

Melchiors, Philipp; Kolisch, Rainer; Kanet, John J.

**Article — Published Version**

## The performance of priority rules for the dynamic stochastic resource-constrained multi-project scheduling problem: an experimental investigation

Annals of Operations Research

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Melchiors, Philipp; Kolisch, Rainer; Kanet, John J. (2024) : The performance of priority rules for the dynamic stochastic resource-constrained multi-project scheduling problem: an experimental investigation, Annals of Operations Research, ISSN 1572-9338, Springer US, New York, NY, Vol. 338, Iss. 1, pp. 569-595,  
<https://doi.org/10.1007/s10479-024-05841-9>

This Version is available at:

<https://hdl.handle.net/10419/315289>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<http://creativecommons.org/licenses/by/4.0/>



# The performance of priority rules for the dynamic stochastic resource-constrained multi-project scheduling problem: an experimental investigation

Philipp Melchiors<sup>1</sup> · Rainer Kolisch<sup>1</sup> · John J. Kanet<sup>2</sup>

Received: 2 December 2022 / Accepted: 13 January 2024 / Published online: 26 February 2024  
© The Author(s) 2024

## Abstract

We consider the problem of a project manager of a matrix organization responsible for the timely completion of projects arriving over time and requiring the usage of a set of resources under his/her purview. A project is defined by a set of activities, precedence relations between activities, resource requirements, a customer due date, and a priority (a weight). Arriving projects are released to a flow control system that monitors the progress of activities and schedules the set of available activities, i.e., activities that are ready to be processed, as appropriate. A critical feature of such control systems is the decision process for choosing the next activity to seize a given resource. This is the focus of this paper. In the past several decades, various heuristic priority rules have been proposed in the literature to support this type of decision in differing settings such as the job shop problem and the deterministic resource-constrained project scheduling problem. A gap exists with respect to testing the various rules all together in the more realistic dynamic-stochastic multi-project environment when the objective is to minimize weighted project tardiness. The purpose of this paper is to fill this gap. Results show that the priority rule “Weighted Critical Ratio and Shortest Processing Time” ( $W(CR + SPT)$ ) is the best performing rule with respect to minimizing weighted project tardiness.  $W(CR + SPT)$  is shown to be a variant of the family of “Modified Due Date” rules first introduced by Baker and Bertrand (J Oper Manage 1(3):37–42, 1982). Repeated application of Duncan’s Multiple Range test demonstrates the robustness of our findings. For the environmental parameters (due date tightness, variation of expected activity durations, and utilization of resource), the  $W(CR + SPT)$  rule is dominant with respect to weighted project tardiness among the eleven priority rules tested. Only when the number of resources is very modest (either 1 or 3 resources) or under a purely parallel resource network

---

✉ Rainer Kolisch  
rainer.kolisch@tum.de

Philipp Melchiors  
philipp.melchiors@tum.de

John J. Kanet  
kanet@udayton.edu

<sup>1</sup> TUM School of Management, Technical University of Munich, Munich, Germany

<sup>2</sup> School of Business Administration, University of Dayton, Dayton, USA

is  $W(CR + SPT)$  not the dominant rule. In those cases, its variant  $W(CR + GSPT)$  is the best performing rule with respect to weighted project tardiness.

**Keywords** Multiple project · Scheduling · Constrained resources · Stochastic · Priority rules

## 1 Introduction

In many project-based organizations such as R&D, a matrix structure is employed with functional departments processing projects. Projects of different types (such as new development projects, incremental development projects) arrive in a stochastic manner and their activities have to be processed by the resources available in the functional departments. A relevant objective is to minimize average weighted project delay in which project delay (tardiness) is defined as the time the project finishes after its due date and weights indicate the importance of projects. The arrival time of projects and the duration of activities are stochastic. This problem is termed as dynamic stochastic resource-constrained multi-project scheduling problem (DSRCMPSP). Despite its practical relevance, only limited research has been undertaken so far to address this problem. In the literature, the multi-project scheduling problem is predominantly stated as static and deterministic with the set of projects given at the outset and all parameters such as activity durations known in advance. However, in reality these assumptions are generally not met. First, projects arrive continuously over time. Second, project parameters such as activity durations are stochastic. Thus, the multi-project scheduling problem is in many cases both dynamic and stochastic in nature. As we discuss in Sect. 2.2, under the assumption of Markovian interarrival times and activity processing times, the problem can be modeled as Markov Decision Processes. However, due the curse of dimensionality large-scale real-world problems cannot be solved to optimality. Hence, the prevailing method for approaching these real-life complicated conditions is via implementing a dynamic scheduling policy to construct a schedule using a priority rule for specifying the order of processing of activities by a resource. In addition to scalability, priority rules are prevalent in industry as they are easy to understand and to communicate. Against this background, this paper aims to provide a comprehensive comparison of well-known priority rules, which have been proposed for simpler settings, for the total weighted tardiness objective and to adapt them as needed for application to the more practically relevant DSRCMPSP. From the literature, in particular from the literature on the deterministic multi-project scheduling problem, we identify a number of priority policies which have performed well in prior computational studies. Whereas all these rules have shown promise for the total weighted tardiness objective, they have never been compared together in the same study nor in the dynamic stochastic resource-constrained setting, which we do here.

In order to introduce the problem formally, we follow the framework of Adler et al. (1995) and depict an organization as a set  $\mathcal{R}$  of resources. Resource  $r \in \mathcal{R}$  comprises  $c_r$  identical units, each capable of processing one activity at a time. Projects arrive dynamically according to a stochastic arrival process. We consider a stream of projects  $j = 1, \dots, J$  where each arriving project  $j$  is of type  $p_j \in \mathcal{P}$ . Project type  $p \in \mathcal{P}$  has a weight  $w_p$ , an interarrival rate  $\lambda_p$ , and is comprised of a set of activities  $\mathcal{V}_p$  and a set of precedence relations  $\mathcal{A}_p$  of the type finish-to-start with minimum time lag of 0. Each precedence relation between activity  $i$  and  $i'$  is written as a tuple  $(i, i') \in \mathcal{A}_p$ . In order to be processed, activity  $i \in \mathcal{V}_p$  seizes one unit of resource  $r_{ip} \in \mathcal{R}$  for a stochastic duration with mean  $\bar{d}_{ip}$ . Resource  $r \in \mathcal{R}$  has a capacity of  $c_r$  units. The set  $\mathcal{E}_r(t)$  refers to the set of activities being processed by resource  $r$  at time

$t \geq 0$ , where each activity  $i$  of project  $j$  is referred to by tuple  $(i, j)$ . On arrival of project  $j$  at time  $a_j$ , a due date  $D_j$  is assigned. We define the variables  $S_{ij} \geq 0$  as the start time of activity  $i$  of project  $j$ . Furthermore, let variable  $T_j \geq 0$  be the tardiness of project  $j$ . The objective is to minimize the long term expected average weighted project tardiness with  $J$  being the total number of projects arriving at the system:

$$\text{Min } Z = \lim_{J \rightarrow \infty} E \left[ \frac{1}{J} \sum_{j=1}^J w_j \cdot T_j \right] \quad (1)$$

subject to the following constraints

$$a_j \leq S_{ij} \quad j = 1, \dots, J; i \in \mathcal{V}_j \quad (2)$$

$$S_{i',j} + d_{i',j} \leq S_{i,j} \quad j = 1, \dots, J; (i', i) \in \mathcal{A}_{p_j} \quad (3)$$

$$|\mathcal{E}_r(t)| \leq c_r \quad r \in \mathcal{R}; t \geq 0 \quad (4)$$

$$S_{ij} + d_{ij} - D_j \leq T_j \quad j = 1, \dots, J; i \in \mathcal{V}_j \quad (5)$$

$$S_{ij}, T_j \geq 0 \quad j = 1, \dots, J; i \in \mathcal{V}_j \quad (6)$$

The decisions are on the start time of the activities  $S_{ij}$ . These must neither be before the arrival time  $a_j$  of their projects (Constraint (2)) nor of the finish time of each of its predecessor activities  $i' \in \mathcal{A}_{p_j}$  (Constraint (3)). Furthermore, for each resource  $r$  the number of activities processed at any time  $t$  must not exceed its capacity  $c_r$  (Constraint (4)). Finally, Constraint (5) sets the tardiness of each project  $j$ . The objective function (1) minimizes the long term expected average weighted project tardiness. Note that project arrival times  $a_j$  and duration of activities  $d_{ij}$  are random and thus the activity start  $S_{ij}$  and the project tardiness  $T_j$  are random variables.

This paper is organized as follows. In Sect. 2 we provide a brief survey of past related work. In Sect. 3 we present the priority rules used in our study. Section 4 provides the experimental design of our computational study. In particular, we define the problem parameters used. Section 5 reports the results of the computational study. Section 6 provides a summary and conclusion.

## 2 Relevant literature

We begin in Sect. 2.1 with a general review of the class of approaches to the DSRCMPSP, namely “scheduling policies” under which our work falls. Then, in Sect. 2.2 we briefly review the relevant literature to identify priority rules as used by scheduling policies found in various environments to be effective for minimizing weighted tardiness. Section 3 then follows with formal descriptions of the rules we consider and how (if needed) the rules were adapted for use in the DSRCMPSP.

### 2.1 Scheduling policies

The general approach that we take to address the DSRCMPSP is to employ a scheduling policy  $\omega$ . Different definitions of a scheduling policy can be found in the literature (e.g., see Stork, 2001). Here, we employ a general definition based on Fernandez et al. (1998).

**Definition 2.1** A scheduling policy  $\omega$  defines actions at decision times. A decision time  $t$  is either the time of a new project arrival  $a_j$  or the completion time of an activity  $S_{ij} + d_{ij}$ . An

action at decision time  $t$  consists for each resource  $r \in \mathcal{R}$  with at least one idle unit to select one activity  $(i, j) \in \mathcal{W}_r(t)$  from the set of activities waiting to be processed in the queue in front of the resource  $r$ .

A scheduling policy has to meet the nonanticipativity constraint (see Fernandez et al., 1996), which states that only information known up to decision time  $t$  may be used. We consider completion times of activities and arrival times of the projects arrived until  $t$ . For stochastic project scheduling problems various classes of policies have been proposed (see for example Möhring et al. 1984; Stork 2001). Our focus is on the class of resource-based priority policies (RBP) of Stork (2001) that is defined as follows:

**Definition 2.2** At decision time  $t$ , a resource-based priority policy (RBP policy) orders the activities in  $\mathcal{W}_r(t)$  based on priorities obtained by a priority rule and selects activities according to this order until there are either no more activities in the queue or no more resource units are available.

