vardır. Kesintiye uğrayan projelerin erken bitirilme yada geç bitirilme durumları işletmeye maliyet oluşturabilir. Aynı zamanda birden çok projeyi birden çok farklı maliyetle üretme alternatifi vardır. Bu çalışmada butik bir mobilya firmasının karşılaştığı gerçek hayat problemini çözmek için ağırlıklı erken ve geç bitirme cezaları ile mod seçim maliyetleri toplamını en aza indiren yeni bir karma tamsayılı doğrusal programlama modeli önerilmektedir. Elde edilen temel çizelgenin kesinti senaryolarıyla bozulması durumunda temel çizelgeden sapmanın maliyetini de hesaba katan dinamik bir model önerilmiştir. Problemler GAMS programı kullanılarak CPLEX çözücü ile çözülmüştür. Elde edilen sonuçlar kesinti senaryolarının temel çizelgeyi kısmen değiştirdiğini ve toplam maliyeti artırdığını göstermiştir. Aynı çizelgede birden fazla kesinti olması durumunda ise geç tamamlanan faaliyetlerin sayısı ve gecikme süreleri artmıştır.

Anahtar Kelimeler: Proje Çizelgeleme, Dinamik Proje, Çoklu Mod, Çoklu Proje, Erken ve Geç Bitirme

JEL Kodları: M11, C61, O22

Introduction

One of the crucial issues researchers have been intensively working on is project scheduling. Project scheduling is to prepare time and resource usage plans that indicate when a project will be completed, which activities will be done, and when and how resources will be assigned to the activities. Unlike job scheduling, project scheduling also includes information on when activities start and end. The activities to be scheduled have precedence relations due to technological constraints or purely administrative reasons. Precedence relationships set the start-finish relationship between activities in the time horizon. While some activities are executed parallel due to time savings and technological availability, activity cannot generally be carried out until its previous activities have been completed (Ulusoy and Hazır, 2021). Assuming that the projects' data come to the execution environment is determined deterministically and that no changes are made to the schedule during execution, the project scheduling problems are called static project scheduling problems. Dynamic project scheduling problems are mentioned if there is a real-time change in deterministic data in the environment or if entirely new projects or activities come to the environment.

Project managers often deal with multiple projects simultaneously in real life and allocate scarce resources in a multi-project environment. Executing multiple projects together is a complex decision-making problem (Gonçalves, Mendes and Resende, 2015), and the multi-project scheduling problem is of the NP-hard type. Different modes can be defined for each activity with a different duration and resource requirement for each mode in some problems. There is a relationship between the processing time of an activity and the resources used. In general, it is expected that the cost will increase, and the activity period will decrease by using more resources per unit of time. Thus, project scheduling problems with one or more activities having more than one mode are called multi-mode resource-constrained project scheduling problems (Ulusoy and Hazır, 2021).

Our problem arises when a boutique furniture company that produces project-type products wants to deliver them to its customers immediately and produce just in time by keeping the baseline schedule they developed. Earliness and tardiness penalties were added to the problem to comply with the due dates of project activities. In addition, more than one mode was defined to carry out some activities in the problem.

It is assumed that some projects have already arrived in the project scheduling environment. Each of these projects has many activities and has priorities among these activities themselves. Some activities can be carried out in multiple modes, with different processing times and resource usage needs. There are renewable resources in the production environment. These resources can re-task after use, and they do not run out as they are used. The total amount of these resources used over time is limited and known in advance. Due dates for each activity are known from the beginning. If the activities are completed early, there is a penalty for earliness due to stock holding.

Conversely, if the activities are completed late, there is a penalty for the delay due to opportunity cost or customer loss. In addition, a mode selection cost must be incurred to produce in each mode. Therefore, the static model aims to find the best baseline schedule by satisfying the constraints and resource limitations while ensuring minimum total project execution cost. The total project execution cost includes weighted earliness, tardiness, and mode selection costs.

The nature of the real-life problem adding some current data entries by producing various scenarios, makes this static problem a dynamic problem. The dynamic model is applied when a disruption occurs in the executed baseline schedule. The dynamic problem minimizes the total cost while minimizing deviations in the reactive and baseline finish times, not getting too far from the baseline schedule. Deblaere, Demeulemeester and Herroelen (2011) used a similar approach in their study and explained why it is essential that the repaired schedule is close to the original baseline schedule. Accordingly, it may be necessary to change contracts with subcontractors by deviating slightly from the baseline schedule, and undesired effects, such as dealing with accumulated stock costs and employee complaints, may occur. Moreover, suppose scarce resources need to be reserved ahead of time. In that case, any schedule disruption can lead to too much delay in activities that require scarce resources, ultimately leading to a breach of the project deadline.

This paper presents a multi-mode resource-constrained multi-project scheduling problem with weighted earliness and tardiness (MRCMPSP-WET). Our real-life multi-project scheduling problem has earliness and tardiness penalties to force the system to produce just-in-time. Furthermore, multimodal activities enable flexible production planning thanks to the different resource-time combinations of activities according to the execution situation. We propose a mixed-integer linear programming model for this problem. Then, we derive the dynamic version for real-life disruption scenarios. We develop a predictive-reactive scheduling procedure including these models and solve the problem using optimization software.

Our paper is organized as follows: Section 2 is the section in which the relevant literature studies are examined. Section 3 introduces uncertainty and dynamism in the project scheduling environment and measures dynamism with the ratios. A static model for the baseline schedule and a dynamic model for the reactive schedule: a predictive-reactive scheduling procedure and two mathematical formulations are presented in Section 4. Section 5 describes our real-life implementation and disruption scenarios. Section 6 gives experimental results, and Section 7 presents the conclusion.

Literature review

According to solution methods, literature studies on resource-constrained project scheduling can be classified as exact and approximate. The optimal solution can be found in small-size problems with exact methods (Bozorgirad and Logendran, 2012).

Schwindt (2000) considered the scheduling of subprojects, including earliness and tardiness costs, and proposed a simple and dual algorithm and a branch-and-bound algorithm. Vanhoucke, Demeulemeester and Herroelen (2001) studied the resource-constrained project scheduling problem with weighted earliness and tardiness (RCPSP-WET), which is assumed to have known deterministic activity data and constant renewable resource requirements. They introduced a depth-first branch-and-bound algorithm and a recursive search algorithm. Kéri and Kis (2006) described a new method for the RCPSP-WET, a branch-and-bound method, but it computed the lower bounds differently and added time windows. Afshar-Nadjafi and Shadrokh (2008) minimized the earliness and tardiness costs of the project scheduling problem with an unconstrained resource. Cheng, Fowler, Kempf and Mason (2015) studied the difference between activity splitting and pre-emption on RCPSPs and showed multi-mode RCPSPs with non-pre-emptive activity splitting. Then, they modified a precedence tree-based branch-and-bound algorithm to find optimal solutions to their generated problem sets. Watermeyer and Zimmermann (2020) used the branch-and-bound algorithm for RCPSP with general temporal constraints and partially renewable resources.

Zhu, Bard and Yu (2005) focused on the disrupted RCPSP and solved a hybrid mixed-integer programming method with CPLEX. Damay, Quilliot and Sanlaville (2007) studied the RCPSP with activities that may or may not be pre-emptive, solved it using a linear programming model, and derived an algorithm based on neighbourhood search. Chtourou and Haouari (2008) addressed a two-stage approach for solving RCPSP to minimize the project duration and maximize the quality-robustness. Hartmann and Briskorn (2010) summarized and classified publications of the well-known RCPSP variants. The most popular extensions were multiple modes, time lags, and objectives based on the net present value. Koné, Artigues, Lopez and Mongeau (2011) presented three discrete and continuous-time mixed-integer linear programming models for RCPSPs based on the Start/End and the On/Off formulations. Kreter, Rieck and Zimmermann (2016) studied the RCPSP with general temporal constraints and presented three binary linear models that use start-based changeover-based or execution-based binary decision variables.

Using LP, Voß and Witt (2007) examined a multi-mode resource-constrained multi-project scheduling problem (MRCMPSP) on real-life problems. Krüger and Scholl (2009) used integer LP for the resource-constrained multi-project scheduling problem with transfer times (RCMPSP-TT). Pamay (2011) also used an LP, a generalization of the shifting-bottleneck approach from the machine scheduling literature for the resource-constrained multi-project scheduling problem with weighted earliness and tardiness (RCMPSP-WET). Chakrabortty, Sarker and Essam (2016) examined the multi-mode RCPSP (MRCPSP) to solve the schedule breakdown scenario using discrete time-based LP. Oztemel and Selam (2017) investigated a real-world multi-mode resource-constrained mould project scheduling problem and proposed a bee-based mould scheduling model to solve problems of this nature. Tao and Dong (2018) proposed alternative project structures to consider MRCPSP and generated a bi-objective linear integer model to minimize the makespan and the total cost. Changchun, Xi, Canrong, Qiang, and Li (2018) presented an LP model for vehicle scheduling based on MRCPSP, and Burgelman and Vanhoucke (2018) studied three mathematical models for MRCPSP maximizing weighted alternative execution modes. Ruhlusaraç and Çalışkan (2018) studied the MRCMPSP-WET using a small-size example which had three projects with 4-activity 2-modes and proposed a model to minimize the total cost. Aouam and Vanhoucke (2019) dealt with MRCPSP containing agent perspective and developed two-LP models. Kurt and Kececi (2019) examined RCMPSP, which was encountered in a software company using LP. Wang, Lappas and Gounaris (2019) proposed seven discrete-time models for generalized MRCPSP with real-life problems. Hauder, Beham, Raggl, Parragh and Affenzeller (2020) studied RCMPSP with activity chains and time flexibility and developed mixed-integer and constraint programming models maximizing time and resource balance. Bold and Goerigk (2021) considered uncertain activity durations on RCPSP and proposed a robust two-stage approach. Yuan, Ye, Lin and Gen (2021) considered prefabricated building construction RCPSP with multi-objective multi-mode in an uncertain environment and formulated a model with four objectives. Then, they proposed a hybrid cooperative co-evolution algorithm to achieve robust project scheduling, reducing the uncertainty of the execution.

