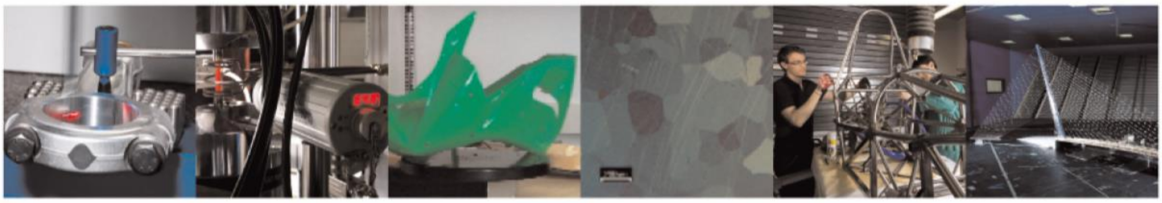


A predictive-reactive approach for the sequencing of assembly operations in an automated assembly line

Massimo Manzini, Erik Demeulemeester, Marcello Urgo

A Predictive-Reactive approach for the sequencing of assembly operations in an automated assembly line

Massimo Manzini[a], Erik Demeulemeester[b], and Marcello Urgo[a]

[a]Politecnico di Milano, Department of Mechanical Engineering, Manufacturing and Production Systems Division, Via La Masa, 1, Milan, Italy, massimo.manzini@polimi.it, marcello.urgo@polimi.it
[b]KU Leuven, Faculty of Business and Economics, Department of Decision Sciences and Information Management, Naamsestraat, 69, Leuven, Belgium, erik.demeulemeester@kuleuven.be

Abstract

The automotive industry is facing a rapid technological evolution and the request of a very high level of customization of the products. This requires production systems able to manage a high variety of products with low volumes. To this aim, this paper focuses on multi-product assembly lines consisting of a set of stations with a robot operating the transportation and handling of the parts. Due to the high variety of the parts to be processed, perfect balancing is not possible, hence, proper control policies are requested to operate the line. The paper proposes a *predictive-reactive* scheduling approach to minimize the batch completion time by sequencing the tasks operated by shared resources in a context with uncertain processing times. The viability of the approach is demonstrated through the application to an industrial problem in the automotive industry.

1 Introduction and motivation

More and more, the automotive industry has to cope with an increasing variety of models as well as multiple and heterogeneous materials and assembly technologies. Although the total number of parts constituting the car body has been significantly reduced, from about 500 in the late '90s to about 250/300 in more recent models [4], assembling the car body and its components (doors, fenders, etc.) still represents a critical phase in the production of a car. These processes are traditionally operated in assembly lines with a very high degree of automation, whose design has evolved in the past 30 years to match the evolution of the requirements for the automotive industry and taking advantage of industrial robots. Moreover, if we focus on the production of spare parts, the very low volumes for a single model requires processing multiple parts on the same assembly line to guarantee a reasonable utilization factor for the equipment. In addition, the rapid evolution of car models drives the need of frequently changing the mix of products to be processed and the associated volumes (at least on an annual basis).

Flexible and/or reconfigurable lines are the main design paradigms to cope with these requirements [**?**] according to the *co-evolution* principle [**?**], defined as the need of modifying the configuration of a production system together with the changes affecting the products or the processes. The technological advances of modern industrial equipment and the high degree of flexible automation are supporting the fulfilling of these needs but, from a design point of view, traditional line-balancing approaches are hardly effective. In fact, as the products (and the processing times) changes, a design that is balanced for a subset of products, is not likely to remain balanced for the whole product mix. The need to cope with unbalanced assembly lines increases the relevance of control policies able to properly schedule the operations to be executed according to the specific parts and processing times.

In this paper, we consider a class of assembly lines consisting of a set of assembly stations, an input and an output station and a handling robot moving on a track (see the example in Figure 1). The stations are positioned on both the sides of the track of the robot and implement technologies (e.g., clinching or hemming), while the handling robot moves the parts to be processed among the stations. Each station operates a specific manufacturing technology, in a way that the entire assembly process can be executed in the assembly line. In the input and output stations, the components to be assembled are loaded and the final part unloaded. Due to the high variety of products, loading and unloading operations are executed by human workers. This class of assembly line usually operates as a multi-product line, thus, the production is operated in batches of the same product type, with set-up phases to move from the production of one part to another.

Grounding on this, considering a single batch, the parts to be produced have the same process and routing, as in a *flow-shop* system, but some operations (e.g., transportation) need a resource (the handling robot) that is shared among different operations. The consequence is the possible presence of simultaneous

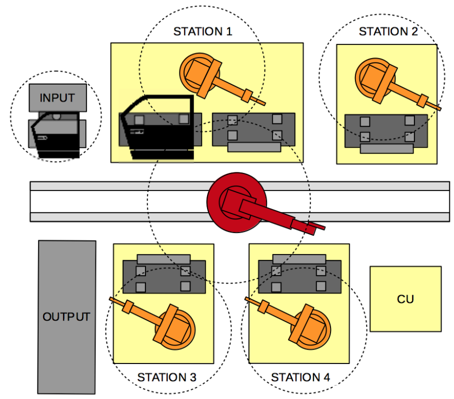


Figure 1: Exemplar assembly line with 4 technological stations.

requests for the same resource that require the sequencing of the operations that share those resources. In addition to this, the duration of every operation executed in the line is subject to uncertainty, e.g., due to the human execution of input and output operations and possible breakdown or micro stops for the automated stations.

The aim of this work is to investigate control policies for the described class of assembly lines grounding on a *predictive-reactive* scheduling approach aiming at minimizing the completion time to produce a batch of identical parts, and able to face the process uncertainty. In particular, the proposed *predictive-reactive* (PR) scheme firstly provides a *baseline* schedule taking into consideration the uncertainty affecting the processing times to a certain extent, then, as soon as the uncertainty discloses, a *reactive* scheduling step is operated to adapt the *baseline* schedule to the actual duration of the operations.

Outline The paper is organized as follows: Section 2 provides an analysis of the literature, while the complete problem statement is presented in Section 3 where the scheduling problem that is dealt with is formalized. In Section 4, the solution approach is described in terms of the *predictive* step in Section 4.1 and the *reactive* step in Section 4.2. The viability of the approach is demonstrated through the application to an industrial problem in Section 5. Conclusions and future development directions are provided in Section 6. Additional data and tables are included in Appendix.

2 State of art

Flow-shop scheduling problems have been extensively investigated in the last decades addressing many variants of the base shop problem.

The most important contributions in the area of control policies for flow shops considering uncertainty are presented in [16, 17], addressing the minimization of the expected value of the completion time by identifying an optimal sequence of jobs to be processed. In this paper, we consider the production of a batch of identical jobs, thus, sequencing is not a decision to be taken. The focus of this work is on sequencing operations involving a shared resource, i.e., a transported moving the parts from a station to another.

Under this perspective, the *flexible assembly system scheduling problem* have been considered. An example is the approach presented in [**?**] where authors consider the scheduling of flexible manufacturing cell to determine the transportation order of the jobs , together to the assignment of jobs to processing resources.

Other relevant contributions have been presented in [**? ? ? ? ?**] considering the material handling cost as one of the main driver for the optimization of a flexible system. Both these approaches can be adapted to match the assembly line under study in this article, but they are limited to deterministic parameters and are not designed to be used as control mechanisms during the execution of the schedules.

Other relevant contributions are those focusing on the *scheduling of robotic cells* and the *cyclic robot scheduling problem* [**?**]. The first group of works [**? ? ?**] study the robotic flow shop systems in which one or more stages are served by one or more robots, very similar to the one under analysis in this paper. These studies focus on the identification of the bottleneck of the process and use it as the decision point for scheduling the operations. These approaches are not suitable with the problem under study because they consider the physical modification of the system. While the second one study the systems in which a manufacturing system is served by a handling robot working as a transportation device [**? ? ?**]. In these works, the main aim is to regulate the handling robot missions in order to minimize the completion time of a single job. Also in these cases, the duration of every operation is considered deterministic without the possibility to regulate the on-line management of the system.

Grounding on this analysis, more general scheduling approaches related to the *Resource-Constrained Project Scheduling Problem* (*RCPSP*) have been taken into consideration [19]. Due to the need to cope with uncertainty, two sub-problems have been addressed: the stochastic *RCPSP* and the *rescheduling* of manufacturing systems.

The research related to the stochastic *RCPSP* aims at minimizing a scheduling objective function, e.g., the completion time, by developing policies rather than schedules. A policy is a set of rules that support scheduling decisions, i.e., if a certain event occurs, then a specific action has to be taken. A first class of approaches formalizes the scheduling decisions as a *multi-stage* decision problem [5, 7, 6]. Specifically, the scheduling problem is decomposed in multiple decision stages and, for each of them, a schedule of the activities is provided, taking into consideration the availability of resources, precedence constraints as well as the available information related to uncertain variables at that stage.

A second class of approaches includes *preselective* and *early-start* policies. *Early-start* policies (*ES*) are first introduced in [10, 11] and further investigated in [18]. These policies are based on the definition of *minimal forbidden sets*, i.e., sets of activities with minimal cardinality whose concurrent execution surely violates the resource constraints. In an *ES* policy, for each *minimal forbidden set* F there exists a pair (i, j), $i, j \in F$, $i \neq j$ that for each sample of activity durations, j cannot start before i has finished. This kind of policies can be implemented by adding a precedence relation (i, j) to the original scheduling problem. On the other hand, *preselective* policies are introduced in [11], also exploiting the notion of *minimal forbidden sets*, and also in [14] where a variation of Dijkstra's shortest path algorithm is used. A policy is defined *preselective* if for each *minimal forbidden set* F there exist an activity $j \in F$ (the preselected one) that, for each sample d of activity durations, j cannot start before the end of one of the other activities.

Due to the difficulty to identify optimal policies in the stochastic version of the problem, dominance rules have been proposed in [19] and [18]. In [19], *branch-and-bound* algorithms are developed to provide upper and lower bounds for a given policy. The author also addresses bounds and dominance rules between *ES*, *preselective* and *job-priority* policies. Heuristic approaches for the stochastic *RCPSP* have also been proposed in [15], combining genetic algorithms and simulation. In [8], an alternative stochastic formulation of the problem is described and solved through a heuristic approach as well. In addition, in [20] and [21] a tabu search algorithm is presented.

More recently, [1] proposed an optimal approach for this class of problems limited to the case of exponential and phase type distributed processing times. All the cited works address the optimization of the expected value of an objective function, e.g., the minimization of the expected completion time. Nevertheless, this does not protect against rare but very extreme scenarios, as discussed in [**?**] and in [**?**] for a generic production plan, in [**?**] for the single machine case, and in [**?**] and [**?**] with regards to *Make-to-Order* processes.

To overcome these limitations, a second class of approaches has been addressed, considering *rescheduling* actions as the process of updating an existing production schedule in response to disruptions or other changes. Grounding on the framework presented in [**?**], we focused our analysis on *stochastic and static rescheduling environments*. This class of problems considers to have a finite set of jobs or operations to be scheduled, whose durations are uncertain [17**? ?**].

We investigated two different classes of methods that are able to solve this problem, *dynamic scheduling* and *predictive-reactive* (or *proactive-reactive*, alternatively). Methods belonging to the first class do not define a baseline schedule, but dispatch jobs and operations as they are ready to start, using only available information [**? ?**]. These methods are closely related to real-time control approaches, not considering the uncertainty in advance before the execution of the process. In the problem under study, we assume that a model of the uncertainty associated to process times is available, and thus, the objective is being able to exploit this information. Methods belonging to the second class start with the definition of a *baseline* schedule, i.e., an initial schedule that takes uncertainty into consideration to a certain extent. Afterwards, triggered by possible deviations with respect to the baseline schedule, a rescheduling step is operated, with the aim of revising the *baseline* one, thus *reacting* to what occurred.

Within this class of approaches, it is relevant to mention [13] who demonstrates that the scheduling problem with a single conflict and precedence constraints is already strongly *NP-hard* even for a single machine. Moreover, [9] propose exact methods to build robust *baseline* schedules. Further relevant approaches from this class are proposed in [12] and [2]. The first one presents a *chance-constrained* approach, while the second one first develops a set of possible scheduling solutions and later on decides how and when to switch among them during the execution of the process.

Differently from these approaches, the one proposed in this paper grounds on a two-stage scheme i) able to identify a set of constraints to be enforced among the operations and addressing rare and extreme scenarios that can affect the completion time, ii) by taking advantage of the specific characteristics of the scheduling problem, e.g., the repetition of the jobs. Moreover, the *reactive* step is intended to be operated on-line during the processing of the assembly process, triggered by possible deviations between the actual and estimated durations of the operations.

3 Problem statement

We consider the process of assembling a batch of identical parts in a manufacturing system organized as a *flow shop*, with no buffer between the stations. The operations to be executed are represented through an *Activity-on-Node* (*AoN*) network of activities where $V = \{0, 1, ..., m\}$ is the set of nodes representing operations and $E = (x, j),\ x, j \in V$ the set of arcs modelling precedence constraints. An example is shown in Figure 2 with an input (I_i) and output (O_i) operations at the beginning and at the end of the process, two assembly operations (A_{1i} and A_{2i}) and three transport operations, one between the input and the assembly station (T_{1i}), a second one between the two assembly stations (T_{2i}) and a third one between the second assembly station and the output (T_{3i}). This process is repeated for each part i in a batch of n identical parts, with $i \in [1, n]$. Being a permutation flow shop, all the parts are processed according to the same sequence in all the stations. Hence, the first assembly operation on the job 1 will be executed before the same operation on job 2, thus, $A_{11} \prec A_{12}$ (where $\prec$ represents a precedence constraint).

Transport operations T_{1i}, T_{2i} and T_{3i} require the handling robot. Due to the absence of buffers between subsequent stations, while a part is waiting for the robot, it blocks the station where it has been processed. For this reason, to define the sequencing of all the operations in the system, additional constraints must be added between transport operations, e.g. $T_{21} \prec T_{12}$ [**?**]. Hence, scheduling the missions of the robot is the main decision impacting the performance of the system. As an example, let us consider operations T_{11}, T_{12}, T_{31} and T_{32}, where T_{ij} is the transport operation i for job j. While the precedence relation $T_{11} \prec T_{32}$ is a consequence of the structure of the process ($T_{11} \prec A_{11} \prec T_{21} \prec A_{21} \prec T_{31} \prec T_{32}$), the sequencing of T_{31} and T_{12} is not *a-priori* defined. We model the described scheduling problem through the introduction of *disjunctive*

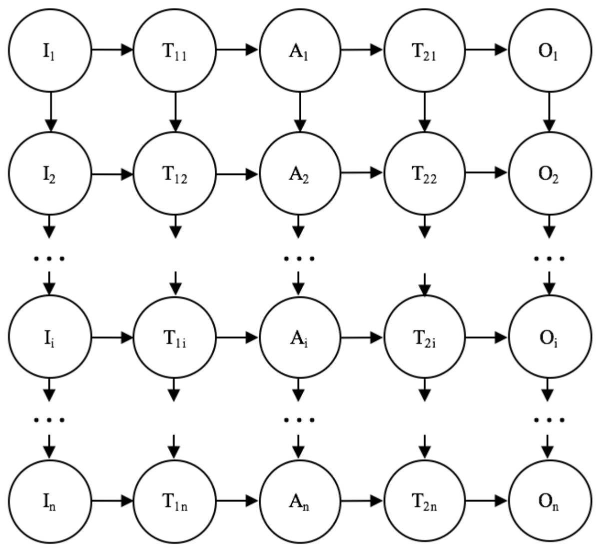


Figure 2: *AoN* network of activities for a n-product batch.

constraints defined as a set of two alternative precedence constraints whereof only one has to be added to E. The selection of these constraints is operated with the aim at minimizing the completion time of a batch of jobs. The constraints of this type are represented with the set E_{DC}, additional to E.