Note that resource-based priority policies have the non-delay property, i.e., no activity, which can be started at time  $t$  will be deliberately delayed to a later start time (see Kolisch, 1996). In processor scheduling this is termed as work-conserving policy (see Tassiulas & Georgiadis, 1996). Resource-based priority policies combine priority rules with the parallel schedule generation scheme (see Kolisch, 1996) that is compatible with the nonanticipativity constraint (see Ballestin, 2007). An alternative to RBPs are activity-based priority policies (ABPs) (see Stork, 2001) where the additional constraint exists that an activity with a lower priority must not be scheduled at an earlier time than an activity with a higher priority. From a theoretical point of view, ABPs have the advantage that Graham anomalies (see Graham, 1966) are avoided. However, experiments by Fliedner (2015) show that ABPs lead in many cases to poor performance or even unstable behavior with an unbounded long term average number of projects in the system. This is due to the fact that idleness of resources occurs quite frequently such that the busy periods of the resources are not sufficient to cope with the workload associated with the arriving projects. Recent computational results for the deterministic single project problem indicate that resource-based policies lead to solutions with smaller expected makespan (see Ballestin & Leus 2009, Ashtiani et al. 2011, Chen et al. 2015).

## 2.2 Relevant priority rule literature

We classify the existing related literature according to the two dimensions: project arrival (static vs. dynamic) and project information (deterministic vs. stochastic) resulting in four fields, which are discussed below.

**Static-deterministic multi-project scheduling** In the static-deterministic case of the multi-project scheduling problem, there are a number of projects which have to be scheduled. Each project is available at the beginning of the planning horizon and all data is deterministic. The objective is to minimize total weighted project delay. A recent survey of the literature in this field is given by Gómez Sánchez et al. (2022). Notable works include Kurtulus and Davis (1982), Kurtulus and Narula (1985), Lawrence and Morton (1993) as well as Tsai and Chiu (1996). In each case, the authors undertake computational studies where they test the performance of different priority rules. Browning and Yassine (2010a) extend the investigation to larger sets of priority rules and give a more in-depth analysis of the relationship between problem parameters and rules performance. Browning and Yassine (2016) extend the

consideration to portfolios of development projects. Building upon the work of Browning and Yassine (2010b), Van Eynde and Vanhoucke (2020) evaluate a large set of priority rules on a new set of instances. They differentiate the priority rules into “single project rules”, “coupled rules”, and “decoupled rules”. For single and coupled project rules, the multiple projects are modeled as a single super-project network. This is solved by single project rules using activity information only, while coupled rules are applying project and activity information. Decoupled rules are comprised of a pair of rules, and the projects are modeled separately. Using project information, the first rule is applied to select a project while the second rule, using activity information, is applied to choose an activity of the selected project. The experimental study shows better results for the decoupled rules. The best pair of priority rules for the project and the activity selection are Minimum Total Work Remaining (MINTWR) and Minimum Latest Start Time (MINLST), respectively, when solving the instances of Browning and Yassine (2010a), while for solving the newly generated instances of Van Eynde and Vanhoucke (2020), the best pair is Minimum Critical Path (MINCP) and Minimum Dynamic Slack (MINSLK). In a follow-up study, Van Eynde and Vanhoucke (2022) propose new summary measures for the static-deterministic multi-project scheduling problem and test the decoupled priority rules on benchmark instances, generated employing the new summary measures. Using the instance set of Van Eynde and Vanhoucke (2020), Bredael and Vanhoucke (2022) undertake an extensive comparison of ten re-implemented metaheuristics from the literature.

**Static-stochastic multi-project scheduling** In the static-stochastic multi-project scheduling setting, each project is available at the beginning of the planning horizon, and activity durations are stochastic. The literature on this problem is rather sparse. Wang et al. (2017) undertake a computational study to assess the performance of priority rules on quality and robustness. Quality is related to time-based objectives being the delay per project relative to its critical path length as well as an aggregated delay measure over all projects. Robustness refers to the stability of the results obtained for the case with deterministic activity durations in the presence of stochastic activity durations. The set of rules as well as the problem parameters to control the generation of problem instances is based on the work of Browning and Yassine (2010a). Wang et al. (2017) find that the Earliest-Due-Date-First (EDDF) rule performs best overall. Since we focus on the weighted tardiness objective in a dynamic-stochastic setting, we expect the results will not be applicable to our problem context.

**Dynamic-deterministic multi-project scheduling** In this setting, projects arrive over time and the arrival time of the projects is stochastic. At the arrival of a project, all information on that project is known with certainty. Dynamic deterministic multi-project scheduling is treated intensively by Dumond and Mabert (1988). In this category we also place relevant studies of the special case of the dynamic job shop scheduling problem. Notable works concerned with project (job) tardiness include: Baker and Kanet (1983), Vepsäläinen and Morton (1987), Anderson and Nyirenda (1990) and Kutunoglu and Sabuncuoglu (1999).

**Dynamic-stochastic multi-project scheduling** In this field of the literature, multiple projects arrive in a stochastic fashion over time. Once a project has arrived, the information about the duration of its activities is stochastic while all other project information is known with certainty. The seminal paper that addresses dynamic-stochastic multi-project scheduling is Adler et al. (1995). There, and later in Levy and Globerson (1997), a multi-project organization is modeled as a fork-join-queueing network in which projects arrive over time and each activity of a project has to be processed by one specific function (resource) of the orga-

nization. Activities of different projects are queued in front of the resources in order to be processed. Scheduling in this context is addressed by different approaches.

Optimal and near optimal policies are computed based on Markov Decision Processes (MDP) by Choi et al. (2007), Melchior (2015) and Satic et al. (2022). Since MDPs suffer from the curse of dimensionality, the authors limit the number of projects to make the problem more tractable. Besides that, Choi et al. (2007) and Satic et al. (2022) restrict their considerations to projects where activities are processed in a strict linear order.

For problem instances of realistic size, priority rules are investigated by different authors. Anavi-Isakow and Golany (2003) investigate the performance of a set of simple priority rules in the context of project control mechanisms where projects are backlogged in front of the system until a threshold for a specific state such as the number of projects in the system is met. They find for a single problem instance, with three project types and up to seven activities per project, that the Shortest Processing Time rule performed best w.r.t. average flow time per project. Melchior and Kolisch (2009) undertake a simulation study in which they show that the MINSLK rule as well as policies based on the bottleneck dynamic approach of Lawrence and Morton (1993) provide good results. Chen et al. (2019) extend the work by Wang et al. (2017) by taking into account new project arrivals. These are modeled using a second set of projects, referred to as insertion portfolio, from which a new project is selected at a random time as long as the number of projects being in process does not exceed a maximum number. On each arrival of a new project, the priority rule for scheduling is selected based on the state of the project portfolio, measured by the fraction of critical activities which have been scheduled. Since the definition of the dynamic-stochastic multi-project environment as well as the evaluation of solutions in Chen et al. (2019) differs fundamentally from ours, the results are difficult to compare.

Besides empirical works of limited scope, there are some theoretical results for scheduling fork-join-queueing networks. Baccelli et al. (1993) consider the case with a single project type and resources with a single unit. For this case, they show the optimality of the First-Come, First-Serve (FCFS) rule applied on the project level (where all activities of one project are prioritized over the activities of another project) for minimizing the long term expected project flow time. Since prioritizing on the project level leaves room for prioritization on the activity level, the rule defines the class of so-called FCFS-policies. In our study we assess the performance of FCFS-policies for minimizing total weighted tardiness for multiple project types.

## 2.3 Positioning our work

Major drawbacks of the studies undertaken so far for the dynamic-stochastic multi-project scheduling problem are the small sets of problem instances and limited time horizon of the investigation. Adler et al. (1995) and Levy and Globerson (1997) aim at demonstrating the power of fork-join-queueing network models as a process-oriented approach for the investigation of uncertainty in the context of dynamic and stochastic multi-project organizations. The authors draw on single case studies from collaborations with industry partners and consider, amongst other approaches, using priority rules for the scheduling task, where only a limited number of rules are considered.

Chen et al. (2019) use only a limited set of projects arriving in the future and thus consider only a limited time horizon. In such settings, long term effects, which may lead to differing outcomes are neglected. A well-known problem related to scheduling in queueing networks in a dynamic-stochastic context is the stability of the system. Stability implies that the long-



term average number of projects in the system is finite. Kumar and Seidman (1990) show that instability may occur for some scheduling policies leading to unbounded numbers of projects in the system over time. This may happen even if the utilization per resource is strictly less than one. Our experimental design ensures stability of the systems.

To the best of our knowledge, no large-scale systematic investigations of the effects of problem parameters on the performance of priority rules have been carried out. An exception holds for the preliminary works of Melchior and Kolisch (2009) and Melchior (2015). In this paper we undertake an elaborate computational study where we assess the impact of several problem parameters on the performance of priority rule based scheduling policies. We thereby extend the works of Melchior and Kolisch (2009) as well as Melchior (2015) in several ways: First, we consider additional priority rules such as switching rules, where one of them turns out to be clearly the best rule. Second, we assess the idea of embedding priority rules into project-based FCFS policies as proposed by Baccelli et al. (1993). Third, we find groups of rules with similar performance and distinguish between groups of rules with statistically different performance by using the Duncan test.

### 3 Priority rules

The selection of the rules was based on their performance for similar problems with total tardiness or total weighted tardiness objectives. Several of the priority rules to be discussed below employ the expected length of the critical path of a project and the expected latest start time of a project's activities needed to meet the project's due date. These parameters are stochastic and depend on the state of the system and future scheduling decisions. As described below, we derive these parameters employing a Monte Carlo simulation of the resource-unconstrained system. The notation employed in presenting the rules is summarized in Table 15 in Appendix A.

#### 3.1 Basic rule types for minimizing tardiness

In searching the literature of scheduling rules designed for minimizing total weighted tardiness, we see that all such rules, in some way or another, combine information about the input parameters due date, activity processing time, and weight. Looking closely at such rules, we see two fundamental ways in which this information is considered. At a very basic level, the first approach is to consider some ratio involving due dates and processing times. A second approach is to consider a time-sensitive binary switch in emphasis from due date to processing time, which gives rise to two basic families of rules: ratio rules and binary switch rules.

Probably the first evidence of the “ratio” approach is found in the work of Carroll (1965) and his so-called “c over t” dispatching rule which was inspirational for a series of works by Lawrence and Morton (1993) and Morton and Pentico (1993). For binary switch rules the seminal work is provided by Baker and Bertrand (1982) with their “modified due date” priority rule and later further developed by Baker and Kanet (1983), Dumond and Mabert (1988), Anderson and Nyirenda (1990) and Kanet and Li (2004). We provide below precise descriptions of how we adapted the ideas of these various authors to the DSRCMPSP environment.