Researchers also used proactive and reactive methods in the dynamic project scheduling literature. However, problem-specific heuristics or other dynamic approaches were proposed. Van de Vonder (2006) developed the myopic activity-based optimization (MABO) heuristic, a proactive procedure to solve real-life construction projects. Van de Vonder, Ballestín, Demeulemeester and Herroelen (2007a) used new heuristic reactive project scheduling procedures to minimize schedule deviations. Van de Vonder, Demeulemeester and Herroelen (2007b) used a predictive-reactive procedure to maximize schedule stability and the possibility of completing projects on time. Da Silva, Ochi and Santos (2008) studied dynamic RCPSP (DRCPSP) and introduced a new evolutionary algorithm to solve it. Van de Vonder, Demeulemeester and Herroelen (2008) presented a heuristic and meta-heuristic and used simulation-based analysis to perform all algorithms. Da Silva and Ochi (2010) proposed two-hybrid heuristics combining an evolutionary algorithm with an exact solution approach. Capa and Ulusoy (2015) studied a real-life DRCPSP from a home appliance company and used a reactive method, including a multi-objective genetic algorithm. Salemi Parizi, Gocgun and Ghate (2017) provided a simulation-based method for DRCPSPs. Davari and Demeulemeester (2017) and (2018) described proactive-reactive (PR) policies. For solving RCPSP under uncertainty, Song, Xi, Kang, and Zhang (2018) presented an agent-based simulation system and PR algorithms; Ma, Demeulemeester, He and Wang (2019) discussed a proactive method that generated robust schedules, and W. Wang, Ge, Li and Su (2019) explained a proactive-reactive multi-project scheduling model using genetic algorithm. Joo, Chua, Cai and Chua (2019) presented a coordination-based reactive MRCPSP for construction projects. Limon and Krishnamurthy (2020) considered the scheduling problem for dynamic biomanufacturing projects, including multi-tasking and no-wait constraints. They formulated a mixed-integer linear programming (MILP) model that minimized total tardiness and proposed a dynamic approach to solve modified MILP models to revise the schedule for new project arrivals, workload balancing, and precedence changes. Ruhlusaraç and Çalışkan (2020) proposed a mathematical model for the dynamic project scheduling problem and its reactive scheduling implementations. Chakrabortty, Rahman and Ryan (2020) reviewed the RCPSP under dynamic environments with stochastic activity durations, unstable resource availability, and time-varying resource demands. They solved the dynamic problem using a variable neighbourhood search-based local search algorithm. Rahman, Chakrabortty and Ryan (2021) studied an RCPSP under dynamic environments with activity durations, resource availabilities, and resource requests that vary with time. They proposed a mathematical model for this problem and then developed a genetic algorithm-based memetic algorithm. Chakrabortty, Rahman, Haque, Paul and Ryan (2021) considered an RCPSP, including real-world disruption scenarios, and proposed a mathematical model for a reactive scheduling method called an event-based reactive approach. He, Zhang and Yuce (2022) proposed a genetic and ant colony optimization algorithm for resource planning and multi-project scheduling. Saif, Yue and Awadh (2022) considered the arrival of new projects in the planning horizons and proposed a drum buffer rope heuristic for scheduling multiple projects. Zhang, Hu, Cao, and Wu (2022) dealt with multi-mode multi-project inverse scheduling problems and proposed a modified particle swarm optimization algorithm combined with tabu search.

Table 1 gives the classification of papers in the literature reviewed in this paper. Regarding Table 1, dynamic methods and metaheuristics have recently been proposed for project scheduling.

The literature has some gap that there are not enough dynamic multi-mode multi-project papers under uncertainty. This study differs from Pamay's (2011) study by its multi-modality, adding selection costs of the modes to the objective function and providing dynamism by using the finish time in the baseline schedule. In addition, the proposal of adding dynamic project arrivals to the problem stated by Yassine, Mostafa and Browning (2017) was evaluated. Furthermore, new mathematical models were introduced to the literature, inspired by Hazır and Ulusoy's (2020) ideas stating that the number of multi-mode scheduling studies under uncertainty is limited.

The contributions of this paper are as follows. We present a new mathematical model for the multi-mode resource-constrained multi-project scheduling problem with weighted earliness and tardiness (MRCMPSP-WET) and derive a new dynamic model for the problem. To our knowledge, it is the first formulation for the problem, including the mode selection costs in the objective function and solving it quickly using standard optimization software. We propose a predictive-reactive scheduling procedure to deal with disruption scenarios. The proposed procedure can be easily adapted to real-life project scheduling problems. Computational experiments confirm that our dynamic model can give optimal results for one 34-activity and three 24-activity projects with two modes.

Table 1: The Classification of Papers in Literature Review

Methods / Papers	Exact Methods		Approximate Methods	
	Branch-and-bound algorithm	Linear Programming	Heuristics and Metaheuristics	Dynamic Programming, Algorithm, Methods
Schwindt (2000)	√			
Vanhoucke et al. (2001)	√			
Zhu et al. (2005)		√		
Kéri and Kis (2006)	√			
Van de Vonder (2006),(2007a),(2007b),(2008)			√ √ √ √	
Damay et al. (2007)		√		
Voß and Witt (2007)		√		
Afshar-Nadjafi and Shadrokh (2008)	√			
Da Silva et al. (2008)			√	
Krüger and Scholl (2009)		√		
Koné et al. (2011)		√		
Pamay (2011)		√		
Capa and Ulusoy (2015)			√	
Cheng et al. (2015)	√			
Chakrabortty et al. (2016)		√		
Kreter et al. (2016)		√		
Davari and Demeulemeester (2017),(2018)			√ √	
Salemi Parizi et al. (2017)			√	
Burgelman and Vanhoucke (2018)		√		
Changchun et al. (2018)		√		
Ruhlusaraç and Çalışkan (2018)		√		
Song et al. (2018)			√	
Tao and Dong (2018)		√		
Aouam and Vanhoucke (2019)		√		
Kurt and Kececi (2019)		√		
W.Wang et al. (2019)		√	√	
Chakrabortty et al. (2020)				√
Hauder et al. (2020)		√		
Limon and Krishnamurthy (2020)		√	√	√
Ruhlusaraç and Çalışkan (2020)		√		√
Watermeyer and Zimmermann (2020)	√			
Chakrabortty et al. (2021)		√	√	√
Rahman et al. (2021)		√	√	√
Yuan et al. (2021)		√		
He et al. (2022)			√	
Saif et al. (2022)			√	
Zhang et al. (2022)			√	

Uncertainty and dynamism in the project scheduling environment

The activity of the project may end earlier or later than planned. Resources may be unavailable for a while, new activities may have to be added to the project, or activities may be interrupted for a while. It is essential for every project manager to effectively deal with these uncertainties and produce solutions (Van de Vonder et al., 2007a).

Dealing with uncertainty in a scheduling environment in which the generation structure of the precedence network is deterministic, Herroelen and Leus (2005) distinguished between five methods: reactive scheduling, stochastic scheduling, fuzzy scheduling, proactive scheduling, and sensitivity analysis.

Most project scheduling research has focused on studies in recent years where a project may experience significant disruptions during execution. The proactive-reactive project scheduling approaches use a reactive method when a disruption occurs in the schedule during project execution. Combined with a proactive scheduling method, this method attempts to deal with disruptions caused by a breakdown to produce baseline schedules expected to be sturdy (Van de Vonder et al., 2007b).

Uncertainty can be classified into two categories, internally and externally (Hazır and Ulusoy, 2020). Internal uncertainties relate to systems and resources directly associated with the project, and mainly, the firm can control them. However, many other uncertainties are due to external factors of the project and cannot be controlled by the firm. When and how to react to real-time disruptions have been defined as two policies. First, it is stated when the system revision decisions are made. There are three alternatives for this (Sabuncuoglu and Bayiz, 2000): periodic, event-based, and mixed. After answering the when question, two types of rescheduling approaches are mentioned in the dynamic scheduling literature; schedule repair and complete rescheduling.

Dynamic scheduling, also known as online scheduling, includes real-time disruptions. With dynamic scheduling, the process of minimizing the negative impact of unexpected real-time events, analysing the current situation, and changing the schedule to reduce interruptions is carried out. The execution of the schedule is considered stepwise. First, priorities are determined and then applied dynamically.

The fact that the environment in which the projects will be carried out is static or dynamic is crucial as it can significantly affect the decision-making and solution process of the project managers affected by the environment. When comparing static and dynamic environments, the main difference is that static environments are easily predictable, and dynamic environments are challenging to predict. Static environments are more rigid, formal, and slow, while dynamic environments are more flexible, informal, and fast. While controlling with detailed plans in static environments, a framework plan is applied in dynamic environments and is open to adaptation. The top management makes decisions in static environments, and the team makes decisions with the vision of the experts in dynamic environments (Collyer, 2015).

Aytug, Lawley, McKay, Mohan and Uzsoy (2005) classified scheduling issues with executional uncertainties into reactive, robust, and predictive-reactive scheduling. Our real-life problem is within the scope of the third category.

In predictive-reactive scheduling, a schedule is generated and rescheduled in response to the disruption in real-time. This method uses a mixture of the baseline schedule and reactive procedures. First, an estimated schedule determining the planned start and completion times of the operations of the works is produced. Then this schedule is updated in response to unexpected disruptions.

The degree of dynamism

Real-life problems are naturally in a dynamic environment, and several changes are likely to occur that could affect the execution of the problem. Therefore, there must be only one measure that can define the system's dynamism for someone who wants to evaluate the performance of a particular algorithm under changing conditions. However, a single measure will not be sufficient to determine the dynamism of all systems. For this reason, Larsen and Madsen (2000) and Larsen, Madsen and Solomon (2007) mentioned two different measurement types with a time window and without a time window to measure the degree of dynamism.

Without time windows, it is possible to mention three parameters in a system. These are the number of static/current projects, dynamic projects, arrival times, or breakdowns. The degree of dynamism (DD) ratio can be defined as below:

$$DD = \frac{Dynamic\ Requests}{Total\ Requests}$$

The effective degree of dynamism

To describe the Effective Degree of Dynamism (EDD), let us consider a scenario where we receive some projects at 0. This time horizon will end at time K and is a dynamic situation at the time i; for example, when a new project comes. We show this period with k_i ($0 < k_i \le K$). If we used $p_{dynamic}$ for new projects or requests that come during the execution (dynamic requests), p_{total} is obtained by adding the existing requests to these demands (total requests).

$$EDD = \frac{\sum_{i=1}^{p_{dynamic}} \left(\frac{k_i}{K}\right)}{p_{total}}$$

EDD represents an average of how late projects or requests are coming, proportioned to K, which is the latest time projects can arrive. It is clear from the formula that it is in the range of 0 ≤ EDD ≤ 1. If the ratio, EDD = 0, there is a pure dynamic system; if EDD = 1, there is a purely static system.

The effective degree of dynamism with time windows

Reaction time is a crucial issue in applications with time windows. When a dynamic project or request comes at a time k_i, we can show the earliest possible response time by e_i and the latest response time by l_i. If the reaction time is r_i, it can be calculated with the formula $r_i = l_i - k_i$.