Random variables are used to model processing times to take into consideration manual activities or the occurrence of micro-failures (e.g., tool changes). Processing times are represented through a vector $\tilde{\mathbf{p}} = \tilde{p}_0, \ldots, \tilde{p}_m$, where p_i is a sample from the distribution associated to $\tilde{p}_i$ and $\mathbf{p} = p_0, \ldots, p_m$ a sample of the entire set of random processing times $\tilde{\mathbf{p}}$, $\forall i \in [1, m]$.

The approach grounds on a formal description of the problem reported in Table 1.

4 Solution approach

The *predictive-reactive* scheduling approach consists of two steps. The first one provides a *baseline* schedule grounding on a given duration of the operations affected by uncertainty (e.g., a quantile of the associated distribution), thus, addressing a deterministic problem. The second step is applied during the execution of the process, considering the actual duration of the operations. Every time a delay from the *baseline* schedule is identified, a *reaction* is evaluated to check whether the constraints selected in the previous step are still optimal. If not, a new set of constraints is selected.

	Sets
V	set of nodes representing operations
E	set of arcs representing precedence constraints
$\tilde{\mathbf{p}}$	vector of random processing times
$\mathbf{p^q}$	vector of processing times, using a quantile q
$A(t)$	set of on-going operations at time t
Ω	state space
O	set of operations in execution
F	set of completed operations
S	set of starting times
$d_O(o, t)$	set of durations of operations in execution, $\forall o \in O$, at time t
$prec(k)$	set of operations preceding operation k
	Variables
E_{DC}	set of arcs added with the *predictive* step
S_j	starting time of operation $j, \forall j \in V$
	Parameters
q	quantile $\in (0, 1)$
p_j^q	processing time of operation j using quantile q
r_j	equal to 1 if operation j needs handling robot, 0 otherwise
$Q_k^{p^q}$	eligible time of operation k
t	time index
n	number of parts in the batch
m	number of operations considered
τ	schedule time horizon
$\Delta_{j,x}^{p^q}$	difference between eligible times of operations j and x
$\Delta_{j,x}^{T}$	threshold of the difference between eligible times of operations j and x
$\omega(\mathbf{p}, t)$	state of the system at time t and processing times $\mathbf{p}$
CT	probability threshold
$\Delta_{k,x}^{\mathbf{p}}(t)$	actual difference between the eligible times of operations x and k at time t

Table 1: Notation for set, parameters and variables.

4.1 Predictive step

The *predictive* step assigns all the operations a duration derived from a quantile $q \in (0, 1)$ of the stochastic distributions modelling the processing times, thus, obtaining a vector $\mathbf{p^q} = p_0^q, \ldots, p_m^q$. The selection of the quantile depends on the risk aversion to be adopted in this step. The higher the quantile, the smaller the probability to experience a delay with respect to the *baseline* schedule during the *reactive* step and, consequently, the more cautious the *baseline* schedule. Although, it is possible to use different quantiles for each operation, but in the proposed approach, a single value is used.

Under these hypotheses, the *predictive* step is operated through a deterministic scheduling approach with the aim at minimizing the batch completion time. In doing this, we adopt a classical formulation of a *RCPSP* [**?**] defined by Equations (1)-(4).

$$
\begin{aligned}
& minimize \qquad S_m + p_m^q && (1)\\
& subject \quad to \\
& S_x + p_x^q \leq S_j \quad \forall (x,j) \in E && (2)\\
& \sum_{j \in A(t)} r_j \leq 1 \quad \forall t \in [0, \tau] && (3)\\
& S_j \geq 0 \quad \forall j \in V && (4)
\end{aligned}
$$

The objective function minimizes the batch completion time (Equation (1)), in terms of the completion time $S_m + p_m^q$ of operation m (the last one in the batch), where S_j represents the starting time of operation $j, \forall j \in V$. Precedence and resource constraints, defined by Equations (2) and (3) have to be respected.

Resource requirements are

are modelled through a parameter r_j equal to 1 if operation j needs the handling robot and 0 otherwise. For every set of on-going operations $A(t) = \{j \in V \mid t \in [S_j, S_j + p_j^q]\}$, defined at time $t \in \tau$, where τ is the schedule time horizon, at most one operation is allowed to be executed by the handling robot. This scheduling problem can be solved using the approach described in [3], able to select the constraints to be activated among the *disjunctive sets* and include them in the set E_{DC} additional to E.

Starting from the *baseline* schedule, a sensitivity analysis is performed on the duration of the operations. For each constraint $(z, x) \in E_{DC}$, the sensitive analysis is used to identify a threshold value for the duration of the operations such that, if exceeded, constraint (z, x) is no longer optimal and the alternative constraint (x, z) should be considered.

To provide an example, let us consider the schedule depicted in Figures 3a - 3d. We consider three jobs 1, 2 and 3 to be processed in an assembly line consisting of three stations and two transport operations among the stations, similar to the example in Figure 2. Hence, job 1 has to undergo five operations: the input I_1, the first transportation T_{11}, the assembly operation A_1, the second transportation T_{21} and the output O_1.

Jobs 2 and 3 follow the same process in terms of the set of operations I_2, T_{12}, A_2, T_{22}, O_2, and I_3, T_{13}, A_3, T_{23}, O_3, respectively. Operations T_{1i} and T_{2i}, with $i \in \{1, 2, 3\}$ are executed by the handling robot and represented in orange in Figures 3a - 3d.

Focusing on transport operations T_{12} and T_{21}, competing for the use of the handling robot, two alternative precedence constraints exist, i.e., (T_{12}, T_{21}) and (T_{21}, T_{12}) in Figure 3a. (T_{21}, T_{12}) results optimal due to the shorter completion time compared to the alternative constraint (25 and 26 time units, respectively).

We define $Q_k^{\mathbf{p^q}}$ as the eligible staring time of an operation k, given the processing times $\mathbf{p^q}$ and considering the precedence constraints in E, hence, without taking into considerations those in the set E_{DC}. The eligible starting time for operation T_{21} is 7 (just after the completion of A_1), while for T_{12} is

6 (after the completion of T_{11}). The difference between these eligible times is $Q^{\mathbf{p^q}}_{T_{21}} - Q^{\mathbf{p^q}}_{T_{12}} = 7 - 6 = 1$.

Hence, we take into consideration the variability of the process times in order to evaluate the viability of the precedence constraint (T_{21}, T_{12}). If the duration of operation A_1 is longer than 1 time unit, value considered for the identification of the optimal schedule in Figure 3a, a delay of operation T_{21} occurs together with an increased completion time of the whole schedule.

We consider the cases where the duration of operation A_1 is 1, 2 or 3 time units longer than the one considered before (Figures 3b - 3d) and suppose to invert the constraint (T_{21}, T_{12}) to (T_{12}, T_{21}) as soon as we realize the delay respect to the *baseline* schedule, e.g., when, after 1 time unit, operation A_1 is not completed yet. In particular, with a delay of 1 time unit (Figure 3b), the completion time enforced by constraint (T_{21}, T_{12}) remains optimal but, with a delay of 2 time units (Figure 3c) the two constraints provide the same completion time. On the contrary, with a delay of 3 time units, the completion time enforced by (T_{12}, T_{21}) is shorter than the one with (T_{21}, T_{12}). Hence, the reversed constraint (T_{12}, T_{21}) is beneficial for the minimization of the completion time if and only if the delay of operation T_{21} is larger than 2 time units.

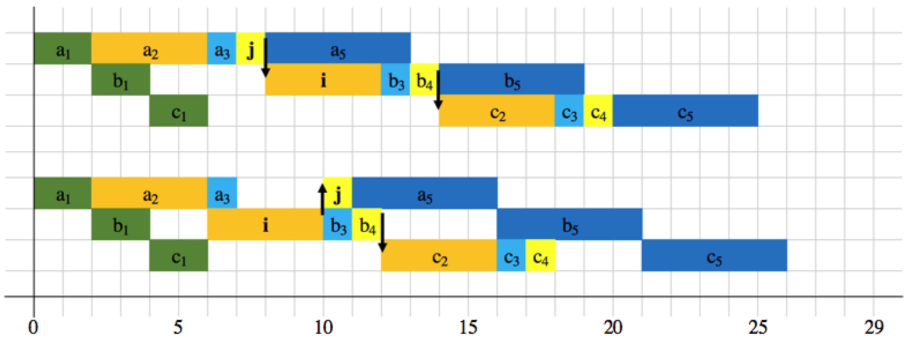
To formalize this reasoning, we define $\Delta^{\mathbf{p^q}}_{T_{21},T_{12}} = Q^{\mathbf{p^q}}_{T_{21}} - Q^{\mathbf{p^q}}_{T_{12}}$ as the difference between the eligible times of the two operations linked by the constraint (T_{21}, T_{12}). Given $S_m^{(T_{21},T_{12})} + p^q_m$ and $S_m^{(T_{12},T_{21})} + p^q_m$ as the completion times of the schedules obtained with precedence constraints (T_{21}, T_{12}) and (T_{12}, T_{21}), respectively. An inversion is effective only if the difference between the eligible times of operations T_{21} and T_{12} is greater than a threshold whose value is the difference between i) the difference of the eligible times and ii) the difference between the completion times, both calculated in the *baseline* situation. More formally, the threshold is defined as $\Delta^{T}_{T_{21},T_{12}} = \Delta^{\mathbf{p^q}}_{T_{21},T_{12}} - \left(S_m^{(T_{21},T_{12})} - S_m^{(T_{12},T_{21})}\right)$.

Following the example in Figures 3a - 3d, $\Delta^{\mathbf{p^q}}_{T_{21},T_{12}} = Q^{\mathbf{p^q}}_{T_{21}} - Q^{\mathbf{p^q}}_{T_{12}} = 1$ and, thus, the value of the threshold is $\Delta^{T}_{T_{21},T_{12}} = \Delta^{\mathbf{p^q}}_{T_{21},T_{12}} - \left(S_m^{(T_{21},T_{12})} - S_m^{(T_{12},T_{21})}\right) = 1 - (25 - 26) = 2$. Indeed, during the execution of the assembly process, if the starting time of operation T_{21} experiences a delay bigger than the threshold, then the opposite constraint (T_{12}, T_{21}) guarantees a shorter completion time than the opposite one.

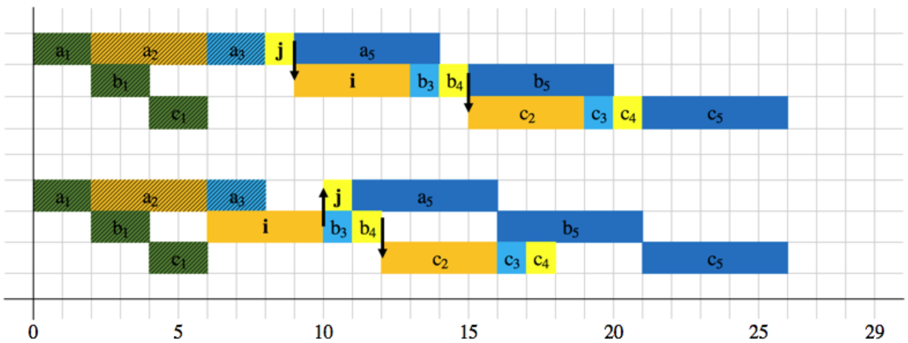
The $\Delta^{T}_{T_{21},T_{12}}$ provides a threshold value to identify if a delay of the start time of an operation causes the *baseline* schedule to be no longer optimal and, hence, it should be modified. This consideration will be used in the *reactive* step to provide an alarm and trigger possible modifications of the schedule during the execution of the process.

4.2 Reactive step

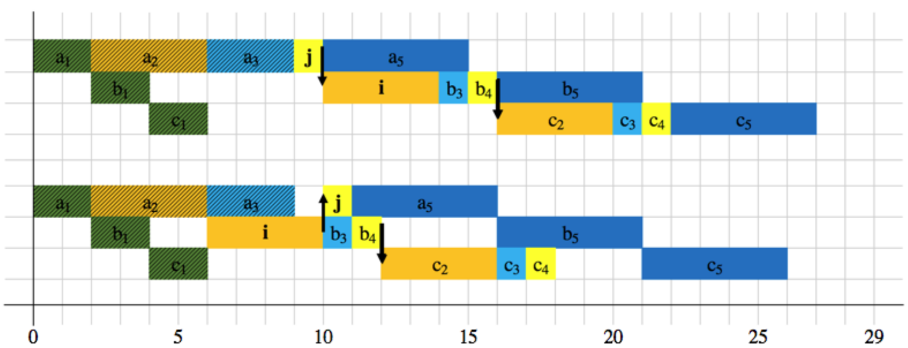
The *reactive* step is applied at the execution phase, taking into consideration the actual duration of the operations under the hypothesis that this information becomes known (available) only when an operation is completed. Before this



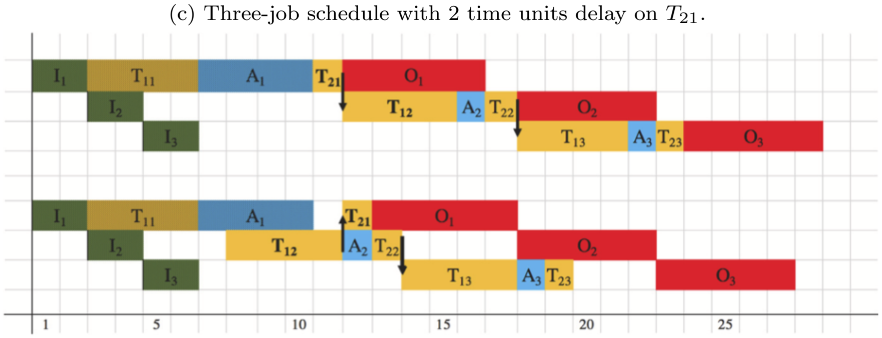
(a) Three-job schedule in the *baseline* situation.



(b) Three-job schedule with 1 time unit delay on T_{21}.



(c) Three-job schedule with 2 time units delay on T_{21}.