### 3.2 Ratio rules

The ratio rules which we investigate include the family of bottleneck dynamic (BD) rules proposed by Lawrence and Morton (1993) as well as Morton and Pentico (1993). The generic form of the rules defines the priority of an activity  $(i, j) \in \mathcal{W}_r(t)$  at time  $t$  to be

$$\frac{w_j \cdot U_{ij}(t)}{\pi_{ij}(t)}. \quad (7)$$

The basic idea of bottleneck dynamic rules is to balance the cost  $w_j \cdot U_{ij}(t)$  of delaying activity  $(i, j)$  for one period against the marginal opportunity cost  $\pi_{ij}(t)$  associated with starting  $(i, j)$  one period earlier. The term  $w_j \cdot U_{ij}(t)$  for the delay cost has two components. The weight  $w_j$ , which reflects the maximum cost for increasing the tardiness of project  $j$  by one period and the urgency factor  $U_{ij}, 0 \leq U_{ij}(t) \leq 1$ , defined as

$$U_{ij}(t) = \exp \left[ \frac{-\max(0, (\bar{l}_{ij} - t))}{k \cdot \bar{d}_{r_{ij}}(t)} \right]. \quad (8)$$

In the numerator of Eq. (8),  $\bar{l}_{ij}$  is the latest start time of activity  $(i, j)$  and thus  $\bar{l}_{ij} - t$  is the slack of activity  $(i, j)$ . In the denominator,  $\bar{d}_{r_{ij}}(t)$  is the average expected duration of activities waiting in front of resource  $r_{ij}$  at time  $t$  where  $r_{ij}$  is the resource required by activity  $(i, j)$ . The factor  $k$  is a scaling factor denoted as the “lookahead parameter”. For nonpositive slack the urgency factor is 1 and it decreases exponentially to zero for positive slack.

To estimate the opportunity costs  $\pi_{ij}(t)$ , Lawrence and Morton (1993) propose “myopic activity costing” (MC) or “global activity costing” (GC). Myopic activity costing considers only activity  $(i, j)$  to be scheduled on resource  $r_{ij}$ , which yields opportunity cost

$$\pi_{ij}(t) = \frac{\pi_{r_{ij}}(t) \cdot \bar{d}_{ij}}{c_{r_{ij}}}. \quad (9)$$

The rationale for myopic activity costing is that scheduling activity  $(i, j)$  seizes fraction  $\frac{1}{c_{r_{ij}}}$  of the capacity from resource  $r_{ij}$  for  $\bar{d}_{ij}$  periods. Each period is worth  $\pi_{r_{ij}}(t)$ , which is the estimated opportunity cost for postponing the activities waiting in front of resource  $r_{ij}$  by one period. As all activities waiting for resource  $r$  only seize this resource, the fraction  $\frac{\pi_{r_{ij}}(t)}{c_{r_{ij}}}$  becomes  $\frac{\pi_r(t)}{c_r}$  and thus a constant term no longer relevant for the selection. Hence, by omitting the constant term, Eq. (9) reduces to  $\pi_{r_{ij}}(t) = \bar{d}_{ij}$ .

Global activity costing (GC) takes into account the opportunity costs of all unfinished activities of project  $j$  including activity  $(i, j)$  given in the set  $\mathcal{U}_j(t)$ . Morton and Pentico (1993) assume that to expedite project  $j$  all not yet finished activities  $(i', j)$  with  $i' \in \mathcal{U}_j(t)$  of project  $j$  have to be started immediately and consecutively. We thus obtain the opportunity cost

$$\pi_{ij}(t) = \sum_{m \in \mathcal{U}_j(t)} \frac{\pi_{r_{mj}}(t) \cdot \bar{d}_{mj}}{c_{r_{mj}}} \quad (10)$$

with resource prices  $\pi_r(t)$  set either by uniform resource pricing according to Eq. (11) given in Lawrence and Morton (1993) or by dynamic resource pricing according to Eq. (12) given in Morton and Pentico (1993).

$$\pi_r^U(t) = 1 \quad (11)$$

$$\pi_r^D(t) = \sum_{(i,j) \in \mathcal{W}_r(t)} w_j \cdot U_{ij}(t) \quad (12)$$

Combining the two activity costing methods, myopic and global, with the resource pricing schemes, uniform and dynamic, we obtain the following bottleneck dynamic (BD) priority rules.

### BD with Myopic Activity Costing (BD-MC)

$$\max_{(i,j) \in \mathcal{W}_r(t)} \left\{ \frac{w_j \cdot U_{ij}(t)}{\bar{d}_{ij}} \right\} \quad (13)$$

### BD with Global Activity Costing and Uniform Resource Pricing (BD-GC-U)

$$\max_{(i,j) \in \mathcal{W}_r(t)} \left\{ w_j \cdot U_{ij}(t) \cdot \left( \sum_{m \in \mathcal{U}_j(t)} \frac{\bar{d}_{mj} \cdot \pi_r^U(t)}{c_{r_{mj}}} \right)^{-1} \right\} \quad (14)$$

### BD with Global Activity Costing and Dynamic Resource Pricing (BD-GC-D)

$$\max_{(i,j) \in \mathcal{W}_r(t)} \left\{ w_j \cdot U_{ij}(t) \cdot \left( \sum_{m \in \mathcal{U}_j(t)} \frac{\bar{d}_{mj} \cdot \pi_r^D(t)}{c_{r_{mj}}} \right)^{-1} \right\} \quad (15)$$

The priority rules (13)–(15) are similar to the well-known weighted shortest processing time (WSPT) rule. A formal definition of the WSPT rule is given in Eq. (25). The BD-MC-rule multiplies for each waiting activity its WSPT-value with its urgency factor  $U_{ij}(t)$ . In this way, the project due date is taken into account. If all activities in the queue of resource  $r$  have nonpositive slack and hence their urgency factor  $U_{ij}(t) = 1$ , the BD-MC rule equals the WSPT rule. The two global activity costing rules BD-GC-U and BD-GC-D generalize the WSPT-rule by considering all activities of the entire project not yet processed.

## 3.3 Binary switch rules

The basic idea behind binary switch rules is to provide a priority value for an activity, in which emphasis switches (at a critical point in time) from the activity's due date to its processing time. The major rules falling into this category include the various forms of the “modified due date” approach first provided by Baker and Bertrand (1982). We study several variations of the modified due date approach applying it to the DSRCMPSP as outlined below.

**Weighted Modified Due Date (WMDD)** At decision time  $t$ , we select an activity according to

$$\min_{(i,j) \in \mathcal{W}_r(t)} \left\{ \frac{1}{w_j} \cdot \max \{ D_j - t, \bar{q}_{ij} \} \right\}. \quad (16)$$

In Eq. (16),  $D_j$  is the due date of project  $j$ ,  $\bar{q}_{ij}$  is the expected time for the activities along the critical path from activity  $(i, j)$  to the project completion, i.e., the tail of activity  $(i, j)$ . Note that WMDD gives an activity a priority at time  $t$  based on either its project's due date or the expected time of its project's earliest completion.

**Weighted Modified Activity Due Date (WMOD)** This rule selects activity  $(i, j)$  following

$$\min_{(i,j) \in \mathcal{W}_r(t)} \left\{ \frac{1}{w_j} \cdot \max \{ \bar{d}_{ij}, \bar{D}_{ij} + t \} \right\}. \quad (17)$$

The estimate of the activity's due date  $\bar{D}_{ij}$  is calculated as  $\bar{D}_{ij} = 0.5 \cdot (\bar{D}_{ij}^{\max} - \bar{D}_{ij}^{\min})$  with the maximum due date

$$\bar{D}_{ij}^{\max} = D_j - \bar{q}_{ij} + \bar{d}_{ij} \quad (18)$$

and the minimum due date

$$\bar{D}_{ij}^{\min} = a_j + \bar{r}_{ij} + \bar{d}_{ij} \quad (19)$$

where  $\bar{r}_{ij}$  refers to the expected time for the activities along the critical path from the start of the project to activity  $(i, j)$ , i.e., the head of the activity  $(i, j)$ .

### Weighted Critical Ratio and Shortest Processing Time (W(CR+SPT))

First developed by Anderson and Nyirenda (1990) and studied by Kutanoglu and Sabuncuoglu (1999) in the dynamic job shop setting, the CR+SPT rule is an interesting variation to the WMOD approach. The name CR+SPT is somewhat of a misnomer as it suggests a linear combination of CR and SPT, rather than the “either-or” logic construction of WMOD. The rule is applied at the activity level (like WMOD) and like WMOD it sets priority to the greater of an activity time till due or its processing time. The variation consists of dynamically changing an activity's due date (and thus its time till due) by multiplying it by the activity's critical ratio. The critical ratio  $CR_{ij}(t)$  is the ratio of time till project  $j$  is due to the remaining work content on the critical path to completion of  $j$ , i.e.,  $CR_{ij}(t) = (D_j - t) / \bar{q}_{ij}$  such that activities with larger  $CR_{ij}(t)$  are less critical. This idea is used to obtain a modified (expected) activity duration  $D'_{ij} = CR_{ij}(t) \cdot \bar{d}_{ij}$  that is inserted into the classical WSPT rule as follows

$$\min_{(i,j) \in \mathcal{W}_r(t)} \left\{ \frac{1}{w_j} \cdot \max \left\{ D'_{ij}, \bar{d}_{ij} \right\} \right\}. \quad (20)$$

For activities with low critical ratio, the activity durations are increased in this rule. If all activities have a critical ratio less than one, then the rule reduces to WSPT.

**Weighted Critical Ratio and Global Shortest Processing Time (W(CR+GSPT))** This rule extends W(CR+SPT) by considering the sum of the expected durations of all unfinished activities, instead of the expected duration of only the current activity  $(i, j)$ . With  $D_{ij}^{G'} = CR_{ij}(t) \cdot \sum_{m \in \mathcal{U}_j(t)} \bar{d}_{mj}$ , we obtain

$$\min_{(i,j) \in \mathcal{W}_r(t)} \left\{ \frac{1}{w_j} \cdot \max \left\{ D_{ij}^{G'}, \sum_{m \in \mathcal{U}_j(t)} \bar{d}_{mj} \right\} \right\}. \quad (21)$$

Following the distinction of the bottleneck dynamic rules of Morton and Pentico (1993) into local and global rules as discussed in Sect. 3.2, we categorize the rule W(CR+SPT) as local and the rule W(CR+GSPT) as global because the former considers only the expected duration of activity  $(i, j)$  while the latter considers the sum of the expected durations of all not finished activities, including activity  $(i, j)$ .

### Weighted Due Date Modified Shortest Activity from Shortest Project (WSASP-DD)

This rule represents a final variation of the binary switch approach. It was originally studied by Kurtulus and Davis (1982) in the job shop environment and later adapted by Kurtulus and Narula (1985) for the static deterministic resource-constrained multi-project case, and by Dumond and Mabert (1988) for the dynamic deterministic multi project case. We adapt it here for the DSRCMPSP by setting the activity's priority at time  $t$  to be

$$\min_{(i,j) \in \mathcal{W}_r(t)} \left\{ \begin{array}{ll} w_j \cdot (\bar{l}_{ij} - t) & \text{for } \bar{l}_{ij} - t < 0 \\ \frac{1}{w_j} \cdot (\bar{q}_{0j} + \bar{d}_{ij}) & \text{otherwise} \end{array} \right\}, \quad (22)$$

where  $\bar{l}_{ij} - t$  is the activity's slack at time  $t$ , and  $\bar{q}_{0j}$  is the tail of the dummy start activity  $(0, j)$  of project  $j$ , i.e., the expected length of the critical path of project  $j$ . As described by Dumond and Mabert (1988), SASP-DD is a refinement of the SASP of Kurtulus and Narula (1985), which combines the features of shortest activity time with minimum slack with the switch occurring when slack is depleted.