In calculating the Effective Degree of Dynamism with Time Windows (EDD-TW), the relationship between the reaction time and the planning horizon's rest is the main component because planners prefer dynamic projects with relatively longer reaction times.

$$EDD-TW = \frac{1}{p_{total}}\sum_{i=1}^{p_{total}}\left(\frac{K-(l_i-k_i)}{K}\right) = \frac{1}{p_{total}}\sum_{i=1}^{p_{total}}\left(1-\frac{r_i}{K}\right)$$

According to the formula, the dynamic event may have a reaction time (r_i) during the time horizon K and the disruption that occurs while the schedule continues to be executed. These reaction times are subtracted from time K for each breakdown and divided by K, and the total sum of these sections is divided by the total number of projects (p_{total}). It is also clear from the formula that it is in the range of 0 ≤ EDD-TW ≤ 1.

The proposed mathematical models and procedure

This section presents a mixed-integer linear programming model for the multi-mode resource-constrained multi-project scheduling problem with weighted earliness and tardiness (MRCMPSP-WET). First, we generate the baseline schedule with this MILP model and then reschedule with the derived dynamic model when a dynamic situation occurs.

Static model for the baseline schedule

In the static model, K (minute) denotes the time horizon of the problem. I gives the complete set of projects in the baseline schedule. J_i represents the set of activities for the project i. M_i represents the mode set of activity j. R denotes the set of renewable resources. H (minute) is a second period with elements of the set K. When a is the predecessor activity and element of the J_i set, b is the successor activity and element of the J_i set. Other indices are i project index, j activity index, m mode index, r resource index, k general time index, and h custom time index.

Our problem involves three types of costs: c_r (₺/hour) unit usage cost of resource *r*, e_{ij} earliness penalty per unit of time and t_{ij} tardiness penalty per unit of time. There are two parameters in the problem that include multiple modes: p_{ijm} (minutes) the processing time of activity *j* of project *i* in *m* mode and q_{ijmr} (number) the *r* type renewable resource requirement of activity *j* of project *i* in unit time in *m* mode. In addition, we know these two parameters beforehand: d_{ij} (minute) the due dates of activity *j* of project *i* and Q_{rk} the available amount of *r* type renewable resource at time *k*.

The outstanding feature of our static model is that the multi-mode feature is added to the model, and the production costs incurred in different modes are added to the objective function. In addition, how the earliness and tardiness penalty costs are calculated at the constraints is also new. We use two binary decision variables and three nonnegative integer variables for the model formulation. The variables are defined as follows.

X_{ijmk}: Binary variable equal to 1 if activity *j* of project *i* starts in *m* mode at *k* time, and 0 otherwise;

Y_{ijm}: Binary variable equal to 1 if activity *j* of project *i* is executed in *m* mode, and 0 otherwise;

f_{ij}: Integer variable specifying the finish time of activity *j* of project *i;*

E_{ij}: Integer variable indicating the unit earliness of activity *j* of project *i;*

T_{ij}: Integer variable indicating the unit delay of activity *j* of project *i.*

Given this notation, the static model (MRCMPSP-WET) can be modelled as follows.

$$Min\ Z = \sum_{i\in I}\sum_{j\in J_i}\left(e_{ij}E_{ij}+t_{ij}T_{ij}\right)+\sum_{i\in I}\sum_{j\in J_i}\sum_{m\in M_i}\left(\sum_{r\in R}q_{ijmr}c_r\frac{p_{ijm}}{60}\right)Y_{ijm} \qquad (1)$$

Constraints:

$$\sum_{m\in M_i}\sum_{k\in K}X_{ijmk}=1 \qquad \forall i\in I,\forall j\in J_i \qquad (2)$$

$$f_{ij}=\sum_{m\in M_i}\sum_{k\in K}k\,X_{ijmk}+\sum_{m\in M_i}p_{ijm}\,Y_{ijm} \qquad \forall i\in I,\forall j\in J_i \qquad (3)$$

$$f_{ib}-f_{ia}\geq p_{ibm}\,Y_{ibm} \qquad \forall i\in I,\forall(a,b)\in J_i,\forall m\in M_j \qquad (4)$$

$$f_{ij}-T_{ij}+E_{ij}\ =d_{ij} \qquad \forall i\in I,\forall j\in J_i \qquad (5)$$

$$\sum_{m\in M_j}Y_{ijm}=1 \qquad \forall i\in I,\forall j\in J_i \qquad (6)$$

$$\sum_{i\in I}\sum_{j\in J_i}\sum_{m\in M_j}\sum_{h=max\{1,k-p_{ijm}+1\}}^{k} X_{ijmh}\, q_{ijmr} \le Q_{rk} \qquad \forall r \in R, \forall k \in K \tag{7}$$

$$\sum_{k\in K} X_{ijmk} \le Y_{ijm} \qquad \forall i \in I, \forall j \in J_i, \forall m \in M_j \tag{8}$$

$$X_{ijmk} \in \{0,1\}, Y_{ijm} \in \{0,1\}, f_{ij}, E_{ij}, T_{ij} \ge 0 \tag{9}$$

The objective function Z (1) aims to minimize total tardiness and earliness penalty costs and total mode selection costs to find a baseline schedule in the static model. (p_{ijm} / 60) is required for minute-hour conversion, i.e., c_r. Constraint (2) ensures that each project activity is done in one of the modes and only once. Constraint (3) gives the finish time of each project activity. Constraint (4) ensures precedence relationships by maintaining the balance between finish times. Constraint (5) calculates the unit's early and tardy finish time for each project activity. Constraint (6) ensures that each project activity takes place in only one mode. Constraint (7) gives limits on renewable resource constraints. Finally, constraint (8) provides the relationship between X_{ijmk} and Y_{ijm} binary variables.

A dynamic model for the reactive schedule

It is assumed that "activities that continue to be executed in the disruption are executed on the reactive schedule with remaining processing times." Any completed activities remain and have the same solution as the baseline schedule. Ongoing activities continue, but only if the resource is available. Awaiting activities are then rescheduled. Other assumptions are the same as in the static model. In the model of the dynamic problem, sets and indices are updated, and updated symbols are used as overlined.

The current version is used for some of the static model parameters in the dynamic model parameters. These are $\overline{Q_{rk}}$, $\overline{p_{ijm}}$, $\overline{q_{ijmr}}$, $\overline{e_{ij}}$, $\overline{t_{ij}}$, $\overline{d_{ij}}$, $\overline{c_r}$. In addition, the static model variable f_{ij} is used in the dynamic model as a parameter. β is included in the objective function as a coefficient between 1 and 1000. β can be increased or decreased depending on how important the cost of deviations from the baseline schedule is to us and how much we want it to affect the objective function in the model. $\overrightarrow{\overline{X_{ijmk}}}$ parameter informs the activities of which period and in which mode the activities continue to be executed when the dynamic situation occurs.

The prominent feature of our dynamic model is that the multi-mode feature uses the baseline schedule's finish time data and compares it with the current schedule's finish time. Then transforming, this negative situation into a cost with the desired beta coefficient and adding it to the objective function. In addition, the dynamic model uses up-to-date disruption data and gains more specificity than the static model with special adjustments to the constraints. We use two binary and seven nonnegative integer variables for the dynamic model. The variables are defined below.

$X_{\overline{ijmk}}$: Binary variable equal to 1 if activity *j* of project *i* starts in *m* mode at *k* time in the reactive schedule, and 0 otherwise;

$Y_{\overline{ijm}}$: Binary variable equal to 1 if activity *j* of project *i* is executed in *m* mode in the reactive schedule, and 0 otherwise;

$\overline{f_{ij}}$: Integer variable specifying the current finish time of activity *j* of project *i*;

$\overline{E_{ij}}$: Integer variable indicating the current unit earliness of activity *j* of project *i*;

$\overline{T_{ij}}$: Integer variable indicating the current unit tardiness of activity *j* of project *i*;

CF_{ij}: Integer slack variable for the deviations from the current finish times;

BF_{ij}: Integer slack variable for the deviations from the baseline finish times;

TC: Integer variable calculating the total cost of the problem (₺);

FT: Integer variable calculating the finish time differences (min.).

The dynamic model (DMRCMPSP-WET) is presented below.

$$Min\ Z' = \ TC + \beta\ FT \tag{10}$$

Constraints:

$$TC = \sum_{i\in I}\sum_{j\in J_i}\left(\overline{e_{ij}}\overline{E_{ij}} + \overline{t_{ij}}\overline{T_{ij}}\right) + \sum_{i\in I}\sum_{j\in J_i}\sum_{m\in M_j}\left(\sum_{r\in R}\overline{q_{ijmr}}\overline{c_r}\frac{\overline{p_{ijm}}}{60}\right)Y_{\overline{ijm}} \tag{11}$$

$$FT = \sum_{i\in I}\sum_{j\in J_i}\left(CF_{ij} + BF_{ij}\right) \tag{12}$$

$$\sum_{m\in\overline{M_j}}\sum_{k\in\overline{K}} X_{\overline{ijmk}} = 1 \qquad \forall i \in \overline{I}, \forall j \in \overline{J_i} \tag{13}$$

$$\overline{f_{ij}} = \sum_{m\in\overline{M_j}}\sum_{k\in\overline{K}} k\ X_{\overline{ijmk}} + \sum_{m\in\overline{M_j}} \overline{p_{ijm}}\ Y_{\overline{ijm}} \qquad \forall i \in \overline{I}, \forall j \in \overline{J_i} \tag{14}$$

$$\overline{f_{ib}} - \overline{f_{ia}} \geq \overline{p_{ijm}}\ Y_{\overline{ijm}} \qquad \forall i \in \overline{I}, \forall (a,b) \in \overline{J_i}, \forall \mathrm{m} \in \overline{M_j} \tag{15}$$

$$\overline{f_{ij}} - \overline{T_{ij}} + \overline{E_{ij}} = \ \overline{d_{ij}} \qquad \forall i \in \overline{I}, \forall j \in \overline{J_i} \tag{16}$$

$$\sum_{m\in\overline{M_j}} Y_{\overline{ijm}} = 1 \qquad \forall i \in \overline{I}, \forall j \in \overline{J_i} \tag{17}$$

$$\sum_{i\in I}\sum_{j\in J_i}\sum_{m\in\overline{M_j}}\sum_{\overline{h}=max\{1,k-\overline{p_{ijm}}+1\}}^{k} X_{\overline{ijmh}}\ \overline{q_{ijmr}} \leq \ \overline{Q_{rk}} \qquad \forall r \in \overline{R}, \forall k \in \overline{\mathrm{K}} \tag{18}$$

$$\sum_{k\in\overline{K}} X_{\overline{ijmk}} \ \leq \ Y_{\overline{ijm}} \qquad \forall i \in \overline{I}, \forall j \in \overline{J_i}, \forall \mathrm{m} \in \overline{M_j} \tag{19}$$

$$\overrightarrow{X_{ijmk}} = 1 \qquad \forall i \in \vec{I}, \forall j \in \vec{J}, \forall \mathrm{m} \in \vec{M}, \forall \mathrm{k} \in \vec{K} \tag{20}$$

$$\overline{f_{ij}} - CF_{ij} = f_{ij} - BF_{ij} \qquad \forall i \in \overline{I}, \forall j \in \overline{J_i} \tag{21}$$

$$X_{\overline{ijmk}} \in \{0,1\}, Y_{\overline{ijm}} \in \{0,1\}, \overline{f_{ij}}, \overline{T_{ij}}, \overline{E_{ij}}, CF_{ij}, BF_{ij}, TC, FT \geq 0 \tag{22}$$