(d) Three-job schedule with 3 time units delay on T_{21}.

Figure 3: The delay effect on a three-job *flow shop*. The operations already finished and for which the actual duration is undisclosed are shaded.

event, it is assumed that durations will be the one hypothesized in the *predictive* step. We also assume to be able to identify a delay of an operation as soon as its completion time become larger than the one in the baseline schedule.

Every time a delay of an operation is identified, the *reactive* policy evaluates if the constraints associated to that operation included in the set E_{DC} are still optimal. This step grounds on the definition of the state space Ω, i.e., a sequence of states $\omega(\mathbf{p}, t)$ varying over time t, defined as $\omega(\mathbf{p}, t) = (O, F, S, d_O) \in \Omega$. Each state is fully described by:

- O, set of operations in execution at time t;
- F, set of completed operations at time t;
- S, set of starting times of the operations. For the operations in O or F, the starting times are already defined, while for operations neither in O nor in F, the starting times are not yet decided;
- $d_O(o, t)$, set of durations of operations in execution ($o \in O$) at time t.

The *reactive* step requires the definition of a probability threshold CT representing the limit probability to trigger a modification of the *baseline* schedule. It could be defined as the inclination to modify the *baseline* schedule during the *reaction* step. The smaller the CT, the higher the probability to change. The *reactive* procedure is described in Algorithm 1.

Reactive-Procedure

$\omega(\mathbf{p}, 0) == (0, \emptyset, 0, 0)$
While $F <> V$
    $t = t + 1$
    If $d_O(x, t) - S(x) = p_x, \ \forall x \in O$
        $F = F + x$
    Else
        $d_O(x, t) = d_O(x, t) + 1$
    If $x \notin O \wedge x \notin F \wedge z \in F, \ \forall z \in (z, x) \in E$
        If $(k, x) \in E_{DC} \wedge \mathbb{P}\left[\Delta^{\mathbf{p}}_{k,x}(t) > \Delta^{T}_{k,x}\right] > CT$
            $E_{DC} = E_{DC} - (k, x) + (x, k)$
            $O = O + x$
            $S(x) = t$
            Update Δ^T
        Else
            $O = O + x$
            $S(x) = t$
End

Algorithm 1: *Reactive* step procedure.

The algorithm models the execution of the operations starting from $t = 0$ with initial state $\omega(\mathbf{p}, 0) = (0, \emptyset, 0, 0)$ and finishes when all the operations are completed, i.e., $F = V$ (steps 1-2). It increases the time t together with the durations of operations in execution; every time an operation is completed, the set F is updated (steps 3-7). If there is an operation x that can start because all its predecessors are completed (step 8), it is put into execution and added to the set of ongoing operations O (steps 15-16). On the contrary, if its execution is constrained by the completion of another operation k due to a precedence included in E_{DC} (step 9), then the algorithm checks whether the constraint (k, x) remains optimal for the values in $\mathbf{p}$. In other words, the algorithm checks whether operation x has to wait the completion of operation k respecting the constraint $(k, x) \in E_{DC}$, or not.

This evaluation is done through the estimation of the probability that the actual difference between the eligible times $\Delta^{\mathbf{p}}_{k,x}(t)$, estimated at time t and considering values in $\mathbf{p}$, exceeds the threshold identified in the *predictive* step: $\mathbb{P}\left[\Delta^{\mathbf{p}}_{k,x}(t) > \Delta^{T}_{k,x}\right]$. If this probability exceeds the threshold CT, the reaction is applied by inverting the constraint (k, x) and operation x is put in execution (steps 10-12). If the reaction is applied, the set containing all the thresholds Δ^T is updated because the sensitivity analysis done during the *predictive* step could be not valid anymore. The new threshold is estimated as depicted in Figures 3a - 3d in Section 4.1, with the difference that the precedence constraint between job 1 and the previous one has been reversed.

The $\mathbb{P}\left[\Delta^{\mathbf{p}}_{k,x}(t) > \Delta^{T}_{k,x}\right]$ is estimated considering the duration of the each operation in O preceding k and their distributions $\tilde{p}$, $\forall u \in prec(k)$, where $prec(k)$ is the set of operations preceding k (Equation (5)).

$$\mathbb{P}\left[\Delta^{\mathbf{p}}_{k,x}(t) > \Delta^{T}_{k,x}\right] =$$

$$= \mathbb{P}\left[Q^{\mathbf{p}}_{k} - Q^{\mathbf{p}}_{x} > \Delta^{T}_{k,x} \mid t, d_O(u,t), \tilde{p}_u\ \right] \tag{5}$$

$$= \mathbb{P}\left[max_{u \in prec(k)}(S^{\mathbf{p}}_{u} + p_u) - Q^{\mathbf{p}}_{x} > \Delta^{T}_{k,x} \mid t, d_O(u,t), \tilde{p}_u\right] \tag{6}$$

$$= \mathbb{P}\left[max_{u \in prec(k)}(S^{\mathbf{p}}_{u} + p_u - d_O(u,t)) > \Delta^{T}_{k,x} + Q^{\mathbf{p}}_{x} - t \mid t, \tilde{p}_u\right] \tag{7}$$

The probability that $\Delta^{\mathbf{p}}_{k,x}(t)$ is bigger than $\Delta^{T}_{k,x}$ is equal to the probability that the difference between the completion time of the last preceding operation of k and the eligible time of x is bigger than $\Delta^{T}_{k,x}$ (Equation (6)). In this case, $max_{u \in prec(k)}(S^{\mathbf{p}}_{u} + p_u)$ represents the completion time of the last preceding operation of k, with p_u as the actual duration of operation u, and $Q^{\mathbf{p}}_{x}$ as the eligible time of x. Obviously, since operation k is not yet eligible at time instant t, $S^{\mathbf{p}}_{u} + p_u$ is unknown for at least one operation u preceding k, thus, its distribution $\tilde{p}_u$ and its on-going duration $d_O(u,t)$ affect the estimation.

In other words, we estimate the probability (Equation (7)) that the residual duration of the operations preceding k at time t ($S^{\mathbf{p}}_{u} + p_u - d_O(u,t)$), representing the time units to be waited until operation k becomes eligible, is bigger than the time units until the threshold is reached ($\Delta^{T}_{k,x} + Q^{\mathbf{p}}_{x} - t$). This estimation is executed at time t, where $d_O(u,t)$ and $Q^{\mathbf{p}}_{x}$ are deterministic values, since the first one is the on-going duration of operation u and the second one is the eligible time of operation x. In the cases where not all the predecessors of k are on-going, thus, $d_O(u,t)$ is unknown for at least one $u^* \in prec(k)$, its starting time $S^{\mathbf{P}}_{u^*}$ has to be estimated considering its set of predecessors (e.g., the set $v \in prec(u^*)$) following the same logic as described for the set $prec(k)$.

5 Application

5.1 Use-case presentation

The proposed approach has been validated on an industrial case considering the assembly process of the door of a car. The process takes as input the structure of the door and applies additional components using different assembly technologies.

This process has been implemented by the *OEM* company providing the industrial case adopting an assembly line following the layout in Figure 1, composed of a set of stations (4 in the example) that operate a specific assembly technology and a handling robot to transport the door and its component through the line, moving on its track. A control unit (CU) as well as input and output stations are also present.

This system operates an assembly process as described in Figure 2. The assembly operations to be executed are reported in Figures 4a - 4d and entails one or more of the following operations:

1. the assembly of two hinges for the opening mechanism through a nut pressing operation (Figure 4a);

2. the assembly of a reinforcement bar through a spot welding operation (Figure 4b);

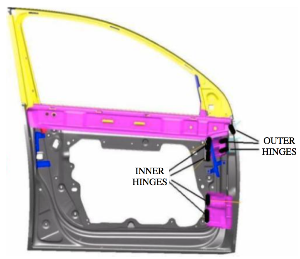
3. the joining of the resulting inner part (in Figure 4c) and outer part of the door (Figure 4d) through a roll hemming operation.

Between any pair of operations, the handling robot transports the inner part of the door from a station to the following one.

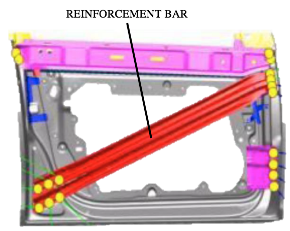
5.2 Testing phase

In order to test the presented approach we consider three different assembly processes, with 1, 2 or 3 of assembly operations and, thus, 5, 7 or 9 operations respectively, including the input and output operations, and the transport ones between assembly operations.

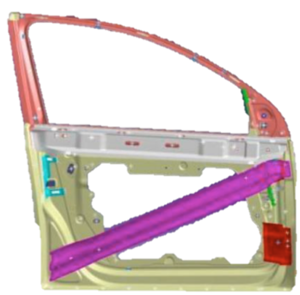
For the cases with 7 and 9 operations, we assume the absence of buffers, while for the case with 5 operations we assume the presence of a one-position buffer after the assembly station. In addition, the duration of every operation is subject to uncertainty (human execution for the input and output operations



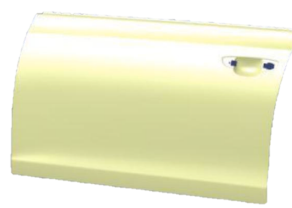
(a) The assembly of hinges through a nut pressing operation.



(b) The assembly of the reinforcement bar through spot welding operations.



(c) The assembled inner part of the door.



(d) The outer and inner parts of the door are joined through a roll hemming operation.

Figure 4: Assembly steps for assembly a door of a car.

and micro stops for the others). Thus, we assume stochastic processing times modelled through uniform or triangular distributions.

In the following testing phase, the parameters of the distributions are randomly generated by choosing the average value (μ for the uniform distributions and the mode value pv for the triangular ones) in the range $[2, 50]$, and the lower and upper limits (ll and ul) using a parameter $\lambda \in (0, 1)$. For the uniform distribution, the μ value is exactly the average value between the lower and the upper limit, thus $ll = \mu(1-\lambda)$ and $ul = \mu(1+\lambda)$. For the triangular distributed variables, the mode value pv is closer to the lower limit than to the upper one in order to provide a reasonable model, thus, $ll = pv(1-\lambda/2)$ and $ul = pv(1+\lambda)$.

In a first experimental phase, we use this model for the investigation of the average behaviour of the approach by generating a series of instances using different values of λ, and a number of jobs equal to 5 or 10. For each combination of the values of λ and number of jobs, 5 different instances have been generated and analyzed by applying Algorithm 1 considering $10,000$ samples from the set p and different quantiles q as well as *Change Threshold* CT. The parameters used are summarized in Table 2. This test is used to evaluate the impact of the

Parameter	Range
λ	0.3 0.5 0.9 0.95
q	0.1 0.3 0.5 0.7 0.9
CT	0.1 0.3 0.5 0.7 0.9

Table 2: Set of parameters used in the experimental phase.

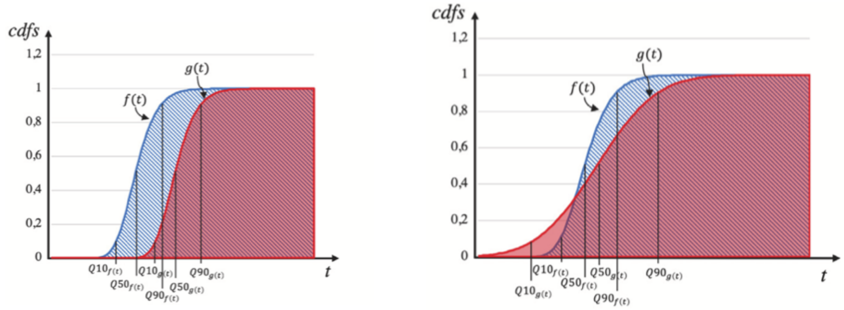
risk aversion used in the *predictive* step (q) and the tendency to change used during the *reaction* step (CT).

For each instance, the proposed approach has been applied and compared with i) the completion time obtained through the application of the *predictive* step only (*P-only*), ii) a conventional dispatching algorithm and iii) the best possible scheduling solution identified by the algorithm in [3], under the hypothesis that processing times are given. In particular, the conventional dispatching algorithm applies the *First Come First Served* rule combined with the *Fewest Remaining Operations* rule [**?**] (*FCFS/FRO*). In this way, the *FCFS* is applied but, if two or more transport activities ask for the handling robot at the same time instant, the one with fewest reaming operations is one served first, thus applying the *FRO* rule.

On the other hand, the algorithm used for the third comparison is able to identify the best solution of the deterministic problem in 100% of the cases (no limit for the computational time is enforced). The performance of the presented approach has been measured in terms of the *Average Quadratic Distance* (AQD) of the estimated *cumulative density function* (*cdf*) of the completion time from the best solution completion time *cdf* and then compared to the same measure for the *P-only* and the*FCFS/FRO* cases.

The AQD between two *cdf*s represents the measure of the area between them, that is the difference between their integrals. If the distribution functions have different shapes, it could happen that the two *cdf*s have an intersection, thus there is no dominance in terms of quantiles (Figure 5b). In this case, the AQD only gives a partial understanding. When one of the two *cdf*s dominates the other one, as depicted in Figure 5a, the AQD represents the absolute difference between the two distributions and, thus, it is a valuable measure of the performance.

Moreover, to provide a more detailed comparison of the *cdf*s associated to the application of the *predictive-reactive*, the *P-only* and the *FCFS/FRO*, we compare their quantiles (see Figures 5a-5b). The results are reported in Tables 7-12 (in Appendix) where for each combination of the values of CT and q, and for different quantiles, we show the difference between the value of the *P-only*'s *cdf* and the *predictive-reactive* one. Aggregated results are reported in Table 3. It is possible to see that, for the three quantiles analyzed (10th, 50th and 90th) this difference is always positive or equal to 0, showing that



(a) Comparison between distribution functions with the same shape.

(b) Comparison between distribution functions with different shapes.

Figure 5: *AQD* as the measure of the area between distribution functions and the representation of different quantiles.

the *predictive-reactive's cdf* lies below the *P-only cdf*. Similar results have been obtained for the *FCFS/FRO cdf*. As a consequence, the *AQD* represents a valuable performance measure in comparison with the *P-only* approach.