### 3.4 Benchmark rules

In addition to the rules outlined above, we investigate the following rules, which are commonly used for comparison in many of the studies cited above.

**First-Come, First-Serve (FCFS)** Here, we select activity  $(i, j)$  according to

$$\min_{(i,j) \in \mathcal{W}_r(t)} \{a_{ij}\}, \quad (23)$$

with  $a_{ij}$  denoting the arrival time of activity  $(i, j)$  at resource  $r_{ij}$ .

**Weighted Minimum Slack (WMINSLK)** Activity selection follows

$$\min_{(i,j) \in \mathcal{W}_r(t)} \left\{ \begin{array}{ll} w_{pj} \cdot (\bar{l}_{ij} - t) & \text{for } \bar{l}_{ij} - t < 0 \\ \frac{1}{w_{pj}} \cdot (\bar{l}_{ij} - t) & \text{otherwise} \end{array} \right\}. \quad (24)$$

Note that we modified WMINSLK such that for the case of negative slack, i.e.,  $\bar{l}_{ij} - t < 0$ , the priority value is still positive.

**Weighted Shortest Processing Time (WSPT)**

$$\min_{(i,j) \in \mathcal{W}_r(t)} \left\{ \frac{\bar{d}_{ij}}{w_j} \right\} \quad (25)$$

**Random (RAN)** This policy randomly selects activities for scheduling. We employ the RAN rule to serve as a baseline for rule comparisons.

Finally, we note that for all rules, ties are broken randomly.

## 4 Experimental design

We begin this section with a discussion of the specific assumptions for the simulation study. We then outline the relevant parameters for the generation of problem instances and the simulation.

### 4.1 Preliminaries

For each project type  $p \in \mathcal{P}$  we assume a Poisson arrival process with rate  $\lambda_p$ . Arriving projects are numbered in the order of arrival by assigning an index  $j$  irrespective of their type. Activity durations are assumed to be exponentially distributed with expected value  $\bar{d}_{ip}$ . The assumption of exponentially distributed interarrival times of projects and duration of activities is quite general and in line with the literature (see, e.g., Satic et al., 2022). An advantage of the exponential distribution is that only limited information is required and that the distributions are general and do not depend on a specific application or industry. Maximum allowed flow times  $A_j$  are drawn from the uniformly distributed interval  $[(1 - \beta) \cdot \bar{A}_p, (1 + \beta) \cdot \bar{A}_p]$ ,

**Table 1** Systematically altered levels of the system parameters

System parameter	Values
$ \mathcal{R} $	1, 3, 5, 10, 15, 20
$\rho$	0.7, 0.9
$CV_r^d$	[0, 0.4], [0.8, 1]

where  $\bar{A}_p$  denotes the mean maximum allowed flow time for project type  $p$ . Then, due dates  $D_j$  are derived from  $A_j$  by  $D_j = a_j + A_j$ . The number of projects in the system is restricted to  $N^{\max}$ ; otherwise the system might not reach steady state in the sense that the system exhibits unstable system behavior if non-optimal scheduling policies are used (see Kumar & Seidman 1990 as well as 2008).  $N^{\max}$  and  $\bar{A}_p$  for all  $p \in \mathcal{P}$  are determined by simulation runs in which activities are scheduled with the RAN policy. In the first run, we do not restrict the number of projects  $|J(t)|$  at any time  $t$ . In the next run, we set  $N^{\max}$  to the number of projects which is not exceeded for 99% of the time. As a consequence, the system is nearly an open system without limitation on the number of projects. In another RAN run, taking into account  $N^{\max}$  as determined above, we calculate  $\bar{A}_p$  such that the percentage of completed projects of type  $p \in \mathcal{P}$  with flow time  $C_j - a_j \leq \bar{A}_p$  equals  $1 - \alpha_p$ . Parameter  $\alpha_p$  measures the percentage of tardy projects of type  $p$  when using the RAN scheduling policy. We employ  $\alpha_p$  to take into account that the difficulty of meeting due dates depends on multiple factors, among others the utilization of resources (see Ramasesh, 1990). As  $\alpha_p$  is a statistical measure defined on the realizations of project flow times, it allows control of the due date tightness to be independent of other problem parameters. Furthermore, as RAN does not use any information, the due date tightness does not depend on specific scheduling policies. Finally, we discuss a normalization of the results. Instead of using the weighted tardiness  $Z(\omega)$  obtained by (priority) policy  $\omega$  for a problem instance, we normalize the results by applying

$$Z^n(\omega) = \frac{Z(\omega)}{Z(\omega_{\text{RAN}})} \quad (26)$$

where  $Z^n(\omega)$  is the average weighted tardiness obtained for a policy  $\omega$  relative to the average weighted tardiness obtained using the random policy  $\omega_{\text{RAN}}$ . In the following, we refer to  $Z^n(\omega)$  as normalized weighted tardiness. The normalization has the advantage that when averaging over the results from a set of problem instances, the values lie in a similar range such that they have a similar impact on the mean. Furthermore, the performance can be entirely explained by the information used by policy  $\omega$ .

## 4.2 Generation of problem instances

The generation of problem instances is controlled by two sets of parameters, system parameters and project type parameters.

**System parameters** are the number of resources  $|\mathcal{R}|$ , the system utilization  $\rho$ , and  $CV_r^d$ , the coefficient of variation of the expected durations of activities processed on resource  $r$ . The system parameters are systematically varied according to the parameter levels given in Table 1.

**Table 2** Systematically altered levels of the project type parameters

Project type parameter	Values
$\alpha$	80%, 20%
$\beta$	50%
$( \mathcal{V}_1 ,  \mathcal{V}_2 )$	(20, 20)
$(w_1, w_2)$	(1, 2)
OS	0, 0.2, 0.4, 0.6, 0.8, 1.0

For each number of resources  $|\mathcal{R}|$  we assume  $c_r = 1 \forall r \in \mathcal{R}$ , i.e., each resource  $r$  can process not more than one activity at a time. We assume identical utilization levels

$$\rho_r = \frac{\lambda \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{V}_{pr}} \bar{d}_{ip}}{|\mathcal{P}|} \quad (27)$$

for every resource  $r \in \mathcal{R}$  (such that  $\rho_r = \rho$  for all  $r \in \mathcal{R}$ ) with  $\mathcal{V}_{pr} = \{i \in \mathcal{V}_p | r_{ip} = r\}$  being the set of activity types of project type  $p$  that are processed by resource  $r \in \mathcal{R}$ . We attain a given utilization level  $\rho$  by controlling the total arrival rate  $\lambda = \lambda^{\max} \cdot \rho$  with  $\lambda^{\max} = 0.1333$ . Utilization  $\rho$  is set to 70% and 90%. For simplicity, the project type related arrival rates  $\lambda_p$  have equal shares of  $\lambda$  such that  $\lambda_p = \lambda/|\mathcal{P}|$  holds. The coefficient of variation of the expected durations of activities processed on resource  $r$  is  $\text{CV}_r^{\bar{d}}$ , where  $\text{CV}_r^{\bar{d}}$  is defined on  $\bar{d}_r$ , the expected duration of any activity arriving at resource  $r$ . For simplifying the computation, we assume that expected durations are independent and identically distributed (i.i.d.). Note that the i.i.d. assumption is normally not met due to the impact of scheduling policies and interdependencies between activities of the same project. For example, multiple activities from the same project may arrive on completion of their common predecessors. The expected duration of all activities processed on a resource  $r$  is given by

$$\bar{d}_r = \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{V}_{pr}} \frac{\lambda_p \cdot \bar{d}_{ip}}{\lambda_r} = \frac{\rho_r}{c_r} \quad (28)$$

in which

$$\lambda_r = \sum_{p \in \mathcal{P}} \lambda_p \cdot |\mathcal{V}_{pr}| \quad (29)$$

is the total arrival rate at resource  $r \in \mathcal{R}$ . Due to  $\rho_r = \rho$ ,  $\rho_r$  is controlled via the utilization. Finally,  $\text{CV}_r^{\bar{d}}$  is given by

$$\text{CV}_r^{\bar{d}} = \sqrt{\left( \lambda_r \sum_{p \in \mathcal{P}} \lambda_p \sum_{i \in \mathcal{V}_{pr}} \frac{\bar{d}_{ip}^2}{\left( \sum_{p \in \mathcal{P}} \sum_{i \in \mathcal{V}_{pr}} \bar{d}_{ip} \right)^2} \right) - 1}, \quad (30)$$

for the derivation see Melchior (2015). For our experiment, we require that  $\text{CV}_r^{\bar{d}} \in [\text{CV}_r^{\bar{d}, \min}, \text{CV}_r^{\bar{d}, \max}]$  for all  $r \in \mathcal{R}$ . The intervals are given in Table 1.

**Project type parameters** are due date tightness  $\alpha$ , the variation of the maximum flow time  $\beta$ , the number of activity types per project type  $|\mathcal{V}_p|$ , weights of project types  $w_p$ , and order strength OS. Table 2 summarizes how the project type parameters were systematically varied.