The objective function Z' (10) minimizes the sum of the total cost variable and finish time difference variable multiplied by beta, a coefficient between 1 and 1000 used to indicate the importance of the variable. The total cost (TC) equation (11) calculates the total cost, including weighted earliness and tardiness penalty costs and total mode selection costs. The finish time (FT) equation (12) consists of how far we deviated from the baseline schedule by the sum of all the deviations from the current finish times (CF) and the finish times in the baseline schedule (BF). As in constraint (2), constraint (13) ensures that each project activity is done in one of the modes and only once. The only difference is that it contains up-to-date project information. Constraint (14) gives the current finish time of each project activity. With binary Y variable, the equation calculates the current finish time of each project's activity by using the information in which mode the project is carried out and when it is executed. We assumed zero-lag finish-start precedence relations, and constraint (15) maintains the current precedence relations by ensuring the balance between the finish times. Constraint (16) calculates each project activity's current early and tardy finish times using finish times and due dates. Constraint (17) ensures that each current project activity takes place in only one mode. Resource capacity constraint (18) bounds using resources on a reactive schedule at the set of $\overline{R}$. Constraint (19) provides the relationship between the binary variables X and Y. Constraint (20) sets the X variable of the activities carried out when a disruption occurs in the baseline schedule as 1. Constraint (21) calculates all the deviations from the current finish times (CF) and the finish times in the baseline schedule (BF). Furthermore, the decision variables require given conditions (22).

The proposed predictive-reactive scheduling procedure

The dynamic problem here is a two-step reactive process; In the first stage, the static problem is solved, and the baseline schedule is obtained. In the second stage, the dynamic problem is solved by rescheduling, using some necessary data from the static problem, and updating some data to achieve the reactive goal. The method gains originality thanks to the innovations of linear programming models, which are given as static and dynamic models used in these stages.

Chakrabortty, Sarker and Essam (2018) proposed three algorithms against disruption and a series of independent disruption cases for the prepared baseline schedule. We were inspired by these approaches and adapted a reactive rescheduling procedure to the problem. We proposed a predictive-reactive scheduling procedure with eight steps to solve our problem. These steps are as follows.

Procedure: Multi-disruption under the pre-empt resume condition

Input: Sample problem data and the parameters: number of projects, number of activities, number of modes, number of resources, number of resources available, processing times of each activity in the modes, mode selection costs, unit earliness and tardiness penalty costs of activities, due dates of activities, precedence relations between activities, resource needs of activities, and periods.

Output: Optimal objective function value Z' or infeasible.

Step 1: Set $Z' = \{ TC + \beta\ FT \}$. Set the TC function using the earliness and tardiness costs of the activities and using the mode selection costs of the activities. Set the FT equation by taking the sum of the new finish time deviation variable of each activity and the finish time deviation variables in the baseline schedule.

Step 2: Solve the MILP model by adding resources and other constraints to an optimization software.

Step 3: Evaluate the initial baseline schedule based on the obtained solution results and record for reactive rescheduling the values of the variables f_{ij} that make up the Z'-value (if any).

Step 4: Identify wholly and partly unfinished activities (if any) when any schedule disruption occurs. Use the given parameters and measure the resource usage values.

Step 5: Measure current resource capacities, consider the breakdown value of each resource according to the disruption time (if any), and consider the current processing times, taking into account the breakdown values of the activities' processing times (if any).

Step 6: Start over by inputting current data and parameters. Solve (reschedule) the LP model within the disruption time window (within the reactive response time) using optimization software to obtain the reactive schedule using the dynamic model.

Step 7: Evaluate the reactive schedule according to the obtained solution results and apply the current schedule according to the objective function that minimizes the sum of TC and FT values.

Step 8: If all of the current projects and activities are completed, start from Step 1 when the new project arrives; Otherwise, if the schedule is broken while the project and activities are in process, continue with Step 4.

Real-life implementation

Project production costs of all modes, earliness and tardiness penalty costs, and due dates for all activities are computed for real-life implementation data. These formulas are given in the following subtitles.

Production costs

General production costs include depreciation of machines, electricity bill, annual rent, and repair and maintenance costs. The company operates 2340 hours a year. Estimated loading rates and activity costs are calculated as (Civelek and Özkan, 2006):

$$Estimated\ Loading\ Rate\ (ELR) = \frac{Estimated\ Total\ General\ Production\ Costs\ (GPC)}{Estimated\ Total\ Direct\ Labor\ Hour\ (DLH)}$$

$$Activity\ Cost\ (AC) = (DLH\ Required\ for\ Activity\ x\ ELR)$$

Earliness penalty costs

Earliness includes extra storage costs, idle times, and indirect opportunity losses (Khoshjahan, Najafi and Afshar-Nadjafi, 2013). After completing each activity, the storage area will determine storage costs. Storage costs come to the fore in the earliness penalty costs, so warehouse rent is essential. In addition, in case of early finishing according to the company's planning, the machines may remain empty in this unexpected situation and cause them to work with idle capacity. We can add this to the earliness penalty in the modes, taking into account the average production cost of the modes (Ruhlusaraç, 2020).

$$Earliness\ Penalty\ Cost\ (EPC) = (Area\ Covered\ by\ the\ Activity\ x\ Warehouse\ Rent) + Average\ Mode\ Cost$$

Tardiness penalty costs

Calculating tardiness penalty cost is related to the tightness of the due date value. A project with a close due date is more likely to be delayed, but penalties per unit time are lower than a looser due date project with a larger time frame to complete the project (Pamay, 2011). The delay in completing the project or activities; causes customer complaints, loss of reputation and income, financial penalties, or goodwill damage. If the tardiness cost is too low, the delay time will usually be high. Van Peteghem and Vanhoucke (2014) defined unit tardiness cost as $\tau\%$ of the total resource unit costs. In addition, 100% of this total and even the number of M, a great value, is reached in the definition of high tardiness cost. In contrast, this definition decreases to 0% in the definition of low tardiness cost. Su, Santoro and Mendes (2018) took this value as an average of 25% of the resource's total unit costs, and these rates are determined as 87.5% in case of high cost and 12.5% in case of low cost. Furthermore, some researchers using test problems assigned random numbers between 1 and 10 (Kolisch, Sprecher and Drexl, 1995), (Afshar-Nadjafi and Shadrokh, 2008), (Pamay, 2011), 10 and 30 (Golestaneh, Jafari, Khalilzadeh and Karimi 2013), 10 and 20 (Afshar-Nadjafi, Basati and Maghsoudlou 2017) for tardiness unit penalty costs.

We can add an unfair occupation of the planned resources to the tardiness penalty because the same activity style could not be carried out during a delayed period, and production opportunities would be missed. Therefore, instead of randomly selecting and assigning specific numbers, customer satisfaction, reputation, or potential opportunity losses can be used to generate these tardiness penalty costs by taking twice the average resource cost of each activity in production modes. In this context, unit tardiness costs are defined by taking a certain percentage (200%) of the sum of the resource unit costs (Ruhlusaraç, 2020).

$$Tardiness\ Penalty\ Cost\ (TPC) = (Average\ Mode\ Cost\ x\ 200\%)$$

Due dates

Determining the due dates of activities is very important for our problem because determining due dates looser or tighter than they should be will significantly affect earliness and tardiness costs. Since the objective function minimizes these costs, it can also influence the choice between modes.

There are several methods known in the literature for calculating the due date. Vanhoucke et al. (2001) suggested one of them. A maximum due date is first created by multiplying each project's critical path length and the 1.5 coefficient. Random numbers are then generated between 1 and the maximum due date. These last generated numbers are assigned to ascending order activities. Another method we use in the problem is by Ruhlusaraç (2020), who used Metra Potential Method (MPM) while determining each activity's due date. The activities are shown in the nodes with a network created by the MPM method, and forward-backwards calculations are made. So, the activity's latest finish times, earliest finish times, latest start times, and earliest start times are calculated. In the problem, it is deemed appropriate to take the latest finish times of the activities as the due date of the activities instead of using random numbers. It is desired to eliminate delays by ensuring that each activity is delivered at the latest finish time.

$$Activity\ Due\ Date\ (ADD) = Activity\ Latest\ Finish\ Time$$

Original problem and its scenarios

The study was conducted using real-life data from a boutique furniture project company that produces cloakrooms, TV units, wardrobes, and kitchen projects in Kayseri, Turkey. The company has a CNC machine, a horizontal cutting machine, and three identical machines with the same estimated loading rate: an edge-banding machine, a hole punching machine, and a spray gun. There are also three masters and four workers using these machines. The resources are as follows: three masters (R_1), four workers (R_2), one CNC machine (R_3), one horizontal cutting machine (R_4) and three identical machines (R_5). The Original Static Problem (OSP) data is given in Table 2-3 and Figure 1-2. Table 2 gives the characteristics of the real-life datasets. Table 3 shows the resource usage amounts under different modes for a kitchen furniture project. The datasets generated by the survey research and analysed during the current study are available in the Open Science Framework repository, osf.io/692xt/.

Kitchen, TV unit, and cloakroom projects consist of 34, 24, and 24 activities, respectively. Figure 1 indicates that the project networks that the first and last nodes represent dummy activities which are the start and finish activities with zero processing time. Figure 2 shows the primary resources of Mode

1 and Mode 2. In Mode 1, four workers can use a horizontal cutting machine and identical other machines. A master and a worker can operate the horizontal cutting machine. In Mode 2, three masters can operate a CNC machine. A master can operate CNC machines.

Original problem scenario 1.1 (OPS-11)

At *k= 120*, an unexpected situation occurs in the execution of the schedule due to the illness of 1 master (R_1 resource). The workplace doctor reports that the treatment of the master will be done quickly, and this process will take about 60 minutes so that the master can return to work at *k = 180*.

Original problem scenario 1.2 (OPS-12)

At *k = 120*, an unexpected situation occurs in the execution of the schedule due to the illness of 1 worker (R_2 resource). The workplace doctor reports that the worker's treatment cannot be done immediately. This process will continue throughout the scheduling, so the worker cannot return to work until *k = 4000*.

Original problem scenario 2.1 (OPS-21)

Suppose a new cloakroom project consisting of 24 activities arrives in our project execution environment at *k = 150,* and our response time is 10 minutes. The reactive schedule will be implemented at *k = 160*. Until *k = 180*, the situation of OPS-11, that is, the loss of 1 unit of R_1 the resource will continue. Thus, two unexpected events will be added to the original problem as a dynamic situation.