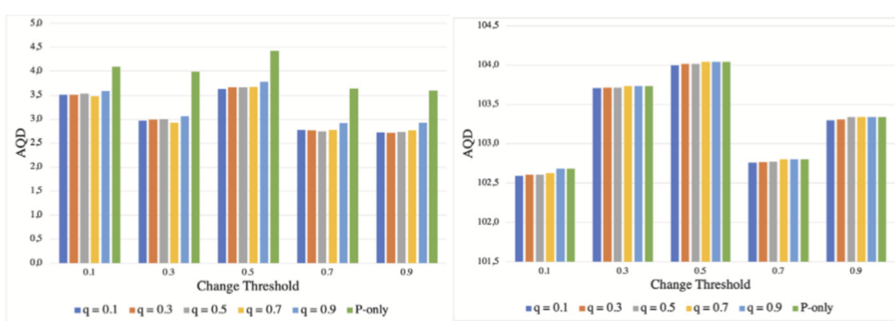
5 **Jobs**			
	Q10	**Q50**	**Q90**
5-uniform	1.6	1.7	0.1
5-triangular	1.4	1.5	0.1
7-uniform	1.5	1.6	0.1
7-triangular	1.3	1.4	0.1
9-uniform	1.6	1.7	0.1
9-triangular	1.3	1.5	0.2
10 **Jobs**			
	Q10	**Q50**	**Q90**
5-uniform	1.2	1.0	0.1
5-triangular	1.0	0.9	0.1
7-uniform	1.4	1.3	0.1
7-triangular	1.0	0.9	0.2
9-uniform	1.8	1.6	0.1
9-triangular	1.1	1.0	0.3

Table 3: Average difference between the 10th, 50th and 90th quantiles*P-only*'s *cdf* and the *predictive-reactive cdf*.

The results of the tests in terms of the *AQD* are reported in Tables 13-18 (in Appendix), each one referring to a different set of instances where both the number of operations to be executed (5, 7 or 9) and the distributions associated to the processing times (uniform or triangular) vary. Each table reports the value of the AQD between the solution obtained with the *predictive-reactive* approach (PR), the *P-only* one, or the *FCFS/FRO* approach against the optimal solution obtained under complete information.

The results show that the PR always performs better than the *P-only* and the *FCFS/FRO*, i.e., the value of the distance between the *cdf* obtained with the PR approach and the one associated to the optimal solution is always smaller or equal to the one obtained with alternative approaches. In the cases where the performance of PR and *P-only* are equal, it always means that no reaction has been operated in the *reactive* step and, thus, the schedule obtained through the *predictive* step was robust enough (or even overcautious). This happens in experiments with the highest value of q, where longer processing times are used in the *baseline* schedule and, consequently, the probability of the operations to last longer at the execution phase is low. On the other side, whenever the value of the AQD for PR is lower than the *P-only* one, then the *reactive* step improves the schedule by reacting to the occurred changes.

It is possible to claim that the PR approach performs better when the dimension of the problem under study is smaller in terms of both the number of operations and jobs. Taking as an example the experiments with 5, 7 and 9 operations with uniform-distributed processing times (Tables 13-15 in Appendix), it is possible to verify that, with the increasing of the number of operations i) the AQD increases and ii) the difference between PR and *P-only* decreases. Consider the results with $\lambda = 0.3$ for the instances with 5 and 7 operations summarized in Figures 6a - 6b for the uniform case. The performance of the PR approach is much better than the *P-only* one in the 5-operation case (Figure 6a), and slightly better than the *P-only* one in the 7-operation case (Figure 6b). This behaviour can be explained with the fact that, as the number of operations



(a) In the 5-operation case, the AQD of the PR is always lower than the *P-only* one.

(b) In the 7-operation case, the AQD of the PR is slightly lower than the *P-only* one.

Figure 6: Comparison between the PR and *P-only* approaches in the 5- and 7-operations cases with uniformly distributed processing times.

are small, every single decision has a higher impact on the objective function and, hence, being able to react at the right time as an important impact. On the contrary, with a larger number of operations, the possibility to absorb possible deviations within a schedule without the need of modifying it is more likely to occur.

On the other side, the parameter λ (influencing the generation of the processing times distribution) seems not to have any impact on the results (namely on the performance of the PR approach). In fact, in some cases, a smaller value for λ entails better performance than with a larger value (e.g., for the 5-operation instances in Table 13, in Appendix); in other cases, this behaviour is not present (e.g., for the 7-operation instances in Table 14, in Appendix). In Table 4, a comparison is provided considering the maximum a and the minimum value of the AQD in respect to the values of λ for both the 5 and 7 operations cases with uniform-distributed processing times. It is possible to see that for the 5-operation case both the minimum and maximum values increase with the increasing of λ; this behaviour is not true for the 7-operation case.

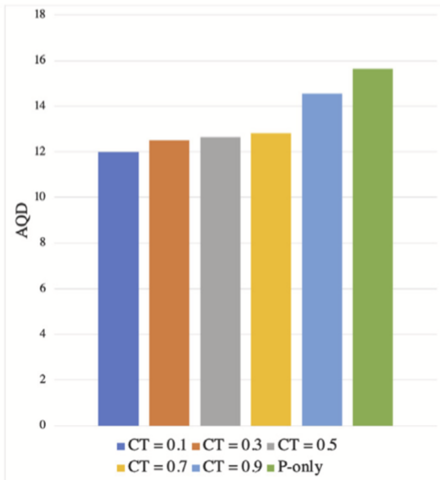
λ	5 operations		7 operations	
	max AQD	min AQD	max AQD	min AQD
0.3	3.589	2.713	104.043	102.595
0.5	4.909	3.095	66.419	61.229
0.9	14.901	11.829	97.585	90.206
0.95	17.910	13.902	107.323	101.653

Table 4: Comparison of the impact of λ on the PR performances for the 5- and 7-operation cases with uniformly distributed processing times.

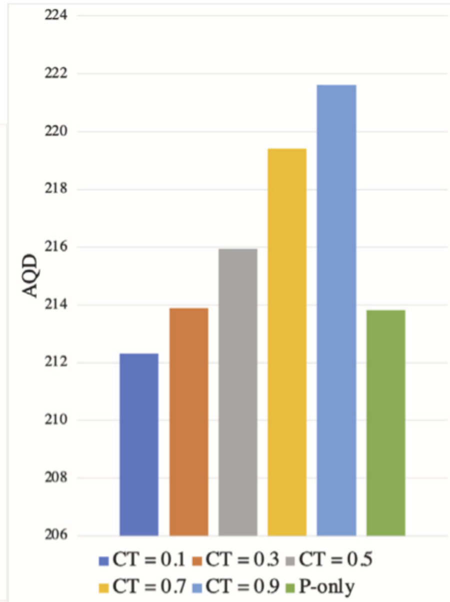
Moreover, also the distribution of the processing times has an impact on the performance. Indeed, all the considered approaches perform better in the instances with uniformly distributed processing times, in comparison with triangularly distributed ones. This is an expected result for robust approaches. In fact, as the difference between extreme scenarios and expected values increases, it is more difficult to protect the schedule. With regards to the shape of the two distributions, given a quantile, i.e., $q = 0.9$, the range of possible values in the remaining 0.1-tail is larger with triangular distributions than with uniform ones, and consequently the impact on the schedule can be higher.

Beyond these considerations, it was not possible to identify a quantitative model explaining the behaviour of the two approaches respect to a variation of the considered parameters, nevertheless, few additional remarks can be provided.

In some cases, the PR improves the *P-only* schedule for all the combinations of q and CT, e.g., in the 5-operation case in Table 13, in Appendix (for which the case with $\lambda = 0.9$ and $q = 0.9$ is reported in Figure 7a); in other cases, the PR is beneficial in a subset of these combinations of parameters only, e.g., in the 9-operation case in Table 15, in Appendix (for which the case with $\lambda = 0.95$ and $q = 0.9$ is reported in Figure 7b).



(a) In the 5-operation case with uniform distributions and $\lambda = 0.9$ and $q = 0.9$, the PR approach performs better in terms of the AQD for any value of CT.



(b) In the 9-operation case with uniform distributions and $\lambda = 0.95$ and $q = 0.9$, the PR approach performs better in terms of the AQD only for $CT = 0.1$.

Figure 7: Comparison between the *PR* and *P-only* approaches in the 5- and 9-operations cases with uniformly distributed processing times.

This entails a difficulty in tuning the parameters of the approach to obtain the best performance in terms of robustness. In fact, in the first case, it is possible to match the aversion to risk and the agility to react, by tuning the parameter q and CT respectively. For example, in the case of a very fast handling robot, the user can select a low CT or, in the case the uses prefers to have a more stable schedule (minimize the number of modifications operated in the *reaction* step), an higher q can be selected. In the second case, the only viable approach to tune the parameters of the approach is through a testing phase during or before the operating phase of the assembly line, to provide the best benefit.

A second set of experiments has been carried out to test the behaviour of the approach in case of extreme and rare cases where the duration of the operations can be longer. We generated 7-operation instances whose processing times are modelled through triangular distributions with longer right tails ($\lambda \in \{1.5, 1.8\}$). From an industrial point of view, these distributions model cases where the operations, e.g., the loading of part or a component, have a small variability but, in case a problem arises then their duration is much longer, although these events have a small occurrence probability. Moreover, we limited the sampling phase for the testing to values p_i in the rightmost 0.1 tail, i.e., $\mathbb{P}[\tilde{p}_i = p_i] \geq 0.9$, with $i = [0, \ldots, m]$. The results of these experiments are reported in Table 5.

Lambda	q	PR Change Threshold 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
1.5	0.1	2.818	2.871	2.731	2.772	2.818	3.265	3.986
	0.3	2.146	2.249	2.302	2.340	2.492	2.730	2.904
	0.5	2.180	1.756	1.762	1.860	1.905	2.140	2.336
	0.7	2.308	2.309	2.431	2.319	2.408	2.792	2.988
	0.9	2.145	2.157	2.213	2.290	2.434	2.213	4.960
	q	**PR** *Change Threshold* 0.1	0.3	0.5	0.7	0.9	**P-only**	**FCFS/FRO**
1.8	0.1	3.210	3.265	3.265	3.219	3.215	3.380	4.653
	0.3	3.300	3.290	3.239	3.239	3.239	3.424	4.578
	0.5	3.508	3.495	3.437	3.437	3.437	3.613	3.989
	0.7	3.970	3.970	3.970	3.313	3.313	3.444	3.481
	0.9	3.253	3.253	3.253	3.253	3.253	4.147	4.420

Table 5: *AQD* between the *cdf* of the optimal solution for the *predictive*-only (*P-only*) and the *predictive-reactive* (*PR*) approaches considering only extreme cases.

These results show that the *PR* approach is always beneficial in the extreme cases. For both values of λ, the *AQD* in the *PR* case is always smaller or equal to the ones related to alternative approaches, with a clear impact of parameter CT. In some cases, e.g., with $\lambda = 1.5$ and a high value of q, the selection of CT significantly affects the results that vary from 1.756 to 2.180, in the case with $q = 0.5$.

Hence, the *PR* approach is significantly relevant to protect the schedule in case of extreme events with low occurrence probability but a consistent impact on the schedule.

5.3 Execution time

The presented approach has been implemented on *MATLAB* version $R2015a$ and executed on a laptop with an Intel Core i5 processor at $2.4GHz$ and $8GB$ RAM. The computation times (in seconds) for the different instances are reported in Table 6, with the details of the time spent for i) the *predictive* step, ii) the *sensitivity* evaluation and iii) the *reactive* step. It is possible to see that the most time consuming phases are the *predictive* and *sensitivity* steps, intended to be operated *off-line*, before the execution of the schedule. On the contrary, the *reactive* step always takes less than 1 second to be executed, with a maximum value of 0.177 seconds for the 9 uniform-distributed operations case. Hence, the execution of the *reactive* step is compatible with the *on-line* utilization of the proposed approach.

	5 **jobs**			10 **jobs**		
	predictive	*sensitivity*	*reaction*	*predictive*	*sensitivity*	*reaction*
5 uniform	0.080	0.370	0.0154	0.250	2.520	0.045
5 triangular	0.070	0.240	0.0168	0.220	2.310	0.029
7 uniform	0.580	1.390	0.0195	13.730	9.780	0.087
7 triangular	1.140	1.710	0.0294	11.420	9.790	0.069
9 uniform	15.510	4.880	0.0434	836.670	40.040	0.177
9 triangular	0.990	2.280	0.0372	794.400	34.080	0.152

Table 6: Computation times.

Besides the *reaction* time, also the time spent to update the list of thresholds every time a reaction is performed has to be considered, with the aim to assess the possibility of running the proposed *reactive* approach in real-time. The results for the update time are in line with the sensitivity ones. When this time is very low, like for 5-job 7-uniform-operation case, it can be executed on-line, during the processing of the batch. In other cases, when the sensitivity analysis requires more calculations, e.g., the 34 seconds measured for the 10-job 9-triangular-operation case, an on-line execution of the approach is not feasible. In these cases, two approaches can be pursued: the first one is to execute parallel computing to reduce the computational time since the calculations for the sensitivity analysis are independent from each other; the second one is to set up sensitivity tables during the off-line computation, before the execution of the process. Through the second approach, during the on-line process, the value stored will be directly used when a *reaction* is needed.

6 Conclusions

In this paper, we presented a *predictive-reactive flow shop* scheduling approach tailored to production lines with a reduced set of shared resources and stochastic processing times. The aim of the approach is to provide a robust schedule considering a batch of repeated jobs of the same type to be scheduled through two steps, the first one able to identify a *baseline* schedule before the execution of the process and the second one able to verify the optimality of the schedule during the execution of the process and to react to the occurrence of unexpected events, namely a deviation from the expected processing times. The industrial motivation for the proposed approach stems from the need of managing modern reconfigurable and automated assembly lines where, due to the intrinsic impossibility to balance them, the scheduling of operations competing for shared resources (e.g., a handling robot or a transporter) is a relevant problem. The approach has been extensively tested on instances generated from a real industrial case in the automotive sector addressing the assembling of a door.

We demonstrated how the use of the approach could be beneficial in a wide range of cases and particularly valuable when coping with extreme and rare events, i.e., when the processing time of an operation could deviate strongly from the expected values although with a very low occurrence probability. The approach has been tested on instances with 9 operations at most, also demonstrating computation times compatible with the declared on-line utilisation.

Future development will investigate methods for selecting the quantile q to be used in the *predictive* step grounding on the characteristics of the process and the system, as well as the application to different classes of assembly systems.

7 Acknowledgments

We thank PhD Morteza Davari for the hours of fruitful discussions on the problem formulation.

References

[1] Creemers, S., 2015. Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. Journal of Scheduling 18 (3), 263–273.

[2] Davari, M., Demeulemeester, E., April 2016. The proactive and reactive resource-constrained project scheduling problem, note.

[3] Demeulemeester, E. L., Herroelen, W. S., 1992. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. Management Science 38, 1803–1818.

[4] European Aluminium, 2016. The aluminium automotive manual. URL https://www.european-aluminium.eu/media/1543/1_aam_body-structures.pdf

[5] Fernandez, A. A., 1995. The optimal solution to the resource-constrained project scheduling problem with stochastic task durations, unpublished Doctoral Dissertation, University of Central Florida.

[6] Fernandez, A. A., Armacost, R., Pet-Edwards, J., 1998. Understanding simulation solutions to resource-constrained project scheduling problems with stochastic task durations. Engineering Management Journal 10, 5–13.

[7] Fernandez, A. A., Armacost, R. L., 1996. The role of the non-anticipativity constraint in commercial software for stochastic project scheduling. Computers and Industrial Engineering 31, 233–236.