We investigate problem instances with two project types ( $\mathcal{P} = \{1, 2\}$ ) with weights  $w_1 = 1$  and  $w_2 = 2$ . As we want to obtain general insights into the effects of problem parameters,

**Table 3**  $|\mathcal{V}_{pr}|$  for all  $p \in \mathcal{P}$  and  $r \in \mathcal{R}$ 

$ \mathcal{R} $	$ \mathcal{V}_{pr} $
1	20
3	7, 7, 6
5	4, 4, 4, 4, 4
10	2, 2, 2, 2, 2, 2, 2, 2, 2, 2
15	2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
20	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

**Table 4** Sample specifications

2 Samples of $(r_{ip}, \bar{d}_{ip})$ for all $i \in \mathcal{V}_p$ and $p \in \{1, 2\}$
2 Network samples for each level of OS

we consider similar (but not identical) project types. For the following parameters the two projects types have identical parameter levels: Two levels of due date tightness  $\alpha = 20\%$  and  $80\%$ , one level for the variation of the maximum flow time  $\beta = 0.5$ , and one level for the number of activity types for each project type  $|\mathcal{V}_p| = 20$ . The network structure is individually controlled for each project type using the order strength (OS). OS is a  $[0, 1]$  normalized measure for the number of all precedence relationships, including the transitive ones. A value of 0 indicates a parallel network with no precedence relations between activities, while a value of 1 indicates a network with all activities in strict sequence. A formal definition of OS is given by Schwindt (1998). We set  $OS = 0, 0.2, 0.4, 0.6, 0.8$ , and 1 for both project types. For each level of OS, except 0 and 1, two pairs of non-identical networks are sampled, one for each project type. For the extreme values of  $OS = 0$  and  $OS = 1$ , only one network exists for each value. Two sets of tuples  $(r_{ip}, \bar{d}_{ip})$  for all  $i \in \mathcal{V}_p$  and  $p \in \mathcal{P}$  are randomly sampled. For our design, we have sampled two sets in order to cover the range of possible sets and to avoid that conclusions are made based on specific properties of a single set. For each activity type  $(i, p)$  we sample  $\bar{d}_{ip}$  taking into account  $CV^{\bar{d}}$ . When determining the resource  $r_{ip}$  where activity type  $(i, p)$  has to be processed, information on the number of activity types  $|\mathcal{V}_{pr}|$  to be processed at each resource  $r \in \mathcal{R}$  is required. In our design, each resource should roughly process the same number of activity types. Thus, for fixed  $|\mathcal{V}_p| = 20$  for all  $p \in \mathcal{P}$ , the values for  $|\mathcal{V}_{pr}|$  depend on  $|\mathcal{R}|$  (see Table 3). Table 4 gives the specification of the samples.

The total number of problem instances results from the product over the number of parameter levels (given in parentheses) with respect to  $|\mathcal{R}|$  (6),  $\rho$  (2),  $CV^{\bar{d}}$  (2),  $\alpha$  (2), OS (6), as well as the number of sampled pairs for each level of OS (2) and the number of the sets of tuples  $(r_{ip}, \bar{d}_{ip})$  (2), such that we obtain 1,152 instances. For each problem instance we simulate each of the 12 priority policies (11 rules and RAN as well as two runs for  $N^{\max}$  and the quantile for  $\alpha_p$  respectively) in a simulation run which results in 16,128 runs.

### 4.3 Simulation set up

For the simulation of the problem instances we have programmed a discrete event simulation tool in Java. When running the simulation, we employed a warm-up period of 10,000 projects to reach steady state. The length of the warm-up period was determined using Welch's procedure (see Law, 2007) for problem instances with  $|\mathcal{R}| = 20$  and  $\rho = 0.9$  in which



a high number of projects is expected to be in the system. In this way, a longer warm up period than necessary is obtained for most instances. The length of the observation period was set to 20,000 projects. As a variance reduction technique, we employed common random numbers (see Law, 2007) such that for each problem instance all priority policies are applied on the same sample of projects with respect to type, interarrival times, and activity durations. For each problem instance and priority policy we ran 10 replications such that the width of the confidence interval of the objective function is at most 10%. For the BD-policies, the look-ahead parameter was set to  $k = 1$ , as for this value good results have been obtained in preliminary experiments.

## 5 Experimental results

In the following we provide the results of the simulation study. The performance of a rule is measured as the normalized weighted tardiness  $Z^n(\omega)$  given in Eq. (26). Duncan tests with a 5% significance level were undertaken to detect groups of rules where rules in different groups show a significant difference in performance while rules in the same group do not. First, in Sect. 5.1 we provide the results for the base case instances. Afterwards, in Sect. 5.2, we discuss the results for variations of the base case.

### 5.1 Results for the base case instances

Table 5 provides the performance of all priority rules as well as the group, which a rule belongs to, according to the Duncan test. Next to the 11 rules presented in Sect. 3, we consider the random rule RAN, which by definition has a normalized performance of  $\bar{Z}^n(\text{RAN}) = 1$ . The groups as obtained from the Duncan test inform us for any priority rule if the rule has a significant different performance than RAN. Obviously, priority rules, which are significantly worse than RAN, should not be selected. To facilitate reading the tables, we have highlighted in this as well as the following tables selected rules with the colors used in Table 5, i.e., W(CR+SPT) in yellow, BD-GC-D in green, BD-MC in gray-blue, BD-GC-U in turquoise, and W(CR+GSPT) in orange. W(CR+SPT) is the best rule, being significantly superior to all other rules. This extends the findings of Kutanoglu and Sabuncuoglu (1999) for the dynamic job shop problem. BD-GC-D, BD-MC, and WMOD rank second. The global counterparts of the local rules, namely W(CR+GSPT) and BD-GC-U, show a significantly worse performance than the local versions W(CR+SPT), BD-GC-D, and BD-MC. This confirms observations made by Kanet and Hayya (1982) and Kutanoglu and Sabuncuoglu (1999) for the job shop problem. The rule WMDD is significantly inferior to the random rule RAN.

In Sects. 5.1.1–5.1.5 we discuss the effects of the problem parameters  $\alpha$ ,  $|\mathcal{R}|$ , OS,  $\text{CV}_r^d$ , and  $\rho$  on the performance of the priority rules for the base case instances. For the analysis we consider the overall effect obtained from averaging over subsets of problem instances.

#### 5.1.1 Due date tightness

Table 6 shows the impact of due date tightness  $\alpha$  defined as the percentage of tardy projects obtained in the simulation study when using the RAN rule. The two levels are  $\alpha = 0.2$  (low due date tightness) with 20% tardy projects and  $\alpha = 0.8$  (high due date tightness) with 80% tardy projects. It turns out that overall good rules are good for both levels of due date tightness. W(CR+SPT) is the best rule for low due date tightness and together with

**Table 5** Overall performance of the priority rules and groups according to the Duncan test

Rule	$\bar{Z}^n(\omega)$	Groups
W(CR+SPT)	0.31	g
BD-GC-D	0.44	f
BC-MC	0.46	f
WMOD	0.51	f
WMINSLK	0.62	e
FCFS	0.63	e
WSASP-DD	0.81	d
BD-GC-U	0.95	c
WSPT	0.98	c
RAN	1.00	b, c
W(CR+GSPT)	1.07	a, b
WMDD	1.11	a

**Table 6** Performances of the priority rules for different due date tightness

$\alpha = 0.2$			$\alpha = 0.8$		
Rule	$\bar{Z}^n(\omega)$	Groups	Rule	$\bar{Z}^n(\omega)$	Groups
W(CR+SPT)	0.12	g	W(CR+SPT)	0.49	g
BD-MC	0.31	f	BD-GC-D	0.54	f
BD-GC-D	0.34	f	BD-MC	0.61	e
WMOD	0.40	e, f	WMOD	0.61	e
FCFS	0.47	e, f	WMINSLK	0.70	d
WMINSLK	0.54	e	WSPT	0.71	d
BD-GC-U	0.88	d	WSASP-DD	0.72	d
WSASP-DD	0.91	c, d	FCFS	0.80	c
WMDD	1.07	b, c	BD-GC-U	1.02	b
W(CR+GSPT)	1.11	a, b	W(CR+GSPT)	1.03	b
WSPT	1.25	a	WMDD	1.15	a

BD-GC-D best for high due date tightness. Note for the three rules WSPT, W(CR+GSPT), and WSASP-DD, the standardized performance improves with higher due date tightness. The performance of FCFS deteriorates with increasing due date tightness. This was to be expected, as giving priority to activities that arrived early is not a meaningful strategy when the majority of projects are tardy.

### 5.1.2 Number of resources

Figure 1 as well as Table 7 show the performance of the rules when increasing the number of resources from 1 to 20. For a clearer presentation, we separate the priority rules into two figures according to their performance: The left part of Fig. 1 provides priority rules, which show deteriorating performance for an increasing number of resources. The right part of Fig. 1 shows priority rules, which are not significantly impacted by the number of resources. The rules depicted in the left part of Fig. 1, namely W(CR+GSPT), WMDD, WMINSLK, WSASP-DD, BD-GC-U, and BD-GC-D do all use due date information. Furthermore, all global rules are included in this set. For these rules we observe for up to 5 resources a small deterioration of performance and small differences between the rules, but for more than 5

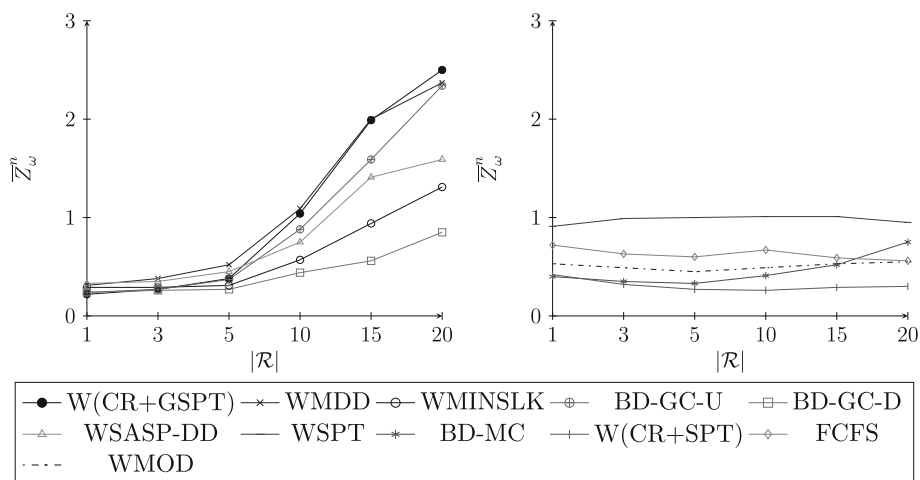
Fig. 1 Impact of  $|\mathcal{R}|$  on the performance of priority rules