Original problem scenario 2.2 (OPS-22)

Suppose a new cloakroom project consisting of 24 activities arrives in the project execution environment at *k = 150,* and our response time is 10 minutes. The reactive schedule will be implemented at *k = 160*. Meanwhile, until *k = 4000*, that is, until the scheduling time horizon ends, the OPS-12 condition (R_2 resource loss) will continue. Thus, two unexpected events will be added to the original problem as a dynamic situation.

Table 2: Characteristics of the Datasets for the Kitchen Project

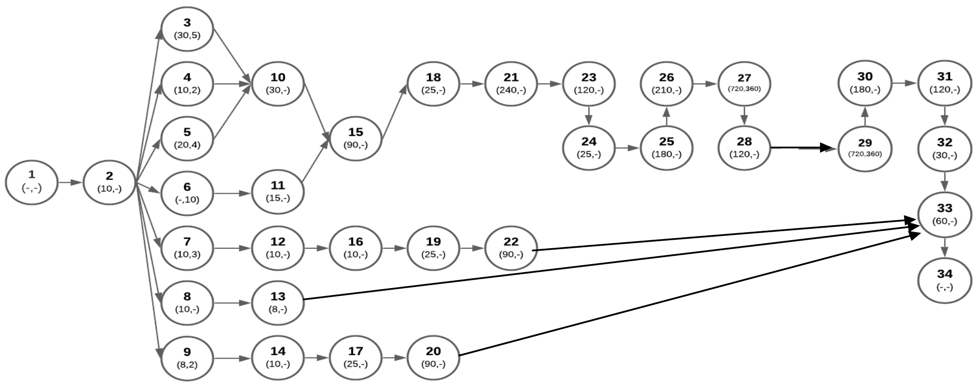
Activities	*Successor*	p_{ij1}	p_{ij2}	ADD	EPC	TPC	AC_1	AC_2
1	2	0	-	1	0	0	0,00	-
2	3,4,5,6,7,8,9	10	-	11	2,78	5,56	2,78	-
3	10	30	5	41	14,39	25,78	22,91	2,88
4	10	10	2	41	4,54	8,79	7,64	1,15
5	10	20	4	41	9,09	17,57	15,27	2,30
6	11	-	10	56	6,21	11,52	-	5,76
7	12	10	3	1971	6,18	9,36	7,64	1,73
8	13	10	-	2098	1,36	1,22	2,29	-
9	14	8	2	1981	5,13	7,26	6,11	1,15
10	15	30	-	71	13,93	24,86	12,43	-
11	15	15	-	71	6,67	12,44	6,22	-
12	16	10	-	1981	6,08	9,16	4,58	-
13	33	8	-	2106	4,42	7,34	3,67	-
14	17	10	-	1991	4,72	6,44	3,22	-
15	18	90	-	161	44,25	82,50	41,25	-
16	19	10	-	1991	4,72	6,44	3,22	-
17	20	25	-	2016	14,18	25,36	12,68	-
18	21	25	-	186	12,96	22,92	11,46	-
19	22	25	-	2016	14,18	25,36	12,68	-
20	33	90	-	2106	5,34	7,68	3,84	-
21	23	240	-	426	111,50	220,00	110,00	-
22	33	90	-	2106	5,34	7,68	3,84	-
23	24	120	-	546	40,87	77,24	38,62	-
24	25	25	-	571	2,57	2,14	1,07	-

Note: ADD: Activity Due Date, EPC: Earliness Penalty Cost, TPC: Tardiness Penalty Cost, AC: Activity Cost

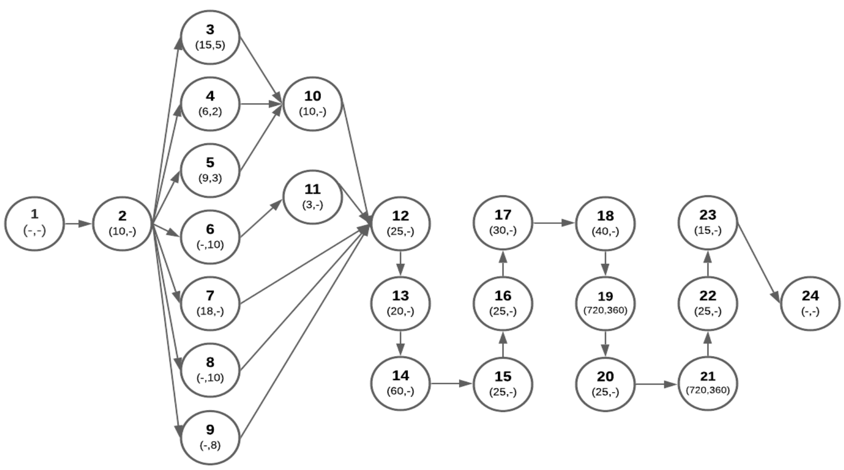
Table 3: The Resource Usage Amounts Under Different Modes for the Kitchen Project

	R_1	R_2	R_3	R_4	R_5
Activities	M_1/M_2	M_1/M_2	M_1/M_2	M_1/M_2	M_1/M_2
1	-/-	-/-	-/-	-/-	-/-
2	-/-	-/-	-/-	-/-	-/-
3	1/1	2/-	-/1	1/-	-/-
4	1/1	2/-	-/1	1/-	-/-
5	1/1	2/-	-/1	1/-	-/-
6	-/1	-/-	-/1	-/-	-/-
7	1/1	2/-	-/1	1/-	-/-
8	-/-	1/-	-/-	-/-	1/-
9	1/1	2/-	-/1	1/-	-/-
10	-/-	2/-	-/-	-/-	1/-
11	-/-	2/-	-/-	-/-	1/-
12	-/-	2/-	-/-	-/-	2/-
13	-/-	2/-	-/-	-/-	2/-
14	1/-	-/-	-/-	-/-	1/-
15	-/-	2/-	-/-	-/-	2/-
16	1/-	-/-	-/-	-/-	1/-
17	1/-	1/-	-/-	-/-	1/-
18	-/-	2/-	-/-	-/-	2/-
19	1/-	1/-	-/-	-/-	1/-
20	-/-	-/-	-/-	-/-	-/-
21	-/-	2/-	-/-	-/-	2/-
22	-/-	-/-	-/-	-/-	-/-
23	1/-	-/-	-/-	-/-	1/-
24	-/-	-/-	-/-	-/-	-/-

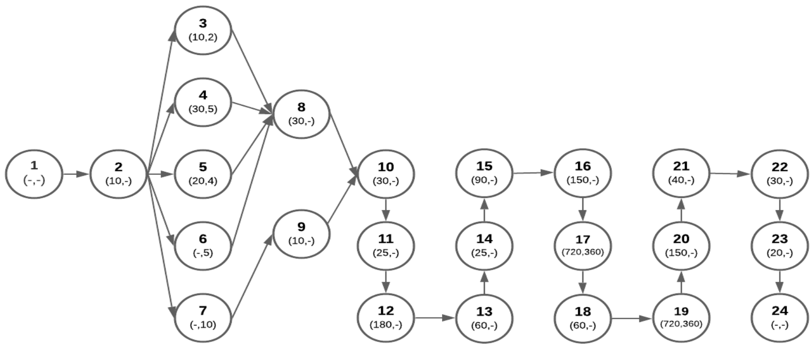
Note: R_1: 3 Masters, R_2: 4 Workers, R_3: 1 CNC, R_4: 1 Horizontal Cutting Machine, R_5: 3 Identical Machines, M: Mode



a) The Network of Kitchen Project (Project 1)



b) The Network of TV Unit Project (Project 2)



c) The Network of Cloakroom Project (Project 3 and 4)

Figure 1: The Projects Networks of the Original Problem

Mode 1: Horizontal Cutting Machine and Workers **Mode 2:** CNC and Masters

Figure 2: The Primary Resources of Mode 1 and Mode 2

Experimental results

The data collected from the company was solved with a CPLEX solver in the GAMS optimization program. A personal computer handled the computational results with an Intel Core i7 processor, 2.0 GHz, and 4 GB RAM.

The results of the original problem and its scenarios

According to the initial problem results, the completion time is 2912 minutes. There are also three activities completed early and 40 activities completed late. The early finished activities are only in the second project and are completed at most 12 minutes early. Late-finished activities are present in all projects, and there was a maximum delay of 379 minutes. The Z value is 453.057. Table 4 shows the Z, Z', TC, FT, and CPU times result from the original problem scenarios with β values. Some OPS-11 solutions seem to improve OSP solutions because some are near-optimal solutions.

Table 4: The Results of the Original Problem and Its Scenarios

		β=1	β=10	β=100	β=1000
OSP	Z=TC	453.057			
OPS-11	Z'	452.481	444.379	460.821	498.048
	TC	452.274	442.039	451.221	452.048
	FT	207	234	96	46
	CPU	1446	1434	824	1443
OPS-12	Z'	495.687	498.372	531.123	892.690
	TC	494.948	493.502	493.523	502.690
	FT	739	487	376	390
	CPU	1029	1732	997	1811
OPS-21	Z'	1.711.683	1.733.286	1.716.728	1.801.326
	TC	1.711.394	1.731.516	1.703.528	1.792.326
	FT	289	177	132	9
	CPU	1828	1913	1976	1928
OPS-22	Z'	1.853.235	1.872.621	1.907.321	1.981.750
	TC	1.852.901	1.869.811	1.888.121	1.927.750
	FT	334	281	192	54
	CPU	2485	2333	1396	1531

Note: Z: Objective Function Value, TC: Total Cost, FT: Finish Times Differences, CPU: Central Process Unit Time, β: Coefficient

Table 5: The Results of Dynamism Measurement of the Original Problem

OPS-11	DD	0,25	Medium-dynamic
	EDD	0,0075	Hard-dynamic
	EDD-TW	0,25	Medium-dynamic
OPS-12	DD	0,25	Medium-dynamic
	EDD	0,0075	Hard-dynamic
	EDD-TW	0,25	Medium-dynamic
OPS-21	DD	0,40	Medium-dynamic
	EDD	0,0135	Hard-dynamic
	EDD-TW	0,3995	Medium-dynamic
OPS-22	DD	0,40	Medium-dynamic
	EDD	0,0135	Hard-dynamic
	EDD-TW	0,3995	Medium-dynamic

Note: DD: Degree of Dynamism, EDD: Effective Degree of Dynamism, EDD-TW: EDD with Time Windows

Table 5 shows that the environment in all scenarios is at least medium-dynamic. According to the calculations of effective dynamism degree (EDD), it is seen that the environments are hard dynamic.