[8] Golenko-Ginsburg, D., Gonik, A., 1997. Stochastic network project scheduling with non-consumable limited resources. International Journal of Production Economics 48, 29–37.

[9] Herroelen, W., Leus, R., 2004. The construction of stable project baseline schedules. European Journal of Operational Research 153 (3), 550–565.

[10] Igelmund, G., Radermacher, F. J., 1983. Algorithmic approaches to preselective strategies for stochastic scheduling problems. Networks 13, 29–48.

[11] Igelmund, G., Radermacher, F. J., 1983. Preselective strategies for the optimization of stochastic project networks under resource constraints. Networks 13, 1–28.

[12] Lamas, P., Demeulemeester, E., 2016. A purely proactive scheduling procedure for the resource- constrained project scheduling problem with stochastic activity durations. Journal of Scheduling 19 (4), 409–428.

[13] Leus, R., Herroelen, W., 2005. The complexity of machine scheduling for stability with a single disrupted job. Operations Research Letters 33 (2), 151–156.

[14] Möhring, R. H., Skutella, M., Stork, F., 2000. Scheduling with and/or precedence constraints. Tech. rep., Technische Universität Berlin, Department of Mathematics, Germany, technical Report 689/2000.

[15] Pet-Edwards, J., Mollaghasemi, M., 1996. A simulation and genetic algorithm approach to stochastic resource-constrained project scheduling. In: Southcon Conference Record, IEEE, Pascataway, NJ. pp. 333–338.

[16] Pinedo, M. L., 1982. Minimizing the expected makespan in stochastic flow shops. Operations Research 30, 148–162.

[17] Pinedo, M. L., 2008. Scheduling: Theory, Algorithms, and Systems. Springer.

[18] Radermacher, F. J., 1985. Scheduling of project networks. Annals of Operations Research 4, 227–252.

[19] Stork, F., 2000. Branch-and-bound algorithms for stochastic resource-constrained project scheduling. Tech. rep., Technische Universität Berlin, research Report No. 702/2000.

[20] Tsai, Y. W., Gemmill, D. D., 1996. Using a simulated annealing algorithm to schedule activities of resource-constrained projects. Tech. rep., Iowa State University, working Paper No. 96-124.

[21] Tsai, Y. W., Gemmill, D. D., 1998. Using tabu search to schedule activities of stochastic resource-constrained projects. European Journal of Operational Research 111, 129–141.

8 Appendix

5 Jobs

Lambda	q	Change Threshold 0.1 Q10	0.1 Q50	0.1 Q90	0.3 Q10	0.3 Q50	0.3 Q90	0.5 Q10	0.5 Q50	0.5 Q90	0.7 Q10	0.7 Q50	0.7 Q90	0.9 Q10	0.9 Q50	0.9 Q90
0.3	0.1	0.7	1.1	0.0	0.7	1.1	0.0	0.7	1.1	0.0	1.0	1.1	0.0	0.6	1.2	0.0
	0.3	1.1	2.7	0.0	1.1	2.7	0.0	1.1	2.7	0.0	0.9	2.2	0.0	0.5	1.7	0.0
	0.5	1.9	4.0	0.0	1.9	4.0	0.0	1.9	4.0	0.0	1.5	3.6	0.0	1.4	2.6	0.0
	0.7	2.0	2.6	0.0	1.9	2.6	0.0	1.8	2.4	0.0	1.0	1.8	0.0	0.4	1.3	0.0
	0.9	1.4	1.9	0.0	1.0	1.9	0.0	1.0	1.9	0.0	0.5	0.9	0.0	0.3	0.7	0.0
0.5	0.1	0.8	0.1	3.2	1.0	0.1	0.8	0.9	0.6	0.8	0.9	0.5	0.8	0.8	0.8	0.8
	0.3	1.0	0.6	0.0	0.5	1.1	0.0	0.4	1.1	0.0	0.4	1.1	0.0	0.4	0.5	0.0
	0.5	0.1	0.2	1.8	0.2	0.2	0.5	0.2	0.2	0.5	0.2	0.2	0.5	0.5	0.0	0.5
	0.7	1.0	0.6	0.0	1.0	0.6	0.0	1.0	1.0	0.0	1.0	1.0	0.0	0.9	0.3	0.0
	0.9	1.6	0.7	0.0	1.6	0.7	0.0	1.6	0.7	0.0	1.1	0.7	0.0	0.1	0.1	0.0
0.9	0.1	0.2	3.6	0.0	0.6	3.6	0.0	0.6	3.8	0.0	0.6	4.0	0.0	0.4	3.6	0.0
	0.3	2.8	2.8	0.0	2.8	2.0	0.0	2.4	2.0	0.0	2.4	1.8	0.0	2.4	1.8	0.0
	0.5	2.4	0.8	0.0	1.8	0.8	0.0	1.8	0.8	0.0	1.8	0.6	0.0	0.4	0.4	0.0
	0.7	2.6	2.8	0.0	2.6	2.4	0.0	2.6	2.4	0.0	1.6	2.4	0.0	0.6	0.4	0.0
	0.9	2.4	1.6	0.0	2.6	1.6	0.0	2.6	1.2	0.0	1.8	0.4	0.0	0.6	0.0	0.0
0.95	0.1	3.0	2.8	0.0	3.0	2.8	0.0	3.8	3.0	0.0	3.8	3.0	0.0	1.6	0.8	0.0
	0.3	2.4	0.6	0.0	2.4	0.6	0.0	2.4	0.8	0.0	2.2	0.8	0.0	3.2	5.0	0.0
	0.5	1.4	1.0	0.0	2.8	2.0	0.0	3.0	1.0	0.0	5.2	1.2	0.0	8.2	3.6	0.0
	0.7	2.6	0.6	0.0	2.0	0.8	0.0	1.6	0.0	0.0	1.6	3.2	0.0	4.4	5.2	0.0
	0.9	3.4	1.0	0.0	3.0	0.2	0.0	0.4	2.4	0.0	1.2	5.2	0.0	2.0	5.4	0.0

10 Jobs

Lambda	q	Change Threshold 0.1 Q10	0.1 Q50	0.1 Q90	0.3 Q10	0.3 Q50	0.3 Q90	0.5 Q10	0.5 Q50	0.5 Q90	0.7 Q10	0.7 Q50	0.7 Q90	0.9 Q10	0.9 Q50	0.9 Q90
0.3	0.1	0.1	0.1	0.3	0.3	0.3	0.3	0.5	0.5	0.2	0.6	0.6	0.2	0.5	0.7	0.2
	0.3	0.2	0.6	0.4	0.4	0.8	0.1	0.6	1.1	0.2	0.5	1.0	0.2	0.4	0.8	0.2
	0.5	0.7	1.3	0.2	0.8	1.4	0.2	1.0	1.5	0.2	0.8	1.4	0.2	0.8	1.0	0.2
	0.7	0.8	1.0	0.0	1.0	1.2	0.0	1.0	1.1	0.0	0.7	0.9	0.0	0.5	0.7	0.0
	0.9	0.7	0.7	0.0	0.7	0.8	0.0	0.7	0.8	0.0	0.5	0.5	0.0	0.2	0.5	0.0
0.5	0.1	0.6	0.5	0.9	0.7	0.5	0.2	0.7	0.6	0.2	0.7	0.6	0.2	0.6	0.6	0.2
	0.3	0.7	0.5	0.0	0.6	0.7	0.0	0.5	0.7	0.0	0.5	0.7	0.0	0.3	0.5	0.0
	0.5	0.9	0.8	0.5	0.8	0.8	0.1	0.8	0.8	0.1	0.8	0.8	0.1	0.7	0.7	0.1
	0.7	1.0	0.8	0.0	1.0	0.8	0.0	1.0	0.9	0.0	1.0	0.9	0.0	1.0	0.6	0.0
	0.9	1.2	0.9	0.0	1.2	0.9	0.0	1.2	0.9	0.0	1.1	0.9	0.0	0.7	0.7	0.0
0.9	0.1	1.1	1.4	0.0	1.2	1.5	0.0	1.3	1.6	0.0	1.2	1.7	0.0	1.1	1.3	0.1
	0.3	2.1	1.2	0.0	2.1	1.1	0.0	2.1	1.2	0.0	2.2	1.0	0.0	1.7	0.9	0.0
	0.5	1.3	1.2	0.2	1.2	1.2	0.2	1.2	1.1	0.1	1.1	0.9	0.1	0.9	0.9	0.1
	0.7	2.1	1.0	0.9	2.1	0.9	0.2	2.0	1.1	0.2	1.6	0.7	0.2	1.5	0.3	0.2
	0.9	1.1	1.1	0.1	1.0	1.1	0.0	1.0	0.9	0.0	0.9	0.8	0.0	0.6	0.8	0.0
0.95	0.1	2.5	2.1	0.0	2.5	2.1	0.0	2.7	2.1	0.0	2.7	2.2	0.0	2.2	1.8	0.0
	0.3	1.9	1.6	0.0	2.0	1.6	0.0	2.0	1.6	0.0	2.1	1.6	0.0	1.1	0.7	0.0
	0.5	1.7	1.8	0.0	1.4	2.0	0.0	1.4	1.8	0.0	1.0	1.3	0.0	0.4	0.9	0.0
	0.7	2.4	1.5	0.0	2.3	1.6	0.0	2.2	1.4	0.0	1.6	0.8	0.0	1.0	0.4	0.0
	0.9	2.3	1.8	0.0	2.2	1.5	0.0	1.7	1.1	0.0	1.4	0.6	0.0	1.2	0.5	0.0

Table 7: Difference between the 10th, 50th and 90th quantiles *predictive*-only's *cdf* and the *predictive-reactive cdf*, for the cases with 5 and 10 jobs, and 5 uniformly distributed processing times.

5 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*
0.3	0.1	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.2	0.0
	0.3	0.2	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0
	0.5	1.0	1.2	0.0	1.0	1.2	0.0	1.0	1.2	0.0	1.0	1.2	0.0	0.5	1.0	0.0
	0.7	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0
	0.9	0.2	0.4	0.0	0.2	0.4	0.0	0.2	0.4	0.0	0.2	0.4	0.0	0.2	0.2	0.0
0.5	0.1	0.2	0.2	0.3	0.2	0.2	0.1	0.2	0.2	0.1	0.0	0.0	0.1	0.0	0.0	0.1
	0.3	0.6	0.4	0.2	0.6	0.0	0.2	0.2	0.0	0.6	0.0	0.0	0.2	0.0	0.0	0.2
	0.5	0.6	0.4	0.0	0.6	0.4	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.7	1.6	0.4	0.0	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.9	0.8	0.2	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.9	0.1	1.6	1.6	0.0	0.8	1.6	0.0	0.2	0.6	0.0	0.2	0.0	0.0	0.2	0.4	0.0
	0.3	1.2	1.2	0.0	1.0	0.8	0.0	0.2	0.4	0.0	0.2	0.4	0.0	0.0	0.4	0.0
	0.5	1.6	1.6	0.0	1.4	1.2	0.0	0.8	0.6	0.0	0.2	0.8	0.0	0.2	0.8	0.0
	0.7	1.8	0.8	0.0	0.8	0.2	0.0	0.8	0.6	0.0	0.8	0.6	0.0	0.6	0.6	0.0
	0.9	0.4	1.6	0.0	0.6	0.6	0.0	0.6	0.6	0.0	0.8	0.8	0.0	0.8	1.0	0.0
0.95	0.1	0.8	0.0	0.0	0.8	0.2	0.0	0.8	0.4	0.0	0.2	0.4	0.0	0.2	0.4	0.0
	0.3	0.0	0.2	0.0	0.4	0.0	0.0	0.4	0.2	0.0	0.4	0.0	0.0	0.4	0.0	0.0
	0.5	0.2	0.4	4.0	0.2	0.8	1.0	0.2	0.8	1.0	0.2	0.6	1.0	0.0	0.6	1.0
	0.7	0.0	0.2	0.0	0.0	0.4	0.0	0.0	0.4	0.0	0.0	0.2	0.0	0.0	0.2	0.0
	0.9	0.0	0.6	0.0	0.0	0.6	0.0	0.0	0.6	0.0	0.0	0.6	0.0	0.0	0.6	0.0

10 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*
0.3	0.1	0.1	0.3	0.0	0.1	0.3	0.0	0.1	0.3	0.0	0.1	0.3	0.0	0.1	0.2	0.0
	0.3	0.2	0.1	0.0	0.2	0.1	0.0	0.2	0.1	0.0	0.2	0.1	0.0	0.1	0.1	0.0
	0.5	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.2	0.3	0.0
	0.7	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0
	0.9	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.1	0.0
0.5	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.3	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
	0.5	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.7	0.3	0.1	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.9	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.9	0.1	0.6	0.5	1.5	0.4	0.4	0.5	0.3	0.3	0.1	0.3	0.2	1.0	0.3	0.2	1.0
	0.3	0.5	0.5	0.0	0.5	0.4	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0
	0.5	0.6	0.4	0.5	0.5	0.4	1.3	0.4	0.3	0.3	0.3	0.0	0.7	0.3	0.0	0.7
	0.7	0.4	0.3	0.0	0.3	0.1	0.1	0.3	0.1	0.0	0.3	0.1	0.0	0.3	0.1	0.0
	0.9	0.3	0.5	1.0	0.1	0.1	0.3	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.1	0.4
0.95	0.1	0.3	0.9	0.7	0.3	0.8	0.7	0.3	0.7	0.3	0.2	0.3	0.9	0.0	0.0	0.9
	0.3	0.4	0.8	0.5	0.2	0.7	0.5	0.2	0.6	0.5	0.0	0.1	0.5	0.0	0.0	0.8
	0.5	0.3	0.6	1.7	0.3	0.4	0.3	0.3	0.3	0.3	0.1	0.0	0.3	0.0	0.1	2.0
	0.7	0.3	0.9	1.1	0.3	0.6	1.1	0.1	0.2	0.4	0.0	0.1	1.0	0.0	0.0	1.0
	0.9	0.3	0.6	0.8	0.2	0.3	0.7	0.1	0.0	0.5	0.0	0.1	0.5	0.0	0.1	0.5

Table 8: Difference between the 10th, 50th and 90th quantiles *predictive*-only's *cdf* and the *predictive-reactive cdf*, for the cases with 5 and 10 jobs, and 5 triangularly distributed processing times.