Table 7 Impact of the number of resources on the performance

$ \mathcal{R}  = 1$			$ \mathcal{R}  = 3$			$ \mathcal{R}  = 5$		
Rule	$\bar{Z}^n(\omega)$	Groups	Rule	$\bar{Z}^n(\omega)$	Groups	Rule	$\bar{Z}^n(\omega)$	Groups
W(CR+GSPT)	0.22	g	BD-GC-D	0.26	g	BD-GC-D	0.27	g
BD-GC-U	0.24	g	W(CR+GSPT)	0.27	f, g	W(CR+GSPT)	0.27	f, g
BD-GC-D	0.24	g	BD-GC-U	0.27	f, g	WMINSLK	0.31	f, g
WMINSLK	0.29	f	WMINSLK	0.30	e, f	BD-MC	0.33	e, f
WMDD	0.31	e, f	W(CR+SPT)	0.32	e	BD-GC-U	0.37	e
WSASP-DD	0.33	e	BD-MC	0.35	d	W(CR+GSPT)	0.38	e
BD-MC	0.40	d	WSASP-DD	0.36	d	WSASP-DD	0.45	d
W(CR+SPT)	0.42	d	WMDD	0.38	d	WMOD	0.45	d
WMOD	0.53	c	WMOD	0.49	c	WMDD	0.52	c
FCFS	0.72	b	FCFS	0.64	b	FCFS	0.60	b
WSPT	0.91	a	WSPT	0.99	a	WSPT	1.00	a
$ \mathcal{R}  = 10$			$ \mathcal{R}  = 15$			$ \mathcal{R}  = 20$		
Rule	$\bar{Z}^n(\omega)$	Groups	Rule	$\bar{Z}^n(\omega)$	Groups	Rule	$\bar{Z}^n(\omega)$	Groups
W(CR+SPT)	0.26	g	W(CR+SPT)	0.29	e	W(CR+SPT)	0.30	e
BD-MC	0.41	f	BD-MC	0.52	e	WMOD	0.55	d, e
BD-GC-D	0.44	f	WMOD	0.53	e	FCFS	0.56	d, e
WMOD	0.49	e, f	BD-GC-D	0.56	d, e	BD-MC	0.75	c, d
WMINSLK	0.57	d, e	FCFS	0.59	c, d, e	BD-GC-D	0.85	c, d
FCFS	0.67	c, d	WMINSLK	0.94	c, d	WSPT	0.95	c
WSASP-DD	0.75	c	WSPT	1.01	c	WMINSLK	1.31	b
BD-GC-U	0.88	b	WSASP-DD	1.41	b	WSASP-DD	1.59	b
WSPT	1.01	a	BD-GC-U	1.59	a, b	BD-GC-U	2.34	a
W(CR+GSPT)	1.04	a	W(CR+GSPT)	1.99	a	WMDD	2.37	a
WMDD	1.09	a	WMDD	2.00	a	W(CR+GSPT)	2.50	a

resources a high deterioration of performance and large differences between the rules. In Table 7 we see that for a small number of resources, i.e., 1 and 3 resources, BD-GC-D, BD-GC-U, and W(CR+GSPT) provide the best performance, showing no significant difference according to the Duncan test. For the case of 5 resources, BD-GC-D is still in the best group, now joined by W(CR+SPT). W(CR+SPT) stays in the best group for more than 5 resources. The better performance of W(CR+GSPT) compared to W(CR+SPT) for a small number of resources and loose due dates can be explained by Theorem B.1 given in Appendix B. In general we observe that global-oriented rules perform better for a low number of resources whereas local-oriented rules excel for high number of resources.

**Table 8** Performance of the priority rules for different levels of order strength

OS=0			OS=0.2			OS=0.4		
Rule	$Z^n(\omega)$	Groups	Rule	$Z^n(\omega)$	Groups	Rule	$Z^n(\omega)$	Groups
W(CR+GSPT)	0.29	f	BD-GC-D	0.26	f	BD-GC-D	0.34	h
BD-GC-D	0.29	f	W(CR+SPT)	0.30	e, f	W(CR+SPT)	0.38	g, h
BD-GC-U	0.29	f	BD-MC	0.33	e	BD-MC	0.41	f, g
WMINSLK	0.35	e	WMINSLK	0.34	e	WMINSLK	0.45	e, f
FCFS	0.36	d, e	WMOD	0.44	d	FCFS	0.50	d, e
WSASP-DD	0.37	d	WSASP-DD	0.45	d	WMOD	0.50	d, e
BD-MC	0.43	c	BD-GC-U	0.48	c, d	WSASP-DD	0.53	d
WMDD	0.43	c	W(CR+GSPT)	0.52	b, c	WMDD	0.59	c
W(CR+SPT)	0.43	c	WMDD	0.55	b	BD-GC-U	0.61	c
WMOD	0.54	b	FCFS	0.55	b	W(CR+GSPT)	0.66	b
WSPT	0.76	a	WSPT	0.85	a	WSPT	0.80	a

OS=0.6			OS=0.8			OS=1.0		
Rule	$Z^n(\omega)$	Groups	Rule	$Z^n(\omega)$	Groups	Rule	$Z^n(\omega)$	Groups
W(CR+SPT)	0.30	f	W(CR+SPT)	0.22	f	W(CR+SPT)	0.23	e
BD-GC-D	0.37	e, f	BD-MC	0.43	e	WMOD	0.65	d, e
BD-MC	0.39	e, f	WMOD	0.43	e	BD-MC	0.77	d
WMOD	0.47	d, e	BD-GC-D	0.45	d, e	FCFS	0.88	d
WMINSLK	0.52	d	WMINSLK	0.65	c, d	BD-GC-D	0.92	c, d
FCFS	0.65	c	FCFS	0.85	b, c	WSPT	1.38	c
WSASP-DD	0.67	c	WSASP-DD	0.93	b	WMINSLK	1.41	c
BD-GC-U	0.85	b	WSPT	1.20	a	WSASP-DD	1.93	b
WSPT	0.88	b	BD-GC-U	1.24	a	BD-GC-U	2.23	a, b
W(CR+GSPT)	1.04	a	W(CR+GSPT)	1.32	a	WMDD	2.59	a
WMDD	1.12	a	WMDD	1.38	a	W(CR+GSPT)	2.59	a

### 5.1.3 Order strength

Table 8 provides the performance of the priority rules for order strength (OS) levels between 0 (purely parallel activity networks) and 1 (purely serial activity networks). For OS-levels between 0 and 0.6, BD-GC-D is in the best group while for OS-levels between 0.2 and 1, W(CR+SPT) is in the best group. For the special case of purely parallel networks with  $OS = 0$ , W(CR+SPT) shows a low performance and thus is not a good choice whereas its global counterpart W(CR+GSPT) is in the best group. However, the performance of W(CR+GSPT) deteriorates rapidly for OS-levels larger than 0. The good performance of W(CR+GSPT) for  $OS = 0$  is due the fact that for instances with  $OS = 0$  there are no precedence constraints between activities. Hence, the start times of activities and the finish time of projects are only constrained by the scarce resources. In this case, giving priority to activities from projects with low total remaining work load is a meaningful strategy to minimize total weighted tardiness.

### 5.1.4 Variation of expected activity durations

Table 9 gives the results for the two ranges of the coefficient of variation of the average activity durations  $CV^d$ . Again, W(CR+SPT) excels, being in the best group for both levels of variation. Table 9 underlines the value of WSPT-information when the variance of activity durations is high. While being in the group of worst performing rules for a low level of variation, the performance of WSPT improves significantly for a high level of variation. This improvement can be explained by the fact that minimizing the average weighted flow time of the activities at each resource becomes more effective when the difference in the duration of the activities becomes more pronounced.

### 5.1.5 Utilization per resource

Table 10 presents the effect of the utilization per resource on the performance of the priority rules. For each level of resource utilization, W(CR+SPT) dominates, while BD-GC-D and BD-MC belong to the second best group. In going from utilization of 0.7–0.9, the performance spread between the best and the worst rule jumps from 50.19 to 110.17. This jump

**Table 9** Performance of the priority for different levels of the coefficient of variation of activity durations

$CV^d \in [0; 0.4]$			$CV^d \in [0.8; 1.0]$		
Rule	$\bar{Z}^n(\omega)$	Groups	Rule	$\bar{Z}^n(\omega)$	Groups
BD-GC-D	0.35	h	W(CR+SPT)	0.24	h
W(CR+SPT)	0.38	gh	WMOD	0.38	g, h
WMINSLK	0.39	g	BD-MC	0.48	f, g
BD-MC	0.43	f	BD-GC-D	0.52	e, f, g
BD-GC-U	0.47	e	WSPT	0.62	ef
W(CR+GSPT)	0.50	d, e	FCFS	0.67	e
WSASP-DD	0.52	d	WMINSLK	0.85	d
FCFS	0.59	c	WSASP-DD	1.11	c
WMDD	0.60	c	BD-GC-U	1.43	b
WMOD	0.64	b	WMDD	1.62	a
WSPT	1.34	a	W(CR+GSPT)	1.63	a

**Table 10** Effect of the resource utilization on the performance of priority rules

0.7			0.9		
Rule	$\bar{Z}^n(\omega)$	Groups	Rule	$\bar{Z}^n(\omega)$	Groups
W(CR+SPT)	0.37	g	W(CR+SPT)	0.24	f
BD-GC-D	0.44	f	BD-GC-D	0.43	e
BD-MC	0.48	e, f	BD-MC	0.44	e
WMOD	0.51	e	WMOD	0.50	e
WMINSLK	0.56	d	FCFS	0.58	d, e
WSASP-DD	0.66	c	WMINSLK	0.68	d
FCFS	0.68	c	WSASP-DD	0.97	c
WSPT	0.76	b	BD-GC-U	1.13	b, c
BD-GC-U	0.76	b	WSPT	1.20	a, b
W(CR+GSPT)	0.83	a	W(CR+GSPT)	1.31	a
WMDD	0.87	a	WMDD	1.35	a

can be explained by growing queue lengths for increased resource utilization, which amplify the effect of the priority rules. Our findings contrast those in the literature. There, the performance of flow time related priority rules such as WSPT (see Kutanoğlu & Sabuncuoğlu 1999 or Vepsäläinen & Morton 1987) improves for high utilizations. The explanation lies in the different definition of the problem parameters and the related design of the computational studies. In the literature, due dates are set without taking into account the resource utilization. In contrast, in this study the due date tightness is controlled by the statistical measure percentage of tardy projects  $\alpha$ . In this way, we implicitly take the resource utilization into account when setting the mean maximum duration  $\bar{D}_p^{\max}$  for the projects. As a consequence, the due date tightness is not impacted by the utilization  $\rho$ .

## 5.2 Variations of the base case

In order to investigate the performance of the rules for settings different to the base case, we considered three different setups: First, the case of a single project type; second, the case of two project types differing in their weights and due date tightness; third, the case of two project types differing in their weights and arrival rates.

**Table 11** Overall performance of the priority rules and group according to the Duncan test for the single project type case

Rule	$\bar{Z}^n(\omega)$	Groups
W(CR+SPT)	0.46	g
WSASP-DD	0.48	f
WMINSLK	0.49	f
BD-GC-D	0.49	f
BD-GC-U	0.49	c
W(CR+GSPT)	0.50	f
BC-MC	0.54	e
FCFS	0.59	d
WMOD	0.66	c
WMDD	0.65	c
RAN	1.00	b
WSPT	1.02	a

### 5.2.1 Single project type

In this experiment we considered a single project type, in contrast to two project types as in the base case. Table 11 provides the results.

The following insights can be obtained: First, the range between the normalized performance of the best and the worst rule  $\max_{\omega}\{\bar{Z}^n(\omega)\} - \min_{\omega}\{\bar{Z}^n(\omega)\}$  decreases from 0.79 to 0.56, meaning that there is less variance between the performance of the rules. This is intuitive as problems with a single project type leave less room for good rules to excel. Second, W(CR+SPT) is still the best performing rule, significantly better than any other rule. Third, the order of performance of the colored rules is quite stable; only priority rule BD-MC deteriorates from the third to the fifth rank. Fourth, the global rules BD-GC-U and W(CR+GSPT) improve in normalized performance from close to one to close to 0.5. These two rules give priority to projects with a small amount of remaining work. In a single project type setting this approach gives rise to better results than in settings with multiple types of projects.