In Figure 3, Z and Z' values, which are the sum of the beta multiplied FT values, and the total cost in the scenarios and the total cost in the original problem generally increase as the scenarios are derived and the beta coefficient increases. In addition, in Figure 3, FT values generally decrease as the beta coefficients increase in each scenario.

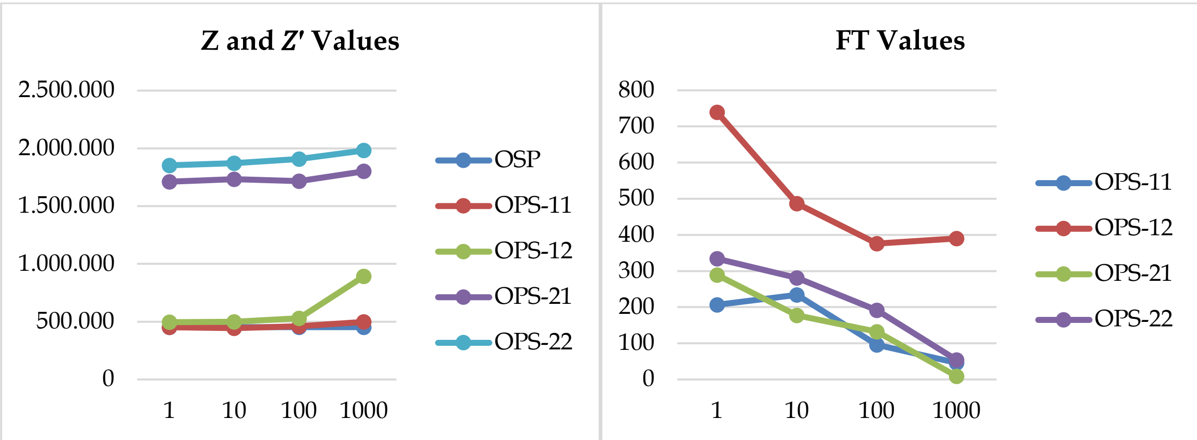


Figure 3: The Graphs of the Z, Z' and FT Values of the Original Problem and Its Scenarios

In Figure 4, the early finished activities are generally average, and seven in the OPS-11 instance. On the other hand, the late finished activities increase even more, especially in the OPS-21 and the OPS-22 instances with multiple disruption situations.

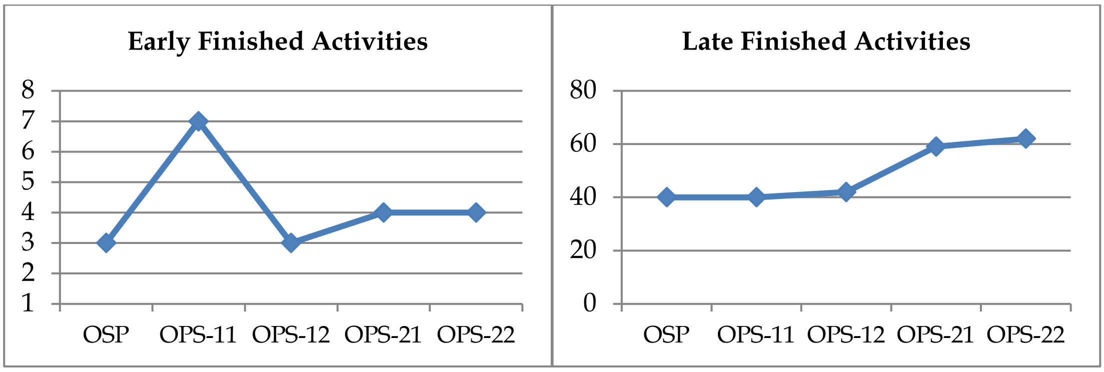


Figure 4: The Graphs of the Early and Late Finished Activities

Figure 5 gives the OSP Gantt diagram. Figure 6 shows disruption events on the first part of the OPS Gantt diagrams with red dashed lines. At the same time, it is possible to see the slight differences in the execution of the activities from these first parts: figures 5 and 6 show i, j, m notation. So, for example, activities 2 (j) of projects 1, 2, and 3 (i) are executed on Mode 1 (m) at the same time (1-11 minutes) for all Gantt diagrams.

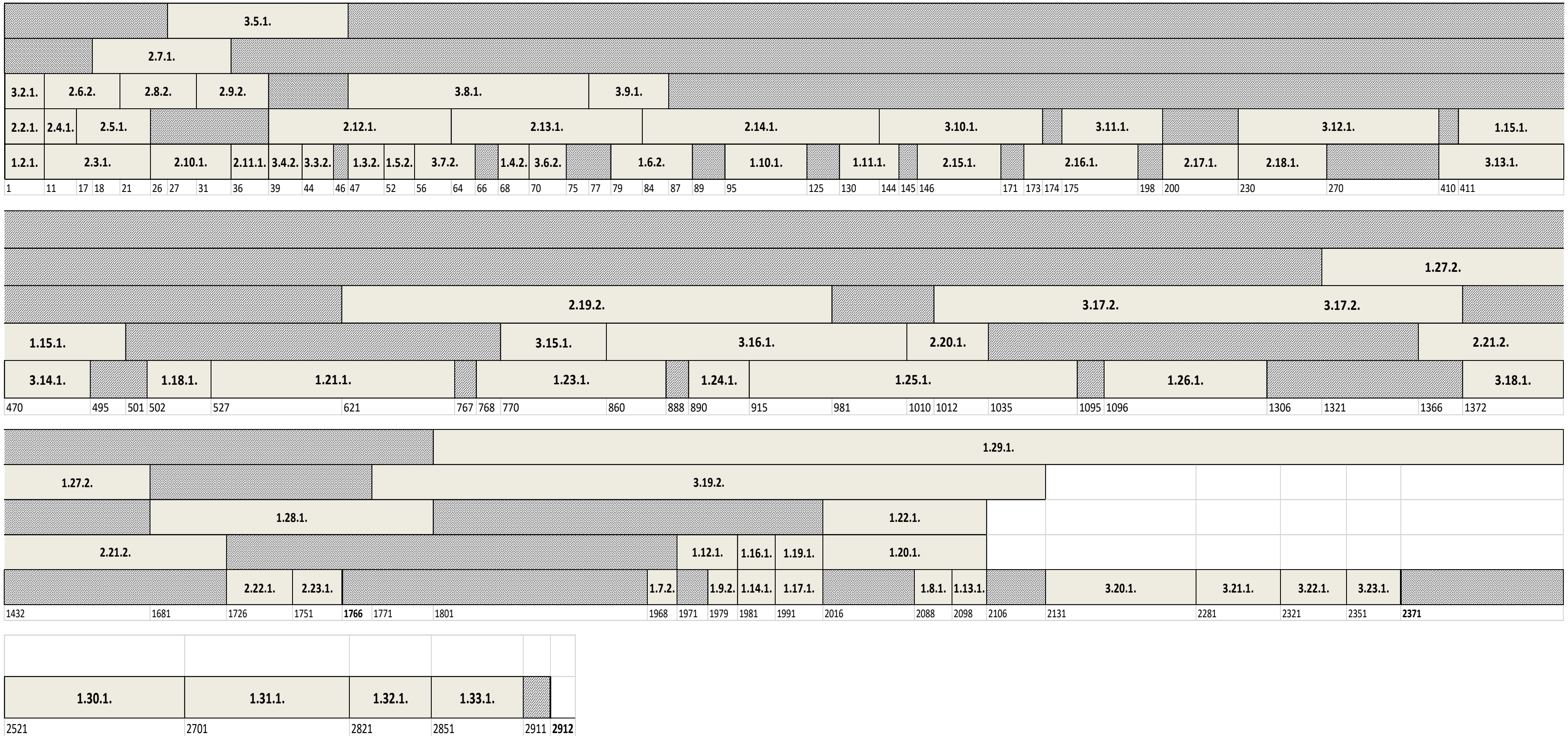


Figure 5: The Gantt Diagram of the Original Static Problem (OSP)

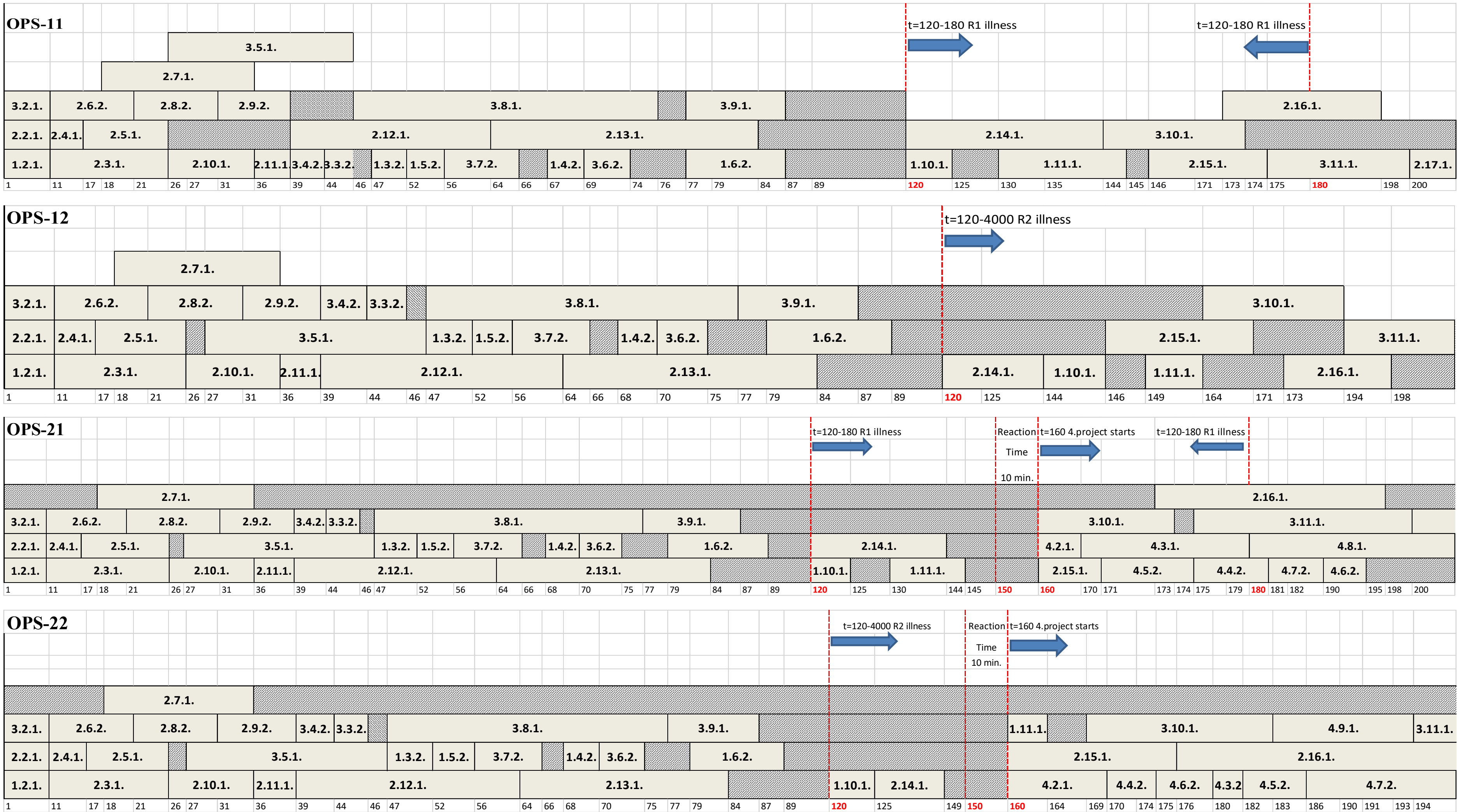


Figure 6: The Gantt Diagram of the Original Problem Scenarios (OPS)

Conclusion

According to the original problem solutions, the completion time of all projects (makespan), which was 2912 minutes in the original static problem, did not increase, even in cases with long-lasting resource shortages. Despite increasing projects in four disruption scenarios, the makespan was maintained by using flexibility in the baseline schedule. In the study, temporary resource loss was not critical and did not increase the makespan. However, it changed some activities' start and finished times and significantly affected earliness and tardiness costs. Therefore, in different problems, the makespan may increase depending on the importance of resource loss or the inflexibility of the schedule.