5 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*
0.3	0.1	0.1	0.1	0.3	0.3	0.3	0.3	0.5	0.5	0.2	0.6	0.6	0.2	0.5	0.7	0.2
	0.3	0.2	0.6	0.4	0.4	0.8	0.1	0.6	1.1	0.2	0.5	1.0	0.2	0.4	0.8	0.2
	0.5	0.7	1.3	0.2	0.8	1.4	0.2	1.0	1.5	0.2	0.8	1.4	0.2	0.8	1.0	0.2
	0.7	0.8	1.0	0.0	1.0	1.2	0.0	1.0	1.1	0.0	0.7	0.9	0.0	0.5	0.7	0.0
	0.9	0.7	0.7	0.0	0.7	0.8	0.0	0.7	0.8	0.0	0.5	0.5	0.0	0.2	0.5	0.0
0.5	0.1	0.6	0.5	0.9	0.7	0.5	0.2	0.7	0.6	0.2	0.7	0.6	0.2	0.6	0.6	0.2
	0.3	0.7	0.5	0.0	0.6	0.7	0.0	0.5	0.7	0.0	0.5	0.7	0.0	0.3	0.5	0.0
	0.5	0.9	0.8	0.5	0.8	0.8	0.1	0.8	0.8	0.1	0.8	0.8	0.1	0.7	0.7	0.1
	0.7	1.0	0.8	0.0	1.0	0.8	0.0	1.0	0.9	0.0	1.0	0.9	0.0	1.0	0.6	0.0
	0.9	1.2	0.9	0.0	1.2	0.9	0.0	1.2	0.9	0.0	1.1	0.9	0.0	0.7	0.7	0.0
0.9	0.1	1.1	1.4	0.0	1.2	1.5	0.0	1.3	1.6	0.0	1.2	1.7	0.0	1.1	1.3	0.1
	0.3	2.1	1.2	0.0	2.1	1.1	0.0	2.1	1.2	0.0	2.2	1.0	0.0	1.7	0.9	0.0
	0.5	1.3	1.2	0.2	1.2	1.2	0.2	1.2	1.1	0.1	1.1	0.9	0.1	0.9	0.9	0.1
	0.7	2.1	1.0	0.9	2.1	0.9	0.2	2.0	1.1	0.2	1.6	0.7	0.2	1.5	0.3	0.2
	0.9	1.1	1.1	0.1	1.0	1.1	0.0	1.0	0.9	0.0	0.9	0.8	0.0	0.6	0.8	0.0
0.95	0.1	2.5	2.1	0.0	2.5	2.1	0.0	2.7	2.1	0.0	2.7	2.2	0.0	2.2	1.8	0.0
	0.3	1.9	1.6	0.0	2.0	1.6	0.0	2.0	1.6	0.0	2.1	1.6	0.0	1.1	0.7	0.0
	0.5	1.7	1.8	0.0	1.4	2.0	0.0	1.4	1.8	0.0	1.0	1.3	0.0	0.4	0.9	0.0
	0.7	2.4	1.5	0.0	2.3	1.6	0.0	2.2	1.4	0.0	1.6	0.8	0.0	1.0	0.4	0.0
	0.9	2.3	1.8	0.0	2.2	1.5	0.0	1.7	1.1	0.0	1.4	0.6	0.0	1.2	0.5	0.0

10 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*
0.3	0.1	0.4	0.7	0.5	0.0	0.0	0.4	0.9	1.0	0.3	1.3	1.2	0.3	1.3	1.2	0.3
	0.3	0.1	0.4	1.1	0.4	0.4	0.2	1.1	1.2	0.5	1.1	1.2	0.5	1.1	1.2	0.5
	0.5	0.3	0.5	0.6	0.5	0.2	0.0	1.0	1.0	0.2	1.0	1.0	0.2	1.0	1.0	0.2
	0.7	0.3	0.3	0.0	0.9	1.0	0.0	1.2	1.0	0.0	1.2	1.0	0.0	1.2	1.0	0.0
	0.9	1.0	0.8	0.0	1.3	1.0	0.0	1.3	1.0	0.0	1.3	1.0	0.0	1.3	1.0	0.0
0.5	0.1	1.6	1.5	0.0	1.6	1.5	0.0	1.6	1.5	0.0	1.6	1.5	0.0	1.6	1.5	0.0
	0.3	1.4	1.4	0.0	1.4	1.4	0.0	1.4	1.4	0.0	1.4	1.4	0.0	1.4	1.4	0.0
	0.5	2.5	2.1	0.0	2.5	2.1	0.0	2.5	2.1	0.0	2.5	2.1	0.0	2.5	2.1	0.0
	0.7	2.4	2.2	0.0	2.4	2.2	0.0	2.4	2.2	0.0	2.4	2.2	0.0	2.4	2.2	0.0
	0.9	2.7	2.3	0.0	2.7	2.3	0.0	2.7	2.3	0.0	2.7	2.3	0.0	2.7	2.3	0.0
0.9	0.1	2.1	3.3	0.3	2.1	3.3	0.3	2.1	3.3	0.3	2.1	3.4	0.3	2.0	3.1	1.1
	0.3	3.0	2.6	0.2	3.2	3.0	0.2	3.4	2.7	0.1	3.0	3.4	0.2	2.8	2.9	0.2
	0.5	2.2	2.1	0.0	2.2	2.2	0.0	2.4	2.4	0.0	2.3	1.9	0.0	2.3	2.0	0.0
	0.7	3.3	2.8	0.2	3.4	2.8	0.2	3.0	2.3	0.2	2.9	2.3	0.3	2.9	2.3	0.3
	0.9	3.8	2.9	0.0	3.6	2.5	0.0	3.3	2.5	0.0	3.3	2.5	0.0	3.4	2.5	0.0
0.95	0.1	4.6	4.2	0.0	4.6	4.2	0.0	4.6	4.4	0.0	4.6	4.3	0.0	4.6	4.4	0.0
	0.3	5.0	4.1	0.0	5.0	4.1	0.0	5.0	4.1	0.0	4.9	4.1	0.0	5.2	4.1	0.0
	0.5	5.3	3.9	0.0	5.3	3.9	0.0	5.3	3.9	0.0	5.3	3.9	0.0	5.3	4.0	0.0
	0.7	5.2	3.9	0.0	5.2	3.9	0.0	5.2	3.9	0.0	5.2	3.9	0.0	5.2	3.9	0.0
	0.9	4.8	4.4	0.0	4.8	4.4	0.0	4.8	4.4	0.0	4.8	4.4	0.0	4.9	4.4	0.0

Table 9: Difference between the 10th, 50th and 90th quantiles *predictive*-only's *cdf* and the *predictive-reactive cdf*, for the cases with 5 and 10 jobs, and 7 uniformly distributed processing times.

5 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	Q50	Q90	Q10	Q50	Q90	Q10	Q50	Q90	Q10	Q50	Q90	Q10	Q50	Q90
0.3	0.1	0.1	0.3	0.0	0.1	0.3	0.0	0.1	0.3	0.0	0.1	0.3	0.0	0.1	0.2	0.0
	0.3	0.2	0.1	0.0	0.2	0.1	0.0	0.2	0.1	0.0	0.2	0.1	0.0	0.1	0.1	0.0
	0.5	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.2	0.3	0.0
	0.7	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0
	0.9	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.1	0.0
0.5	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.3	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
	0.5	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.7	0.3	0.1	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.9	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.9	0.1	0.6	0.5	1.5	0.4	0.4	0.5	0.3	0.3	0.1	0.3	0.2	1.0	0.3	0.2	1.0
	0.3	0.5	0.5	0.0	0.5	0.4	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0
	0.5	0.6	0.4	0.5	0.5	0.4	1.3	0.4	0.3	0.3	0.3	0.0	0.7	0.3	0.0	0.7
	0.7	0.4	0.3	0.0	0.3	0.1	0.1	0.3	0.1	0.0	0.3	0.1	0.0	0.3	0.1	0.0
	0.9	0.3	0.5	1.0	0.1	0.1	0.3	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.1	0.4
0.95	0.1	0.3	0.9	0.7	0.3	0.8	0.7	0.3	0.7	0.3	0.2	0.3	0.9	0.0	0.0	0.9
	0.3	0.4	0.8	0.5	0.2	0.7	0.5	0.2	0.6	0.5	0.0	0.1	0.5	0.0	0.0	0.8
	0.5	0.3	0.6	1.7	0.3	0.4	0.3	0.3	0.3	0.3	0.1	0.0	0.3	0.0	0.1	2.0
	0.7	0.3	0.9	1.1	0.3	0.6	1.1	0.1	0.2	0.4	0.0	0.1	1.0	0.0	0.0	1.0
	0.9	0.3	0.6	0.8	0.2	0.3	0.7	0.1	0.0	0.5	0.0	0.1	0.5	0.0	0.1	0.5

10 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	Q50	Q90	Q10	Q50	Q90	Q10	Q50	Q90	Q10	Q50	Q90	Q10	Q50	Q90
0.3	0.1	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0	0.3	0.3	0.0
	0.3	0.6	0.3	0.0	0.6	0.3	0.0	0.6	0.3	0.0	0.6	0.3	0.0	0.6	0.3	0.0
	0.5	0.4	0.4	0.0	0.4	0.4	0.0	0.4	0.4	0.0	0.4	0.4	0.0	0.4	0.4	0.0
	0.7	0.4	0.3	0.0	0.4	0.3	0.0	0.4	0.3	0.0	0.4	0.3	0.0	0.4	0.3	0.0
	0.9	0.4	0.3	0.0	0.4	0.3	0.0	0.4	0.3	0.0	0.4	0.3	0.0	0.4	0.3	0.0
0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.9	0.1	0.5	0.5	0.0	0.5	0.5	0.0	0.5	0.6	0.2	0.6	0.8	0.0	0.6	0.8	0.0
	0.3	1.0	0.8	0.0	1.0	0.8	0.0	1.0	0.8	0.0	1.0	0.8	0.0	1.0	0.8	0.0
	0.5	0.8	0.6	1.6	0.9	0.6	0.3	0.9	0.6	0.4	0.9	0.8	0.4	0.9	0.8	0.4
	0.7	1.0	0.5	0.8	1.0	0.5	3.8	1.1	0.7	1.0	1.1	0.7	1.0	1.1	0.7	1.0
	0.9	0.8	0.7	0.5	0.8	0.7	0.5	0.9	0.8	0.4	0.9	0.8	0.4	0.9	0.9	0.4
0.95	0.1	0.8	1.8	2.3	0.7	1.5	2.3	0.6	1.2	0.8	0.2	0.3	1.6	0.0	0.0	3.8
	0.3	0.6	1.6	1.8	0.4	1.2	1.8	0.4	0.8	1.8	0.1	0.1	2.7	0.0	0.0	2.7
	0.5	0.4	1.5	1.6	0.4	1.3	1.6	0.3	0.8	1.6	0.1	0.2	2.0	0.0	0.0	2.6
	0.7	0.4	1.3	3.4	0.3	1.0	2.6	0.1	0.3	2.0	0.1	0.1	2.0	0.0	0.0	2.0
	0.9	0.6	1.0	4.5	0.4	0.6	0.7	0.1	0.2	1.7	0.1	0.1	1.7	0.1	0.1	1.7

Table 10: Difference between the 10th, 50th and 90th quantiles *predictive*-only's *cdf* and the *predictive-reactive cdf*, for the cases with 5 and 10 jobs, and 7 triangularly distributed processing times.

5 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*
0.3	0.1	0.4	0.7	0.6	0.2	0.2	0.6	0.6	0.3	0.4	0.8	0.8	0.4	0.8	0.8	0.4
	0.3	0.3	0.6	1.1	0.1	0.2	0.3	0.7	0.7	0.4	0.7	0.7	0.4	0.7	0.7	0.4
	0.5	0.2	0.1	0.6	0.6	0.3	0.6	0.9	0.5	0.4	0.9	0.5	0.4	0.9	0.5	0.4
	0.7	0.4	0.3	0.0	1.1	0.8	0.0	1.1	0.8	0.0	1.1	0.8	0.0	1.1	0.8	0.0
	0.9	0.6	0.2	0.0	0.8	0.6	0.0	0.8	0.6	0.0	0.8	0.6	0.0	0.8	0.6	0.0
0.5	0.1	0.9	0.9	0.0	0.9	0.9	0.0	0.9	0.9	0.0	0.9	0.9	0.0	0.9	0.9	0.0
	0.3	0.9	0.8	0.0	0.9	0.8	0.0	0.9	0.8	0.0	0.9	0.8	0.0	0.9	0.8	0.0
	0.5	1.8	1.5	0.0	1.8	1.5	0.0	1.8	1.5	0.0	1.8	1.5	0.0	1.8	1.5	0.0
	0.7	1.6	1.3	0.0	1.6	1.3	0.0	1.6	1.3	0.0	1.6	1.3	0.0	1.6	1.3	0.0
	0.9	1.6	1.4	0.0	1.6	1.4	0.0	1.6	1.4	0.0	1.6	1.4	0.0	1.6	1.4	0.0
0.9	0.1	2.3	1.6	0.1	2.3	1.7	0.1	2.4	1.8	0.1	2.3	2.0	0.1	2.1	1.4	0.2
	0.3	3.3	1.5	0.0	3.3	1.5	0.0	3.3	1.6	0.0	3.6	1.4	0.0	2.6	1.2	0.0
	0.5	1.8	2.1	0.4	1.7	2.1	0.4	1.8	1.9	0.3	1.6	1.6	0.3	1.7	1.7	0.3
	0.7	3.3	0.9	1.9	3.3	1.0	0.5	3.1	1.4	0.5	2.7	0.5	0.5	2.7	0.5	0.5
	0.9	1.4	1.6	0.2	1.0	1.6	0.0	1.0	1.3	0.0	1.1	1.5	0.0	1.1	1.5	0.0
0.95	0.1	3.3	2.7	0.0	3.3	2.7	0.0	3.3	2.7	0.0	3.3	2.8	0.0	3.3	2.8	0.0
	0.3	2.5	3.0	0.0	2.6	3.0	0.0	2.6	3.0	0.0	2.9	3.0	0.0	2.9	2.9	0.0
	0.5	3.4	2.7	0.0	3.4	2.7	0.0	3.4	2.7	0.0	3.4	2.7	0.0	3.4	2.7	0.0
	0.7	3.3	2.5	0.0	3.3	2.5	0.0	3.3	2.5	0.0	3.3	2.5	0.0	3.3	2.5	0.0
	0.9	2.8	2.7	0.0	2.8	2.7	0.0	2.8	2.7	0.0	2.8	2.7	0.0	2.8	2.7	0.0