### 5.2.2 Project type differentiation according to due date tightness

We consider two types of projects. Project type 1 with low due date tightness  $\alpha_1 = 20\%$  and low weight  $w_1 = 1$  and project type 2 with high due date tightness  $\alpha_2 = 80\%$  and high weight  $w_2 = 2$ . Table 12 shows the results for this setting.

From the results we can obtain the following insights. First, the performance of all rules except FCFS improves compared to the random rule. In fact, all rules except WSPT are now significantly better than the random rule RAN. Second, W(CR+SPT) is still the best performing rule at the 5% level of significance, according to the Duncan test. Third, the order in performance of the colored rules is not changed and the order of all rules tested is quite stable.

### 5.2.3 Project type differentiation according to arrival rate

We consider two types of projects. Project type 1 with a share of 30% of the total arrival rate, i.e.,  $\lambda_1 = 0.3 \cdot \lambda$ , and low weight  $w_1 = 1$  and project type 2 with a share of 70% of the total arrival rate, i.e.,  $\lambda_1 = 0.7 \cdot \lambda$ , and high weight  $w_2 = 2$ . Table 13 gives the results for this setting. As for the case of different levels of due date tightness, the performance of all rules

**Table 12** Overall performance of the priority rules and group according to the Duncan test for the case with different due date tightnesses

Rule	$\overline{Z}^n(\omega)$	Groups
W(CR+SPT)	0.28	h
BD-GC-D	0.35	g
BC-MC	0.38	f
WMOD	0.42	f
WMINSLK	0.48	e
WSASP-DD	0.51	e
WSPT	0.59	d
BD-GC-U	0.74	c
FCFS	0.76	c
W(CR+GSPT)	0.50	bc
WMDD	0.82	b
RAN	1.00	a

**Table 13** Overall performance of the priority rules and group according to the Duncan test for different arrival rates

Rule	$\overline{Z}^n(\omega)$	Groups
W(CR+SPT)	0.25	i
BD-GC-D	0.28	h
W(CR+GSPT)	0.32	g
BC-MC	0.32	g
BD-GC-U	0.33	g
WMINSLK	0.36	f
WMDD	0.38	e
WMOD	0.39	e
WSASP-DD	0.42	d
FCFS	0.61	c
WSPT	0.78	b
RAN	1.00	a

when compared to the random rule improves, with BD-GC-U and W(CR+GSPT) improving considerably. In fact, all rules are now significantly better than the random rule RAN. Again, W(CR+SPT) is significantly the best rule. The order of the colored rules is not changed with the exception of W(CR+SPT).

## 6 Results for FCFS-policies

In the following we investigate the potential of First-Come, First-Serve (FCFS) policies first proposed by Baccelli et al. (1993) for the case of multi-project scheduling with a single type of project and general distributed i.i.d. interarrival times and activity durations. At first, we provide a definition extended to multiple project types.

**Definition 6.1** Given two activities  $(i, j_1)$  and  $(i, j_2)$  of the two projects  $j_1$  and  $j_2$  of the same type  $p$ , where project  $j_1$  has arrived before project  $j_2$ , then a First-Come, First-Serve (FCFS) policy always schedules activity  $(i, j_1)$  before  $(i, j_2)$ .



**Table 14** Performance comparison of FCFS-policies and Non-FCFS-policies

Rule	$\bar{Z}^n(\omega)$	$\bar{Z}^n(\omega, \text{FCFS})$	Improvement(%)	Significance
W(CR+SPT)	0.31	0.30	3.84	*
BD-GC-D	0.44	0.44	0.09	
BD-MC	0.46	0.45	2.14	*
WMOD	0.51	0.44	12.84	*
WMINSLK	0.62	0.62	0.00	
FCFS	0.63	0.63	0.01	
WSASP-DD	0.81	0.80	1.69	*
BD-GC-U	0.95	0.95	0.37	
WSPT	0.98	0.80	18.60	*
RAN	1.00	0.74	25.66	*
W(CR+GSPT)	1.07	1.08	-0.75	
WMDD	1.11	1.10	1.27	

Baccelli et al. (1993) show that the class of FCFS-policies contains an optimal policy for the multi-project scheduling problem with a single project type when minimizing the expected project flow time. Thus, when due dates are tight, we can expect the FCFS-policy to provide good results for the weighted tardiness objective. In our experiment we compare the performance of the priority rules presented above with and without embedding them into the FCFS-policy scheme. This is done as follows: Instead of having one queue in front of each resource  $r \in \mathcal{R}$ , we have a dedicated queue for each activity type  $i \in \mathcal{V}_p$  of each project  $p \in \mathcal{P}$ , which has to be processed on resource  $r$ , i.e., for which  $r_{i,p} = r$  holds. Within queue  $(i, p)$  all activities are of type  $(i, p)$  but belong to different projects. These activities are sorted according to increasing arrival times  $a_i$  of their projects, i.e., the activity  $(i, j)$  with the earliest project arrival time  $a_j$  is at the head of the queue. When one unit of resource  $r$  becomes available at time  $t$ , one of the priority rules presented in Sect. 3 is applied to determine the activity queue  $(i, p)$ , from which the activity at the head of the queue is started at  $t$ .

Table 14 provides the performance of FCFS-policies relative to the non-FCFS-policies where the rules are sorted according to decreasing performance for the non-FCFS-policy. Differing from the tables above, we added the RAN rule with a performance of 100% in the non-FCFS-setting. In the last column, we indicate a significantly better result for the FCFS-policy on the 95% level of confidence using the Duncan-test. The results show that FCFS-policies provide better results than non-FCFS-policies. The relative ranking remains approximately the same. The improvement when using a rule with FCFS-policy depends on the rules used. The largest improvement is obtained with the RAN rule followed by WSPT. For the W(CR+GSPT) rule, using the FCFS-policy leads to a slight deterioration. The best rule W(CR+SPT) benefits through the FCFS-policy by 3.84%. Obviously, policies not based on the GSPT idea and those that do not imitate the idea such as WMINSLK benefit most from the FCFS.

## 7 Conclusions and future research

The main take-away of this study is that the W(CR+SPT) rule is consistently the best rule or in the group of the best performing rules for minimizing average weighted project tardiness. This holds for the base case study undertaken in Sect. 5.1 as well as for the variation of

the base case in Sect. 5.2. Only for some very special settings in the base case study, i.e., few resource types or purely parallel project networks, does this not hold; in these cases W(CR+GSPT), the global variant of W(CR+SPT), is best. The rule W(CR+SPT) is simple regarding the required parameters, easy to implement, and straightforward to communicate to practitioners.

Another important finding is that our result differ from several of those found in the literature. There, the performance of flow time related priority rules such as WSPT (see Kutanoglu & Sabuncuoglu 1999 or Vepsalainen & Morton 1987) improve for high utilizations. The explanation lies in the different definition of the problem parameters and the related design of the computational studies. In the literature due dates are set without taking into account the resource utilization or other problem parameters. In contrast, in this study the due date tightness is controlled by the statistical measure percentage of tardy projects  $\alpha$ . As a consequence, the due date tightness is not impacted by the utilization  $\rho$ .

Further research can be undertaken in several directions. First, a study could be conducted with different type of distribution functions for the activity durations. Such studies have been undertaken for the single static resource-constrained project scheduling problem with makespan objective (see, e.g., Ballestin, 2007) and they show that priority rules perform better for problems with activity duration distributions, which have a smaller coefficient of variation than the exponential distribution. Second, the study at hand could be extended for more project types and larger projects. Third, the performance could be assessed for policies, which do not have the non-delay or work-conserving property. Finally, our analysis shows that the difference in performance of priority rules increases remarkably with the number of resources. Since studies for the dynamic job shop problem have not investigated instances with more than 10 resources, this could be a further fruitful area of research.

**Funding** Open Access funding enabled and organized by Projekt DEAL. This research was partially funded by the August-Wilhelm Scheer Visiting Professorship Program of the Technical University of Munich by financing the research stay of John Kanet at the Technical University of Munich.

## Declarations

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Conflict of interest** Author Philipp Melchior declares that he has no conflict of interest. Author Rainer Kolisch declares that he has no conflict of interest. Author John J. Kanet declares that he has no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## A Table of notation

See the Table 15.

**Table 15** Table of notation

Symbol	Explanation
$i$	Activity index
$j$	Project index
$J$	Number of projects arrived
$p$	Project type index
$t$	Time
$a_j$	Arrival time of project $j$
$\alpha$	Percentage of tardy projects as global parameter for due date tightness
$\alpha_p$	Percentage of tardy projects of type $p$
$A_j$	Maximum allowed flow time for project $j$
$\bar{A}_p$	Mean maximum allowed flow time for project type $p$
$\mathcal{A}_p$	Arc set for precedence relations of project type $p$
$\mathcal{E}_r(t)$	Set of activities being in process at resource $r$ at time $t$
$\mathcal{P}$	Set of project types
$\mathcal{R}$	Set of resources
$\mathcal{V}_j$	Index set for the activities of project $j$
$\mathcal{V}_p$	Index set for the activities of project type $p$
$\mathcal{W}_r(t)$	Set of activities waiting at resource $r$ at time $t$
$\beta$	Variation parameter for the maximum flow times around the mean maximum flow time
$c_r$	Number of capacity units of resource $r$
$C_{ij}$	Completion time of activity $i$ of project $j$
$CR_{ij}(t)$	Critical ratio of activity $i$ of project $j$
$CV^{\bar{d}}$	Coefficient of variation of the expected activity durations as global parameter
$CV_r^{\bar{d}}$	Coefficient of variation of the expected durations of the activities processed by resource $r$
$\bar{D}_{ij}^{\max}$	Upper estimate for activity due date in the WMOD rule based on the project due date
$\bar{D}_{ij}^{\min}$	Lower estimate for activity due date in the WMOD rule based on the project arrival time
$\bar{D}_{ij}$	Activity due date of the WMOD rule as the mean of the upper and the lower estimate
$\bar{d}_r(t)$	Mean of expected duration of the activities waiting at resource $r$ at time $t$
$D_j$	Due date of project $j$
$D'_{ij}$	Critical ratio modified expected activity duration
$D^{G'}_{ij}$	Critical ratio modified expected duration of the unfinished project activities
$d_{ij}$	Duration (ex-post) of activity $i$ of project $j$
$\bar{d}_{ij}$	Expected duration of activity $i$ of project $j$
$\bar{l}_{ij}$	Expected latest start time of activity $i$ of project $j$ without project $j$ being delayed
$\lambda^{\max}$	Maximum arrival rate
$\lambda$	Overall arrival rate of projects at the system
$\lambda_p$	Arrival rate for the projects of type $p$