Based on the scenario solution results, it was observed that activities could be completed late rather than early finish potential in the baseline schedule when various disruptions occur. In addition, it was determined that the number of delayed activities increased in scenario examples with more than one disruption in the baseline schedule. Three activities were completed early in the original static problem, and forty were completed late.

Looking at total costs (TC), it was seen that reactive schedules were often more costly than the baseline schedule, and the cost increased, even more when multiple or prolonged disruptions occurred. In addition, it was determined that this significant increase in the share of mode selection costs was minimal. Instead, the large share belonged to the high number and long-delayed activities.

While generating optimal reactive schedules for dynamic situations, the beta coefficients were taken as 1000 in each sample by asking that the FT value be less. It was tried to approach zero for the FT value in all scenario examples. However, FT values were not zero in any scenario, and increasing the beta coefficient increased the Z' value.

Due to the intensity experienced in some production departments, some bottlenecks occurred. Therefore, it was observed that some activities were completed later than the desired due dates. If these delays were prolonged, late completed activities increased the total cost considerably. We proposed that the company increase capacities in the paint and sanding departments, which have bottlenecks according to the problem solutions. As seen from the results, increasing the number of resources will also prevent bottlenecks. Looking at the examples involving dynamic scenarios, the significant changes, especially at the beginning of the scheduling, greatly affected the functioning of the baseline schedule. However, the deviation of the finish time between the schedules was kept to a minimum.

This study is the first to consider the DMRCMPSP-WET via MILP and a predictive-reactive scheduling procedure. A set of disruption scenarios were considered to evaluate the performance of the procedure. Computational results indicated that the proposed models could provide optimal solutions to small-size real-life problems. A metaheuristic algorithm may be required to obtain near-optimal solutions for larger-size problems. Future studies may focus on developing metaheuristics to solve more complicated real-life problems or benchmark instances.

Peer-review:

Externally peer-reviewed

Conflict of interests:

The authors have no conflict of interest to declare.

Grant Support:

The authors declared that this study had received no financial support.

Author Contributions:

Idea/Concept/Design: **M.R., F.Ç.** Data Collection and/or Processing: **M.R.** Analysis and/or Interpretation: **M.R.** Literature Review: **M.R.** Writing the Article: **M.R.** Critical Review: **F.Ç.** Approval: **M.R., F.Ç.**

References

Afshar-Nadjafi, B., Basati, M., & Maghsoudlou, H., (2017). “Project scheduling for minimizing temporary availability cost of rental resources and tardiness penalty of activities”. Applied Soft Computing Journal, Vol. 61, 536–548. https://doi.org/10.1016/j.asoc.2017.08.033

Afshar-Nadjafi, B., & Shadrokh, S., (2008). "An algorithm for the weighted earliness-tardiness unconstrained project scheduling problem". Journal of Applied Sciences, Vol. 8 (9), 1651–1659. https://doi.org/10.3923/jas.2008.1651.1659

Aouam, T., & Vanhoucke, M., (2019). "An agency perspective for multi-mode project scheduling with time/cost trade-offs". Computers and Operations Research, Vol. 105, 167–186. https://doi.org/10.1016/j.cor.2019.01.012

Aytug, H., Lawley, M.A., McKay, K., Mohan, S., & Uzsoy, R., (2005). "Executing production schedules in the face of uncertainties: A review and some future directions". European Journal of Operational Research. Vol. 161, 86-110. https://doi.org/10.1016/j.ejor.2003.08.027

Bold, M., & Goerigk, M., (2021). "A compact reformulation of the two-stage robust resource-constrained project scheduling problem". Computers and Operations Research, Vol. 130, 105232. https://doi.org/10.1016/j.cor.2021.105232

Bozorgirad, M. A., & Logendran, R., (2012). "Sequence-dependent group scheduling problem on unrelated-parallel machines". Expert Systems with Applications, Vol. 39 (10), 9021–9030. https://doi.org/10.1016/j.eswa.2012.02.032

Burgelman, J., & Vanhoucke, M., (2018). "Maximising the weighted number of activity execution modes in project planning". European Journal of Operational Research. Vol. 270 (3), 999–1013. https://doi.org/10.1016/j.ejor.2018.04.035

Capa C., & Ulusoy, G., (2015). "Proactive project scheduling in an R&D department a bi-objective genetic algorithm". 2015 International Conference on Industrial Engineering and Operations Management (IEOM), 1-6, doi: 10.1109/IEOM.2015.7093733.

Chakrabortty, R.K., Rahman, H.F., & Ryan, M.J., (2020). "Efficient priority rules for project scheduling under dynamic environments: A heuristic approach". Computers and Industrial Engineering.140, 106287. https://doi.org/10.1016/j.cie.2020.106287

Chakrabortty, R.K., Rahman, H.F., Haque, K.M.A., Paul, S.K., & Ryan, M.J., (2021). "An event-based reactive scheduling approach for the resource constrained project scheduling problem with unreliable resources". Computers and Industrial Engineering. Vol. 151, 106981. https://doi.org/10.1016/j.cie.2020.106981

Chakrabortty, R.K., Sarker, R.A., & Essam, D.L., (2016). "Multi-mode resource-constrained project scheduling under resource disruptions". Computers and Chemical Engineering. Vol. 88, 13–29. https://doi.org/10.1016/j.compchemeng.2016.01.004

Chakrabortty, R.K., Sarker, R.A., & Essam, D.L., (2018). "Single mode resource-constrained project scheduling with unreliable resources". Operational Research. 1–35. https://doi.org/10.1007/s12351-018-0380-7

Changchun, L., Xi, X., Canrong, Z., Qiang, W., & Li, Z., (2018). "A column generation based distributed scheduling algorithm for multi-mode resource-constrained project scheduling problem". Computers and Industrial Engineering. Vol. 125(September), 258–278. https://doi.org/10.1016/j.cie.2018.08.036

Cheng, J., Fowler, J., Kempf, K., & Mason, S., (2015). "Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting". Computers and Operations Research. Vol. 53, 275-287. http://dx.doi.org/10.1016/j.cor.2014.04.018

Chtourou, H., & Haouari, M., (2008). "A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling". Computers and Industrial Engineering. Vol. 55 (1), 183–194. https://doi.org/10.1016/j.cie.2007.11.017

Collyer, S., (2015). Managing amidst rapid change : management approaches for dynamic environments. Project Management Institute. Inc. Retrieved from https://books.google.co.uk/books/about/Managing_Amidst_Rapid_Change.html?id=g_KuDgAAQBAJ&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q&f=false

Da Silva, A.R.V., & Ochi, L.S., (2010). Hybrid heuristics for dynamic resource-constrained project scheduling problem. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6373 LNCS, 73–87. https://doi.org/10.1007/978-3-642-16054-7_6

Da Silva, A.R.V., Ochi, L.S., & Santos, H.G., (2008). "New effective algorithm for dynamic resource-constrained project scheduling problem". EngOpt 2008 - International Conference on Engineering Optimization. Rio de Janeiro, Brazil, 01 - 05 June 2008.

Damay, J., Quilliot, A., & Sanlaville, E., (2007). "Linear programming based algorithms for preemptive and non-preemptive RCPSP". European Journal of Operational Research. Vol. 182 (3), 1012–1022. https://doi.org/10.1016/j.ejor.2006.09.052

Davari, M., & Demeulemeester, E., (2017). "The proactive and reactive resource-constrained project scheduling problem: the crucial role of buffer-based reactions". KU Leuven, Faculty of Economics and Business, KBI_1707. http://dx.doi.org/10.2139/ssrn.2976350

Davari, M., & Demeulemeester, E., (2018). "Important classes of reactions for the proactive and reactive resource-constrained project scheduling problem". Annals of Operations Research, 1–24. https://doi.org/10.1007/s10479-018-2899-7

Deblaere, F., Demeulemeester, E., & Herroelen, W., (2011). "Reactive scheduling in the multi-mode RCPSP". Computers and Operations Research. Vol. 38 (1), 63–74. https://doi.org/10.1016/j.cor.2010.01.001

Golestaneh, R., Jafari, A., Khalilzadeh, M., & Karimi, H., (2013). "Minimizing total resource tardiness penalty costs in the resource-constrained project scheduling problem with metaheuristic algorithms". International Journal of Research in Industrial Engineering, Vol. 2(3), 47-57.

Gonçalves, J.F., Mendes, J.J.M., & Resende, M.G.C., (2015). The basic multi-project scheduling problem. In Handbook on Project Management and Scheduling Vol. 2 (pp. 667–683). Springer International Publishing.

Hartmann, S., & Briskorn, D., (2010). "A survey of variants and extensions of the resource-constrained project scheduling problem". European Journal of Operational Research. Vol. 207 (1), 1–14. https://doi.org/10.1016/j.ejor.2009.11.005

Hauder, V.A., Beham, A., Raggl, S., Parragh, S.N., & Affenzeller, M., (2020). "Resource-constrained multi-project scheduling with activity and time flexibility". Computers and Industrial Engineering. Vol. 150, 106857. https://doi.org/10.1016/j.cie.2020.106857

Hazır, Ö., & Ulusoy, G., (2020). "A classification and review of approaches and methods for modeling uncertainty in projects". International Journal of Production Economics, Vol. 223, 107522, ISSN 0925-5273, https://doi.org/10.1016/j.ijpe.2019.107522.