10 Jobs

Lambda	q	Change Threshold 0.1			0.3			0.5			0.7			0.9		
		Q10	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*	*Q10*	*Q50*	*Q90*
0.3	0.1	0.6	1.3	1.0	0.0	0.0	0.8	1.6	1.7	0.6	2.4	2.2	0.6	2.4	2.2	0.6
	0.3	0.3	0.7	2.0	0.7	0.6	0.5	2.0	2.1	0.8	2.0	2.1	0.8	2.0	2.1	0.8
	0.5	0.5	0.8	1.1	1.0	0.5	0.1	1.8	1.8	0.4	1.8	1.8	0.4	1.8	1.8	0.4
	0.7	0.5	0.5	0.0	1.6	1.7	0.0	2.1	1.8	0.0	2.1	1.8	0.0	2.1	1.8	0.0
	0.9	1.7	1.3	0.0	2.2	1.8	0.0	2.2	1.8	0.0	2.2	1.8	0.0	2.2	1.8	0.0
0.5	0.1	2.8	2.5	0.0	2.8	2.5	0.0	2.8	2.5	0.0	2.8	2.5	0.0	2.8	2.5	0.0
	0.3	2.4	2.3	0.0	2.4	2.3	0.0	2.4	2.3	0.0	2.4	2.3	0.0	2.4	2.3	0.0
	0.5	4.2	3.6	0.0	4.2	3.6	0.0	4.2	3.6	0.0	4.2	3.6	0.0	4.2	3.6	0.0
	0.7	4.0	3.7	0.0	4.0	3.7	0.0	4.0	3.7	0.0	4.0	3.7	0.0	4.1	3.7	0.0
	0.9	4.6	3.9	0.0	4.6	3.9	0.0	4.6	3.9	0.0	4.6	3.9	0.0	4.6	3.9	0.0
0.9	0.1	3.8	6.1	0.5	3.8	6.2	0.5	3.8	6.1	0.5	3.9	6.3	0.5	3.6	5.8	2.1
	0.3	5.4	4.6	0.4	5.7	5.3	0.4	6.1	4.9	0.1	5.3	6.0	0.3	5.0	5.2	0.3
	0.5	4.0	3.7	0.0	4.0	4.0	0.0	4.4	4.4	0.0	4.1	3.4	0.0	4.1	3.6	0.0
	0.7	5.9	5.0	0.3	6.0	4.9	0.3	5.4	4.1	0.3	5.1	4.1	0.5	5.2	4.1	0.5
	0.9	6.7	5.1	0.0	6.5	4.4	0.0	5.9	4.4	0.0	5.9	4.4	0.0	6.0	4.4	0.0
0.95	0.1	7.6	6.9	0.0	7.6	6.9	0.0	7.6	7.3	0.0	7.6	7.2	0.0	7.6	7.3	0.0
	0.3	8.3	6.8	0.0	8.3	6.8	0.0	8.3	6.8	0.0	8.2	6.9	0.0	8.7	6.9	0.0
	0.5	8.8	6.5	0.0	8.8	6.5	0.0	8.8	6.5	0.0	8.8	6.5	0.0	8.8	6.7	0.0
	0.7	8.6	6.5	0.0	8.6	6.5	0.0	8.6	6.5	0.0	8.6	6.5	0.0	8.6	6.5	0.0
	0.9	7.9	7.3	0.0	7.9	7.3	0.0	7.9	7.3	0.0	7.9	7.3	0.0	8.2	7.3	0.0

Table 11: Difference between the 10th, 50th and 90th quantiles *predictive*-only's *cdf* and the *predictive-reactive cdf*, for the cases with 5 and 10 jobs, and 9 uniformly distributed processing times.

5 Jobs

Lambda	q	Change Threshold 0.1 Q10	0.1 Q50	0.1 Q90	0.3 Q10	0.3 Q50	0.3 Q90	0.5 Q10	0.5 Q50	0.5 Q90	0.7 Q10	0.7 Q50	0.7 Q90	0.9 Q10	0.9 Q50	0.9 Q90
0.3	0.1	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0
	0.3	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0
	0.5	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0
	0.7	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0	0.2	0.2	0.0
	0.9	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.2	0.0	0.1	0.2	0.0
0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.9	0.1	0.7	0.4	3.7	0.6	0.3	1.3	0.6	0.4	0.3	0.6	0.6	2.4	0.6	0.6	2.4
	0.3	0.8	0.6	0.0	0.8	0.6	0.0	0.8	0.6	0.0	0.8	0.6	0.0	0.8	0.6	0.0
	0.5	0.6	0.3	1.2	0.6	0.3	3.2	0.7	0.4	0.8	0.8	0.5	1.9	0.8	0.5	1.9
	0.7	0.2	0.3	0.0	0.3	0.4	0.3	0.4	0.6	0.0	0.4	0.6	0.1	0.4	0.6	0.1
	0.9	0.5	0.5	2.6	0.5	0.6	0.7	0.6	0.7	1.1	0.6	0.8	1.1	0.6	0.8	1.1
0.95	0.1	0.3	1.4	1.1	0.3	1.3	1.1	0.3	1.2	0.5	0.2	0.6	1.4	0.0	0.2	1.4
	0.3	0.6	1.2	0.8	0.5	1.2	0.8	0.4	0.9	0.8	0.1	0.2	0.9	0.1	0.1	1.4
	0.5	0.4	1.2	1.7	0.4	0.9	0.8	0.4	0.8	0.8	0.1	0.1	0.3	0.1	0.1	3.0
	0.7	0.6	1.5	1.9	0.5	1.1	1.9	0.2	0.4	0.6	0.0	0.2	1.6	0.0	0.0	1.6
	0.9	0.4	1.2	1.4	0.3	0.7	1.2	0.1	0.2	0.9	0.0	0.1	0.9	0.0	0.1	0.9

10 Jobs

Lambda	q	Change Threshold 0.1 Q10	0.1 Q50	0.1 Q90	0.3 Q10	0.3 Q50	0.3 Q90	0.5 Q10	0.5 Q50	0.5 Q90	0.7 Q10	0.7 Q50	0.7 Q90	0.9 Q10	0.9 Q50	0.9 Q90
0.3	0.1	0.5	0.4	0.0	0.5	0.4	0.0	0.5	0.4	0.0	0.5	0.4	0.0	0.5	0.4	0.0
	0.3	0.8	0.5	0.0	0.8	0.5	0.0	0.8	0.5	0.0	0.8	0.5	0.0	0.8	0.5	0.0
	0.5	0.6	0.5	0.0	0.6	0.5	0.0	0.6	0.5	0.0	0.6	0.5	0.0	0.6	0.5	0.0
	0.7	0.5	0.5	0.0	0.5	0.5	0.0	0.5	0.5	0.0	0.5	0.5	0.0	0.5	0.5	0.0
	0.9	0.5	0.4	0.0	0.5	0.4	0.0	0.5	0.4	0.0	0.5	0.4	0.0	0.5	0.4	0.0
0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.9	0.1	1.1	1.1	0.1	1.1	1.1	0.1	1.1	1.3	0.3	1.2	1.5	0.1	1.2	1.5	0.1
	0.3	2.1	1.7	0.0	2.1	1.7	0.0	2.1	1.7	0.0	2.1	1.7	0.0	2.1	1.7	0.0
	0.5	1.6	1.2	3.2	1.7	1.2	0.6	1.8	1.2	0.9	1.8	1.5	0.9	1.8	1.5	0.9
	0.7	2.1	1.1	1.5	2.1	1.1	7.7	2.2	1.4	2.0	2.2	1.4	2.0	2.2	1.4	2.0
	0.9	1.6	1.3	1.0	1.6	1.4	1.0	1.7	1.6	0.7	1.7	1.6	0.7	1.7	1.7	0.7
0.95	0.1	1.0	2.4	3.1	0.9	1.9	3.1	0.8	1.6	1.1	0.3	0.4	2.1	0.1	0.1	5.1
	0.3	0.8	2.2	2.5	0.6	1.7	2.5	0.5	1.2	2.5	0.1	0.2	3.7	0.1	0.1	3.7
	0.5	0.6	2.1	2.2	0.6	1.8	2.2	0.4	1.1	2.2	0.2	0.3	2.8	0.1	0.1	3.7
	0.7	0.6	1.8	4.7	0.4	1.4	3.6	0.2	0.4	2.8	0.1	0.1	2.8	0.1	0.1	2.8
	0.9	0.8	1.4	6.2	0.6	0.9	1.0	0.2	0.3	2.4	0.1	0.2	2.4	0.1	0.2	2.4

Table 12: Difference between the 10th, 50th and 90th quantiles *predictive*-only's *cdf* and the *predictive-reactive cdf*, for the cases with 5 and 10 jobs, and 9 triangularly distributed processing times.

5 Jobs

Lambda	q	PR (Change Threshold) 0.1	PR 0.3	PR 0.5	PR 0.7	PR 0.9	P-only	FCFS/FRO
0.3	0.1	3.514	3.508	3.529	3.478	3.589	4.097	4.176
	0.3	2.971	2.991	2.997	2.926	3.066	3.988	4.246
	0.5	3.628	3.654	3.664	3.670	3.775	4.422	5.100
	0.7	2.779	2.796	2.748	2.773	2.916	3.637	4.875
	0.9	2.725	2.713	2.736	2.763	2.922	3.597	4.786
0.5	0.1	4.406	4.407	4.407	4.291	4.740	5.849	6.435
	0.3	4.326	4.323	4.232	4.157	4.566	5.661	6.745
	0.5	4.255	4.140	4.142	4.144	4.823	5.919	6.982
	0.7	3.368	3.377	3.402	3.491	4.596	5.533	6.103
	0.9	3.977	3.974	3.975	3.095	4.909	5.660	6.144
0.9	0.1	11.829	11.830	11.840	11.798	12.145	16.098	17.011
	0.3	12.635	12.660	12.763	13.059	13.426	16.185	17.123
	0.5	12.147	12.204	12.186	12.795	14.551	16.313	17.342
	0.7	12.033	12.176	12.474	12.607	14.901	15.765	16.983
	0.9	11.982	12.494	12.659	12.810	14.544	15.641	16.878
0.95	0.1	15.038	15.082	15.130	15.211	15.699	22.200	23.323
	0.3	13.902	14.088	14.457	14.612	16.686	20.852	22.231
	0.5	15.393	15.661	15.944	16.356	17.388	22.943	24.092
	0.7	14.637	14.795	14.917	15.028	16.438	21.597	22.874
	0.9	15.519	15.702	15.935	16.241	17.910	23.120	25.001

10 Jobs

Lambda	q	PR (Change Threshold) 0.1	PR 0.3	PR 0.5	PR 0.7	PR 0.9	P-only	FCFS/FRO
0.3	0.1	4.086	4.154	4.338	4.908	7.319	7.752	8.892
	0.3	4.117	4.390	4.667	5.872	7.547	8.101	9.234
	0.5	4.480	4.757	4.949	6.038	7.507	7.908	9.012
	0.7	1.742	1.829	2.462	2.758	2.767	3.224	4.124
	0.9	2.096	2.486	2.888	2.958	2.958	3.341	5.213
0.5	0.1	3.916	3.916	3.916	4.179	6.228	6.401	7.564
	0.3	3.418	3.450	3.593	4.459	5.322	5.808	7.019
	0.5	5.323	5.602	6.651	8.690	8.690	8.690	10.023
	0.7	6.544	7.217	8.971	9.845	9.845	9.845	11.023
	0.9	6.905	7.828	8.888	8.958	8.958	8.958	10.234
0.9	0.1	15.400	15.414	15.507	15.574	15.990	25.122	27.234
	0.3	17.684	17.797	17.768	17.821	20.552	29.194	29.987
	0.5	16.514	16.515	16.559	16.672	18.297	25.375	25.987
	0.7	14.190	14.483	15.409	16.468	17.343	19.787	20.342
	0.9	15.064	15.621	16.497	16.791	17.726	21.203	21.987
0.95	0.1	19.253	19.253	19.253	19.253	19.253	25.361	26.128
	0.3	20.038	20.038	20.038	20.038	20.038	23.373	24.099
	0.5	19.599	19.599	19.599	19.599	19.599	23.331	23.878
	0.7	19.657	19.666	19.765	19.879	19.802	22.756	22.998
	0.9	20.002	20.050	20.510	20.158	20.190	22.776	23.347

Table 13: *AQD* between the *cdf* of the optimal solution for the *predictive*-only (*P-only*) and the *predictive-reactive* (*PR*) approaches, using 5 and 10 jobs, and 5 uniformly distributed processing times.

5 Jobs

Lambda	q	PR Change Threshold 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	102.595	102.604	102.604	102.623	102.680	102.680	102.998
	0.3	103.706	103.715	103.715	103.731	103.731	103.731	104.274
	0.5	103.998	104.015	104.015	104.043	104.043	104.043	104.877
	0.7	102.757	102.763	102.770	102.802	102.802	102.802	103,556
	0.9	103.295	103.308	103.339	103.339	103.339	103.339	104.033
0.5	0.1	62.431	62.514	62.823	63.424	64.785	64.958	65.566
	0.3	61.452	61.509	61.752	63.117	63.918	64.003	64.998
	0.5	62.519	62.679	63.213	63.878	64.363	64.425	64.565
	0.7	61.229	61.638	62.313	62.747	63.084	63.189	64.927
	0.9	64.591	65.017	65.549	66.085	66.419	66.565	67.223
0.9	0.1	92.140	92.196	92.304	92.022	95.739	100.216	100.877
	0.3	92.563	92.730	92.966	93.642	97.166	98.294	99.231
	0.5	91.869	92.333	92.813	94.100	96.681	97.237	98.332
	0.7	93.506	93.909	95.114	96.613	97.585	97.662	98.915
	0.9	90.206	90.903	91.965	93.204	94.347	94.784	98.147
0.95	0.1	102.035	102.257	102.473	103.373	107.323	106.897	108.832
	0.3	103.018	103.601	104.122	106.274	105.997	106.564	107.733
	0.5	103.587	104.848	106.084	106.506	106.049	106.773	107.288
	0.7	101.653	102.734	102.862	102.804	102.760	103.223	104.989
	0.9	105.659	106.466	106.395	106.075	105.928	106.591	107.453

10 Jobs

Lambda	q	PR Change Threshold 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	341.720	341.720	341.720	341.720	341.720	342.053	343.475
	0.3	389.425	389.425	389.425	389.425	389.425	389.560	390.873
	0.5	391.455	391.455	391.455	391.455	391.455	391.945	392.867
	0.7	390.290	390.290	390.290	390.290	390.290	390.880	392.033
	0.9	386.930	386.930	386.930	386.930	386.930	387.285	389.825
0.5	0.1	290.654	290.888	291.070	291.208	291.254	291.280	293.284
	0.3	294.532	294.620	294.606	294.610	294.660	294.678	296.915
	0.5	294.196	294.292	294.380	294.424	294.536	294.536	296.782
	0.7	290.002	290.048	290.042	290.062	290.076	290.076	292.872
	0.9	292.630	292.638	292.628	292.686	292.712	292.712	294.881
0.9	0.1	230.992	230.933	230.622	231.167	233.376	230.376	232.920
	0.3	234.635	234.940	236.307	237.086	237.370	236.228	238.838
	0.5	224.252	225.102	225.782	226.065	226.918	228.184	230.273
	0.7	247.644	248.820	249.150	249.778	251.520	251.468	255.936
	0.9	234.524	235.053	235.185	235.093	237.185	236.945	238.846
0.95	0.1	382.578	382.695	382.398	382.078	382.978	386.045	388.745
	0.3	382.367	384.493	387.677	388.823	389.127	388.673	391.756
	0.5	394.330	397.333	397.473	397.517	398.483	397.773	398.568
	0.7	400.697	400.663	400.457	400.780	401.837	401.407	405.846
	0.9	401.377	401.207	401.200	401.013	401.370	401.123	403.475

Table 14: *AQD* between the *cdf* of the optimal solution for the *predictive*-only (*P-only*) and the *predictive-reactive* (*PR*) approaches, using 5 and 10 jobs, and 7 uniformly distributed processing times.