**Table 15** continued

Symbol	Explanation
$\lambda_r$	Arrival rate of the activities queued in front of resource $r$
$N^{\max}$	Maximum number of projects in the system
OS	Order strength as global parameter
$OS_p$	Order strength of project type $p$
$\pi_{ij}(t)$	Opportunity cost of scheduling of activity $i$ of project $j$ at time $t$
$\pi_r(t)$	Price of resource $r$ at time $t$
$\pi_r^U(t)$	Price of resource $r$ at time $t$ calculated using uniform resource pricing
$\pi_r^D(t)$	Price of resource $r$ at time $t$ calculated using dynamic resource pricing
$\bar{q}_{ij}$	Expected time for the activities along the critical path from activity $(i, j)$ to the end of project $j$
$r_{ij}$	Resource required for activity $i$ of project $j$
$r_{ip}$	Resource required for activity $i$ of project type $p$
$\bar{r}_{ij}$	Expected time for the activities along the critical path from the start of project $j$ to activity $(i, j)$
$\rho$	Resource utilization as global parameter
$\rho_r$	Utilization of resource $r$
$U_{ij}(t)$	Urgency factor of activity $i$ project $j$ at time $t$
$\mathcal{U}_j(t)$	Set of not finished activities of project $j$
$w_j$	Weight of project $j$
$Z$	Average weighted tardiness
$Z(\omega)$	Average weighted tardiness when using policy $\omega$ for scheduling
$Z^n(\omega)$	Normalized average weighted tardiness when using policy $\omega$ for scheduling

## B Results for the single resource case

For the single resource case, each activity within each project has to be processed by the single resource with capacity 1. As due dates get tighter, the weighted tardiness objective approaches weighted flow time as given by

$$\lim_{J \rightarrow \infty} \left[ \sum_{j=1}^J w_j \cdot (C_j - a_j) \right] \quad (31)$$

where  $C_j$  denotes the finish time of project  $j$ .

**Theorem B.1** *The optimal scheduling policy minimizing the long term expected weighted project flow time is a priority policy where an activity is selected according to*

$$\max_{(i,j) \in \mathcal{W}_r(t)} \left\{ \frac{w_j}{\sum_{m \in \mathcal{U}_j} \bar{d}_{mj}} \right\} \quad (32)$$

and ties are broken arbitrarily. The policy can be seen as a global WSPT rule for selecting activities. Instead of the expected duration of an activity, it takes into account the total expected workload of all unfinished activities of project  $j$  from the set  $\mathcal{U}_j(t)$  including activity  $(i, j)$ .

For the proof see Melchioris (2015).

Thus, in particular when due dates are tight, global WSPT-based rules (BD-GC-U, BD-GC-D and W(CR+GSPT)) are near optimal policies, for the single resource case.

## References

- Adler, P. S., Mandelbaum, A., Nguyen, V., & Schwerer, E. (1995). From project to process management: An empirically-based framework for analyzing product development time. *Management Science*, 41(3), 458–484.
- Anavi-Isakow, S., & Golany, B. (2003). Managing multi-project environments through constant work-in-process. *International Journal of Project Management*, 21(1), 9–18.
- Anderson, E. J., & Nyirenda, J. C. (1990). Two new rules to minimize tardiness in a job shop. *International Journal of Production Research*, 28(12), 2277–2292.
- Ashtiani, B., Leus, R., & Aryanezhad, M.-B. (2011). New competitive results for the Stochastic resource-constrained project-scheduling problem: Exploring the benefits of pre-processing. *Journal of Scheduling*, 14(2), 157–171.
- Baccelli, F., Liu, Z., & Towsley, D. (1993). Extremal scheduling of parallel processing with and without real-time constraints. *Journal of the Association for Computing Machinery*, 40(5), 1209–1237.
- Baker, K. R., & Bertrand, J. W. M. (1982). A dynamic priority rule for scheduling against due-dates. *Journal of Operations Management*, 1(3), 37–42.
- Baker, K. R., & Kanet, J. J. (1983). Job shop scheduling with modified due dates. *Journal of Operations Management*, 4(1), 11–22.
- Ballestin, F. (2007). When it is worthwhile to work with the stochastic RCPSP? *Journal of Scheduling*, 10(3), 153–166.
- Ballestin, F., & Leus, R. (2009). Resource-constrained project scheduling for timely project completion with stochastic activity durations. *Production and Operations Management*, 18(4), 459–474.
- Bredael, D., & Vanhoucke, M. (2022). Multi-project scheduling: a benchmark analysis of metaheuristic algorithms on various optimisation criteria and due dates. *European Journal of Operational Research*, 308(1), 54–75.
- Browning, T. R., & Yassine, A. A. (2010). Resource-constrained multi-project scheduling: Priority rule performance revisited. *International Journal of Production Economics*, 226(2), 212–228.
- Browning, T. R., & Yassine, A. A. (2010). A random generator of resource constrained multi-project network problems. *Journal of Scheduling*, 13(2), 143–161.
- Browning, T. R., & Yassine, A. A. (2016). Managing a portfolio of product development projects under resource constraints. *Decision Sciences*, 47(2), 333–372.
- Carroll, D.C. (1965) *Heuristic sequencing of single and multiple component jobs*. PhD Thesis, Sloan School of Management, M.I.T., Cambridge, MA.
- Chen, H., Ding, G., Zhang, J., & Qin, S. (2019). Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival. *Computers & Industrial Engineering*, 137, 106060.
- Chen, F., Kolisch, R., Wang, L., & Mu, C. (2015). An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem. *Flexible Services and Manufacturing Journal*, 27(4), 585–605.
- Choi, J., Realff, M. J., & Lee, J. H. (2007). A Q-learning-based method applied to stochastic resource constrained project scheduling with new project arrivals. *International Journal of Robust and Nonlinear Control*, 17, 1214–1231.
- Dumond, J., & Mabert, V. A. (1988). Evaluating project scheduling and due date assignment procedures: An experimental analysis. *Management Science*, 34(1), 101–118.
- Fernandez, A. A., Armacost, R. L., & Pet-Edwards, J. J. A. (1996). The role of the nonanticipativity constraint in commercial software for stochastic project scheduling. *Computers & Industrial Engineering*, 31(1), 233–236.
- Fernandez, A. A., Armacost, R. L., & Pet-Edwards, J. K. (1998). Understanding simulation solutions to resource-constrained project scheduling problems with stochastic task durations. *Engineering Management Journal*, 10(4), 5–13.
- Fliedner, T. (2015). *Considering uncertainty in project management and scheduling*. PhD-thesis: Technical University of Munich.
- Gómez Sánchez, M., Lalla-Ruiz, E., Fernández Gil, A., Castro, C., & Voß, S. (2022). Resource-constrained multi-project scheduling problem: A survey. *European Journal of Operational Research*, 309(3), 958–976.

- Graham, R. L. (1966). Bounds for certain multiprocessing anomalies. *The Bell System Technical Journal*, 45(9), 1563–1581.
- Kanet, J. J., & Hayya, J. C. (1982). Priority dispatching with operations due date in a job shop. *Journal of Operations Management*, 2(3), 167–175.
- Kanet, J. J., & Li, X. (2004). A weighted modified due date rule for sequencing to minimize weighted tardiness. *Journal of Scheduling*, 7(4), 261–276.
- Kolisch, R. (1996). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, 14(3), 179–192.
- Kumar, P. R., & Seidman, T. I. (1990). Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Transactions on Automatic Control*, 35(3), 289–298.
- Kurtulus, I. S., & Davis, E. W. (1982). Multi-project scheduling: Categorization of heuristic rule performance. *Management Science*, 28(2), 161–172.
- Kurtulus, I. S., & Narula, S. C. (1985). Multi-project scheduling: Analysis of project performance. *IIE Transactions*, 17(1), 58–66.
- Kutanoglu, E., & Sabuncuoglu, I. (1999). An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. *International Journal of Production Research*, 37(1), 165–187.
- Law, A. (2007) *Simulation modeling and analysis* (4 edn). McGraw-Hill.
- Lawrence, S. R., & Morton, T. E. (1993). Resource-constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck, and resource pricing heuristics. *European Journal of Operational Research*, 64(2), 168–187.
- Levy, N., & Globerson, S. (1997). Improving multiproject management by using a queueing theory approach. *Project Management Journal*, 28(4), 40–46.
- Melchior, P. (2015). *Dynamic and stochastic multi-project planning, volume 672 of lecture notes in economics and mathematical systems*. Springer International Publishing.
- Melchior, P., & Kolisch, R. (2009). Scheduling of multiple R&D-projects in a dynamic and stochastic environment. In B. Fleischmann, K. H. Borgwardt, R. Klein, & A. Tuma (Eds.), *Operations Research Proceedings 2008* (pp. 135–140). Heidelberg: Springer.
- Meyn, S. (2008). *Control techniques for complex networks*. Cambridge: Cambridge University Press.
- Möhring, R. H., Radermacher, F. J., & Weiss, G. (1984). Stochastic scheduling problems I. *Zeitschrift für Operations Research*, 28(7), 193–196.
- Morton, T. E., Pentico, D. W. (1993). Heuristic scheduling systems. In *Wiley Series in Engineering & Technology Management*. Wiley, Hoboken.
- Ramasesh, R. (1990). Dynamic job shop scheduling: A survey of simulation research. *Omega*, 18(1), 43–57.
- Satic, U., Jacko, P., & Kirkbride, C. (2022). Performance evaluation of scheduling policies for the dynamic and stochastic resource-constrained multi-project scheduling problem. *International Journal of Production Research*, 60(4), 1411–1423.
- Schwindt, C. (1998). *Generation of resource-constrained project scheduling problems subject to temporal constraints*. Technical Report WIOR-543 University Karlsruhe.
- Stork, F. (2001). *Stochastic resource-constrained project scheduling*. PhD Thesis: Technical University Berlin.
- Tassioulas, L., & Georgiadis, L. (1996). Any work-conserving policy stabilizes the ring with spatial re-use. *IEEE/ACM Transactions on Networking*, 4(2), 205–208.
- Tsai, D. M., & Chiu, H. N. (1996). Two heuristics for scheduling multiple projects with resource constraints. *Construction Management and Economics*, 14(4), 325–240.
- Van Eynde, R., & Vanhoucke, M. (2020). Resource-constrained multi-project scheduling: Benchmark datasets and decoupled scheduling. *Journal of Scheduling*, 23(3), 301–325.
- Van Eynde, R., & Vanhoucke, M. (2022). New summary measures and datasets for the multi-project scheduling problem. *European Journal of Operational Research*, 299(3), 853–868.
- Vepsäläinen, A. P. J., & Morton, T. E. (1987). Priority rules for job shops with weighted tardiness costs. *Management Science*, 33(8), 1035–1047.
- Wang, Y., He, Z., Kerkhove, L.-P., & Vanhoucke, M. (2017). On the performance of priority rules for the stochastic resource constrained multi-project scheduling problem. *Computers & Industrial Engineering*, 114, 223–234.