He, N., Z. Zhang, D., & Yuce, B., (2022) "Integrated multi-project planning and scheduling - a multiagent approach", European Journal of Operational Research, Vol. 302, 2, 688-699, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2022.01.018.

Herroelen, W., & Leus, R., (2005). "Project scheduling under uncertainty: survey and research potentials". European Journal of Operational Research. Vol. 165, 289-306.

Joo, B. J., Chua, T. J., Cai, T. X., & Chua, P. C., (2019). "Coordination-based reactive resource-constrained project scheduling". Procedia CIRP. Vol. 81, 51–56. https://doi.org/10.1016/j.procir.2019.03.010

Kéri, A., & Kis, T., (2006) Primal-dual combined with constraint propagation for solving RCPSPWET. In: Haasis HD., Kopfer H., Schönberger J. (eds) Operations Research Proceedings 2005. Operations Research Proceedings, vol 2005. Springer, Berlin, Heidelberg . https://doi.org/10.1007/3-540-32539-5_107

Khoshjahan, Y., Najafi, A.A., & Afshar-Nadjafi, B., (2013). "Resource-constrained project scheduling problem with discounted earliness-tardiness penalties: Mathematical modeling and solving procedure". Computers and Industrial Engineering. Vol. 66 (2), 293–300. https://doi.org/10.1016/j.cie.2013.06.017

Kolisch, R., Sprecher, A., & Drexl, A., (1995). "Characterization and generation of a general class of resource-constrained project scheduling problems". Management Science. Vol. 41 (10), 1693–1703. https://doi.org/10.1287/mnsc.41.10.1693

Koné, O., Artigues, C., Lopez, P., & Mongeau, M., (2011). "Event-based MILP models for resource-constrained project scheduling problems". Computers and Operations Research. Vol. 38, 3–13. https://doi.org/10.1016/j.cor.2012.10.018

Kreter, S., Rieck, J., & Zimmermann, J., (2016). "Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars". European Journal of Operational Research. Vol. 251 (2), 387–403. https://doi.org/10.1016/j.ejor.2015.11.021

Krüger, D., & Scholl, A., (2009). "A heuristic solution framework for the resource-constrained (multi-)project scheduling problem with sequence-dependent transfer times". European Journal of Operational Research. Vol. 197 (2), 492–508. https://doi.org/10.1016/j.ejor.2008.07.036

Kurt P.A., & Kececi B. (2019) Resource Constrained Multi-project Scheduling: Application in Software Company. In: Karwowski W., Trzcielinski S., Mrugalska B., Di Nicolantonio M., Rossi E. (eds) Advances in Manufacturing, Production Management and Process Control. AHFE 2018. Advances in Intelligent Systems and Computing, Vol 793. Springer, Cham. https://doi.org/10.1007/978-3-319-94196-7_51

Larsen, A., & Madsen, O.B.G., (2000). The dynamic vehicle routing problem. Kgs. Lyngby, Denmark: Technical University of Denmark (DTU). (IMM-PHD; No. 2000-73).

Larsen A., Madsen O.B.G., & Solomon M.M., (2007). Classification of dynamic vehicle routing systems. In: Zeimpekis V., Tarantilis C.D., Giaglis G.M., Minis I. (eds) Dynamic Fleet Management. Operations Research/Computer Science Interfaces Series, Vol 38. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-71722-7_2

Ma, Z., Demeulemeester, E., He, Z., & Wang, N., (2019). "A computational experiment to explore better robustness measures for project scheduling under two types of uncertain environments". Computers and Industrial Engineering, 1–28. https://doi.org/10.1016/j.cie.2019.04.014

Oztemel, E., & Selam, A.A., (2017). "Bees Algorithm for multi-mode, resource-constrained project scheduling in molding industry". Computers and Industrial Engineering, Vol. 112, 187-196. http://dx.doi.org/10.1016/j.cie.2017.08.012

Pamay, M.B., (2011). A linear programming based method for the resource-constrained multi-project scheduling problem with weighted earliness/tardiness costs. [MSc dissertation]. The University of Sabancı, Istanbul.

Rahman, H.F., Chakrabortty, R.K., & Ryan, M.J., (2021). "Scheduling project with stochastic durations and time-varying resource requests: A metaheuristic approach". Computers and Industrial Engineering, Vol. 157, 107363. https://doi.org/10.1016/j.cie.2021.107363

Ruhlusaraç, M., (2020). Dynamic multi-mode resource-constrained multi-project scheduling with weighted earliness and tardiness and its application. [PhD dissertation]. The University of Erciyes, Kayseri.

Ruhlusaraç, M., & Çalışkan, F., (2018). Multi-mode resource-constrained multi-project scheduling problem with weighted earliness and tardiness. 9th International Science and Technology Conference. July 18-20, 2018 Paris, France, 60-64.

Ruhlusaraç, M., & Çalışkan, F., (2020). "A mathematical model for dynamic project scheduling problem and reactive scheduling implementation". Business & Management Studies: An International Journal. Vol. 8 (4), 83-97. https://doi.org/10.15295/bmij.v8i4.1708

Sabuncuoglu, I., & Bayiz, M., (2000). "Analysis of reactive scheduling problems in a job shop environment". European Journal of Operational Research. Vol. 126 (3), 567–586. https://doi.org/10.1016/S0377-2217(99)00311-2

Saif, U., Yue, L., & Awadh, M. A., (2022). "Coordinated Planning and Scheduling of Multiple Projects With New Projects Arrival Under Resource Constraint Using Drum Buffer Rope Heuristic", IEEE Access, Vol. 10, 84244-84266, https://doi.org/10.1109/ACCESS.2022.3195045.

Salemi Parizi, M., Gocgun, Y., & Ghate, A., (2017). "Approximate policy iteration for dynamic resource-constrained project scheduling". Operations Research Letters. Vol. 45 (5), 442–447. https://doi.org/10.1016/j.orl.2017.06.002

Schwindt C., (2000). Minimizing earliness-tardiness costs of resource-constrained projects. In: Inderfurth K., Schwödiauer G., Domschke W., Juhnke F., Kleinschmidt P., Wäscher G. (eds) Operations Research Proceedings 1999. Operations Research Proceedings 1999 (Selected Papers of the Symposium on Operations Research (SOR' 99) Magdeburg, September 1–3, 1999), vol 1999. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-58300-1_62

Song, W., Xi, H., Kang, D., & Zhang, J., (2018). "Simulation modelling practice and theory an agent-based simulation system for multi-project scheduling under uncertainty". Simulation Modelling Practice and Theory. 86(November 2017), 187–203. https://doi.org/10.1016/j.simpat.2018.05.009

Su, C.T., Santoro, M.C., & Mendes, A.B., (2018). "Constructive heuristics for project scheduling resource availability cost problem with tardiness". Journal of Construction Engineering and Management, Vol. 144 (8). https://doi.org/10.1061/(ASCE)CO.1943-7862.0001524.

Tao, S., & Dong, Z.S., (2018). "Multi-mode resource-constrained project scheduling problem with alternative project structures". Computers and Industrial Engineering, Vol. 125, 333–347. https://doi.org/10.1016/j.cie.2018.08.027

Ulusoy, G., Hazır, Ö., (2021). An introduction to project modeling and planning, Springer Texts in Business and Economics, https://doi.org/10.1007/978-3-030-61423-2.

Van de Vonder, S., (2006). "Proactive-reactive procedures for robust project scheduling". Applied Economics, (247), 239.

Van de Vonder, S., Ballestín, F., Demeulemeester, E., & Herroelen, W., (2007a). "Heuristic procedures for reactive project scheduling". Computers and Industrial Engineering. Vol. 52 (1), 11–28. https://doi.org/10.1016/j.cie.2006.10.002

Van de Vonder, S., Demeulemeester, E., & Herroelen, W., (2008). "Proactive heuristic procedures for robust project scheduling: an experimental analysis". European Journal of Operational Research. 189 (3), 723–733. https://doi.org/10.1016/j.ejor.2006.10.061

Van de Vonder, S., Demeulemeester, E., & Herroelen, W., (2007b). "A classification of predictive-reactive project scheduling procedures". Journal of Scheduling. Vol. 10 (3), 195–207. https://doi.org/10.1007/s10951-007-0011-2

Van Peteghem, V., & Vanhoucke, M., (2014). "An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances". European Journal of Operational Research. Vol. 235 (1), 62–72. https://doi.org/10.1016/j.ejor.2013.10.012

Vanhoucke, M., Demeulemeester, E., Herroelen, W., (2001). "An exact procedure for the resource-constrained weighted earliness – tardiness project scheduling". Annals of Operations Research. Vol. 102, 179–196. https://doi.org/10.1023/A:1010958200070

Voß, S., & Witt, A., (2007). "Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements : a real-world application". Int. J. Production Economics. Vol. 105, 445–458. https://doi.org/10.1016/j.ijpe.2004.05.029

Wang, H., Lappas, N. H., & Gounaris, C., (2019). "Multi-mode resource-constrained project scheduling with alternative prerequisites : new models and computational studies". Industrial & Engineering Chemistry Research. 1–35. https://doi.org/10.1021/acs.iecr.9b02455

Wang, W., Ge, X., Li, L., & Su, J., (2019). "Proactive and reactive multi-project scheduling in uncertain environment". IEEE Access. Vol. 7, 88986–88997. https://doi.org/10.1109/access.2019.2926337

Watermeyer, K., & Zimmermann, J., (2020). "A branch-and-bound procedure for the resource-constrained project scheduling problem with partially renewable resources and general temporal constraints". OR Spectrum. Vol. 42, 427-460. https://doi.org/10.1007/s00291-020-00583-z

Yassine, A.A., Mostafa, O., & Browning, T.R., (2017). "Scheduling multiple, resource-constrained, iterative, product development projects with genetic algorithms". Computers and Industrial Engineering. Vol. 107, 39-56, ISSN 0360-8352, https://doi.org/10.1016/j.cie.2017.03.001.

Yuan, Y., Ye, S., Lin, L., & Gen, M., (2021). "Multi-objective multi-mode resource-constrained project scheduling with fuzzy activity durations in prefabricated building construction". Computers and Industrial Engineering, doi: https://doi.org/10.1016/j.cie.2021.107316

Zhang, Y., Hu, X., Cao, X., & Wu, C., (2022). "An efficient hybrid integer and categorical particle swarm optimization algorithm for the multi-mode multi-project inverse scheduling problem in turbine assembly workshop". Computers & Industrial Engineering. Vol. 169, 108148, ISSN 0360-8352, https://doi.org/10.1016/j.cie.2022.108148.

Zhu, G., Bard, J. F., & Yu, G., (2005). "Disruption management for resource-constrained project scheduling". Journal of the Operational Research Society. Vol. 56, 365–381. https://doi.org/10.1057/palgrave.jors.2601860