5 Jobs

Lambda	q	PR *Change Threshold* 0.1	PR 0.3	PR 0.5	PR 0.7	PR 0.9	P-only	FCFS/FRO
0.3	0.1	418.766	412.753	414.263	414.276	414.291	421.088	423.843
	0.3	462.315	466.516	467.199	467.223	467.223	479.304	481.043
	0.5	462.490	467.029	467.778	467.778	467.778	479.513	482.943
	0.7	462.844	466.159	466.159	466.159	466.159	478.088	480.713
	0.9	464.750	466.304	466.338	466.354	466.354	477.900	481.923
0.5	0.1	114.092	113.888	114.431	116.644	117.249	116.932	118.938
	0.3	115.574	116.081	116.874	118.028	117.749	118.351	120.634
	0.5	113.471	113.981	114.555	114.634	114.290	115.564	126.934
	0.7	113.801	114.667	115.060	114.920	114.354	114.436	115.937
	0.9	113.256	113.418	113.226	113.090	113.092	114.509	115.934
0.9	0.1	497.103	497.275	497.468	497.330	497.278	575.810	577.821
	0.3	498.390	498.460	498.513	498.665	499.523	575.403	576.511
	0.5	502.525	502.438	498.207	497.880	497.800	586.956	588.623
	0.7	405.940	406.390	406.390	406.390	406.390	406.390	406.425
	0.9	415.770	415.910	415.910	415.910	415.910	415.910	416.845
0.95	0.1	210.134	210.324	210.233	211.702	217.506	217.449	219.733
	0.3	213.474	213.943	215.386	216.933	224.511	213.444	215.994
	0.5	217.824	218.835	220.444	222.358	227.065	214.020	215.333
	0.7	216.199	217.316	219.400	222.946	226.552	218.969	220.138
	0.9	212.305	213.881	215.948	219.408	221.621	213.808	215.233

10 Jobs

Lambda	q	PR *Change Threshold* 0.1	PR 0.3	PR 0.5	PR 0.7	PR 0.9	P-only	FCFS/FRO
0.3	0.1	415.430	415.443	415.463	415.463	415.463	415.463	427.143
	0.3	414.167	414.177	414.177	414.177	414.177	414.177	415.362
	0.5	414.523	414.523	414.553	414.553	414.553	414.553	416.833
	0.7	414.490	414.503	414.503	414.503	414.503	414.503	416.226
	0.9	413.523	413.543	413.543	413.543	413.543	413.543	416.826
0.5	0.1	3.916	3.916	3.916	4.179	6.228	6.401	8.125
	0.3	3.418	3.450	3.593	4.459	5.322	5.808	7.145
	0.5	5.323	5.602	6.651	8.690	8.690	8.690	10.012
	0.7	6.544	7.217	8.971	9.845	9.845	9.845	10.265
	0.9	6.905	7.828	8.888	8.958	8.958	8.958	10.129
0.9	0.1	500.213	500.269	500.501	502.263	506.986	531.273	534.712
	0.3	509.957	511.166	512.610	514.509	520.461	562.717	564.912
	0.5	452.314	452.626	452.982	454.294	458.132	506.774	508.622
	0.7	456.850	457.440	457.930	459.266	464.704	506.970	509.893
	0.9	454.732	455.556	456.122	456.888	461.934	502.894	504.837
0.95	0.1	408.693	408.823	409.320	410.210	317.370	475.747	477.823
	0.3	314.220	314.750	316.047	318.920	320.480	320.590	323.918
	0.5	251.480	252.785	255.025	255.975	255.975	255.975	256.810
	0.7	248.020	248.425	248.985	249.220	249.340	249.340	258.934
	0.9	179.460	180.650	180.650	180.650	180.650	180.650	182.304

Table 15: *AQD* between the *cdf* of the optimal solution for the *predictive*-only (*P-only*) and the *predictive-reactive* (*PR*) approaches, using 5 and 10 jobs, and 9 uniformly distributed processing times.

5 Jobs

Lambda	q	PR Change Threshold 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	17.505	18.929	23.763	25.668	27.303	31.766	33.292
	0.3	15.833	23.249	24.157	26.188	27.265	30.717	31.029
	0.5	4.244	4.990	5.077	7.173	8.790	7.490	8.349
	0.7	17.553	22.141	22.272	24.240	26.220	29.770	29.990
	0.9	10.321	15.462	16.769	18.708	19.953	23.215	23.845
0.5	0.1	1.595	1.645	1.685	2.211	2.476	3.426	5.934
	0.3	1.276	1.276	1.376	1.909	2.471	3.132	3.234
	0.5	1.677	1.677	1.829	2.284	2.738	3.461	3.762
	0.7	1.486	1.486	1.866	2.215	2.537	3.109	3.928
	0.9	1.311	1.458	1.731	2.267	2.523	3.022	4.056
0.9	0.1	5.563	5.441	4.792	4.923	5.059	5.715	7.029
	0.3	5.469	5.034	4.451	4.649	4.735	6.110	9.934
	0.5	5.152	5.096	4.844	5.085	5.141	6.259	9.238
	0.7	5.280	4.997	4.938	4.047	5.075	6.590	7.974
	0.9	5.352	5.253	5.255	5.315	5.355	6.695	8.986
0.95	0.1	6.910	6.910	7.125	7.223	7.357	7.638	9.235
	0.3	7.352	7.417	7.524	7.590	7.730	8.261	9.277
	0.5	7.174	7.227	7.356	7.398	7.429	7.918	8.723
	0.7	6.609	6.712	6.856	6.858	6.980	7.204	8.566
	0.9	7.550	7.803	7.820	7.962	7.963	8.155	8.349

10 Jobs

Lambda	q	PR Change Threshold 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	1.906	1.906	1.906	1.906	1.906	3.112	4.926
	0.3	2.413	2.413	2.413	2.413	2.413	4.388	4.932
	0.5	2.478	2.478	2.478	2.478	2.478	4.178	6.837
	0.7	2.184	2.184	2.184	2.184	2.184	3.264	4.924
	0.9	2.578	2.578	2.578	2.578	2.578	4.305	6.384
0.5	0.1	0.777	0.777	0.777	0.777	0.777	0.777	1.382
	0.3	0.698	0.698	0.698	0.698	0.698	0.698	1.843
	0.5	0.901	0.901	0.901	0.901	0.901	0.901	2.094
	0.7	1.396	1.396	1.396	1.396	1.396	1.396	2.973
	0.9	0.823	0.823	0.823	0.823	0.823	0.823	2.945
0.9	0.1	3.359	3.359	3.359	3.443	3.398	5.204	6.934
	0.3	3.517	3.517	3.517	3.517	3.517	5.834	5.983
	0.5	3.547	3.547	3.547	3.547	3.380	4.828	5.823
	0.7	3.779	4.064	4.014	4.046	4.211	6.224	7.834
	0.9	3.293	3.340	3.345	3.662	4.272	6.853	8.335
0.95	0.1	4.998	4.902	5.002	5.122	5.132	5.768	6.834
	0.3	4.920	4.999	5.006	5.130	5.152	5.563	9.354
	0.5	5.003	5.020	5.098	5.120	5.120	5.239	8.823
	0.7	5.027	5.019	4.973	5.197	5.824	5.689	6.929
	0.9	5.117	5.078	5.861	5.847	5.883	5.330	6.834

Table 16: *AQD* between the *cdf* of the optimal solution for the *predictive*-only (*P-only*) and the *predictive-reactive* (*PR*) approaches, using 5 and 10 jobs, and 5 triangularly distributed processing times.

5 Jobs

Lambda	q	PR (Change Threshold) 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	75.843	76.345	76.417	76.417	76.422	78.312	80.763
	0.3	77.464	77.990	77.990	77.990	78.005	77.962	78.526
	0.5	77.340	77.693	77.695	77.695	77.742	77.671	79.536
	0.7	77.399	77.493	77.493	77.493	77.517	77.446	79.523
	0.9	78.227	78.235	78.239	78.239	78.260	78.123	80.745
0.5	0.1	83.807	83.837	83.0730	79.738	78.381	85.248	86.934
	0.3	83.479	83.364	81.851	96.072	90.539	89.110	91.845
	0.5	83.063	82.911	81.235	78.913	78.705	82.385	83.748
	0.7	84.065	83.758	81.335	79.447	78.801	78.660	80.356
	0.9	83.326	80.927	79.082	79.023	78.722	78.722	80.253
0.9	0.1	47.285	47.134	47.381	47.925	49.330	48.343	48.845
	0.3	46.069	46.102	46.296	46.884	48.159	47.582	47.927
	0.5	46.083	46.664	46.541	47.408	48.685	46.972	47.763
	0.7	46.206	46.328	46.902	47.964	49.134	47.253	48.734
	0.9	47.446	47.481	47.798	48.640	49.698	48.135	49.346
0.95	0.1	100.999	100.988	101.186	101.805	103.091	102.045	103.467
	0.3	100.647	100.603	100.828	101.501	102.760	101.677	103.264
	0.5	99.568	99.695	100.173	100.421	101.630	101.1163	102.341
	0.7	91.831	92.136	92.690	93.407	93.845	94.218	97.845
	0.9	89.184	89.488	90.269	91.259	91.432	91.704	92.846

10 Jobs

Lambda	q	PR (Change Threshold) 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	221.418	221.316	221.322	221.602	221.064	221.440	223.945
	0.3	218.908	218.908	218.908	218.908	218.908	218.810	220.125
	0.5	219.510	219.540	219.540	219.540	219.540	219.270	220.844
	0.7	218.940	218.940	218.940	218.940	218.940	218.813	220.312
	0.9	252.540	252.540	252.540	252.540	252.540	252.180	253.663
0.5	0.1	139.073	139.103	139.238	138.211	133.520	140.943	142.745
	0.3	137.666	137.537	137.394	135.111	130.776	139.558	140.427
	0.5	136.296	136.033	136.033	135.448	130.633	138.860	139.629
	0.7	137.660	138.223	137.656	136.229	132.500	139.234	140.723
	0.9	136.418	136.146	135.943	133.629	136.030	136.146	137.899
0.9	0.1	228.149	227.468	227.528	228.095	220.269	229.986	230.884
	0.3	222.929	222.926	223.491	224.327	226.551	227.521	228.894
	0.5	222.292	222.256	222.872	224.061	226.061	226.889	228.845
	0.7	145.594	146.150	147.502	148.729	150.055	148.136	150.839
	0.9	198.980	199.790	202.050	204.340	205.610	206.450	208.673
0.95	0.1	262.910	263.177	263.437	263.437	263.437	267.177	268.937
	0.3	261.800	261.940	262.007	262.007	262.007	265.683	267.783
	0.5	210.205	210.715	210.810	210.810	210.810	211.620	212.005
	0.7	210.635	210.900	210.900	210.900	210.900	210.535	210.989
	0.9	210.755	210.755	210.755	210.755	210.755	210.255	210.748

Table 17: *AQD* between the *cdf* of the optimal solution for the *predictive*-only (*P-only*) and the *predictive-reactive* (*PR*) approaches, using 5 and 10 jobs, and 7 triangularly distributed processing times.

5 Jobs

Lambda	q	PR Change Threshold 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	122.046	121.547	120.956	121.040	121.260	121.377	122.849
	0.3	121.829	121.115	121.081	121.085	121.342	121.414	122.684
	0.5	121.313	120.355	120.186	120.280	120.504	120.689	121.839
	0.7	120.743	120.269	120.345	120.345	120.345	120.821	121.673
	0.9	123.084	123.148	122.938	120.534	121.980	120.750	121.745
0.5	0.1	1.595	1.645	1.685	2.211	2.476	3.426	4.923
	0.3	1.276	1.276	1.376	1.909	2.471	3.132	4.910
	0.5	1.677	1.677	1.829	2.284	2.738	3.461	4.923
	0.7	1.486	1.486	1.866	2.215	2.537	3.109	5.023
	0.9	1.311	1.458	1.731	2.267	2.523	3.022	5.834
0.9	0.1	5.563	5.441	4.792	4.923	5.059	5.715	7.937
	0.3	5.469	5.034	4.451	4.649	4.735	6.110	7.836
	0.5	5.152	5.096	4.844	5.085	5.141	6.259	6.789
	0.7	5.280	4.997	4.938	4.047	5.075	6.590	6.988
	0.9	5.352	5.253	5.255	5.315	5.355	6.695	7.748
0.95	0.1	6.910	6.910	7.125	7.223	7.357	7.638	8.984
	0.3	7.352	7.417	7.524	7.590	7.730	8.261	8.872
	0.5	7.174	7.227	7.356	7.398	7.429	7.918	8.783
	0.7	6.609	6.712	6.856	6.858	6.980	7.204	7.838
	0.9	7.550	7.803	7.820	7.962	7.963	8.155	8.358

10 Jobs

Lambda	q	PR Change Threshold 0.1	0.3	0.5	0.7	0.9	P-only	FCFS/FRO
0.3	0.1	418.766	422.753	424.263	424.276	424.291	421.088	423.928
	0.3	462.315	466.516	467.199	467.223	467.223	479.304	480.847
	0.5	462.490	467.029	467.778	467.778	467.778	479.513	480.145
	0.7	462.844	466.159	466.159	466.159	466.159	478.088	479.233
	0.9	464.750	466.304	466.338	466.354	466.354	477.900	478.859
0.5	0.1	148.207	148.447	148.538	148.544	148.544	148.429	149.758
	0.3	148.197	148.412	148.488	148.554	148.653	148.549	149.875
	0.5	148.208	148.400	148.466	148.496	148.524	148.411	149.039
	0.7	148.709	148.840	148.889	149.047	149.026	148.867	151.849
	0.9	148.459	148.592	148.693	148.725	148.738	148.648	152.957
0.9	0.1	162.246	162.280	162.244	161.879	162.104	162.063	163.957
	0.3	173.228	172.937	172.809	170.601	172.839	172.896	175.937
	0.5	173.142	173.107	173.008	172.866	172.986	172.932	175.836
	0.7	173.787	173.751	173.721	173.958	174.168	174.175	175.937
	0.9	173.304	173.290	173.327	173.675	177.975	174.547	175.836
0.95	0.1	183.193	182.955	182.600	182.393	182.767	219.791	220.473
	0.3	181.431	181.450	181.256	181.578	182.420	219.759	220.934
	0.5	185.169	184.866	184.796	184.629	185.077	222.083	222.449
	0.7	182.410	182.198	182.166	182.253	182.420	219.657	222.754
	0.9	182.623	182.512	182.365	182.671	182.805	220.138	221.947

Table 18: *AQD* between the *cdf* of the optimal solution for the *predictive*-only (*P-only*) and the *predictive-reactive* (*PR*) approaches, using 5 and 10 jobs, and 9 triangularly distributed processing times.