J Sched (2013) 16:3–28
DOI 10.1007/s10951-012-0303-z

A survey on offline scheduling with rejection

Dvir Shabtay · Nufar Gaspar · Moshe Kaspi

Published online: 20 December 2012

Abstract In classical deterministic scheduling problems, it is assumed that all jobs have to be processed. However, in many practical cases, mostly in highly loaded make-to-order production systems, accepting all jobs may cause a delay in the completion of orders which in turn may lead to high inventory and tardiness costs. Thus, in such systems, the firm may wish to reject the processing of some jobs by either outsourcing them or rejecting them altogether. The field of scheduling with rejection provides schemes for coordinated sales and production decisions by grouping them into a single model. Since scheduling problems with rejection are very interesting both from a practical and a theoretical point of view, they have received a great deal of attention from researchers over the last decade. The purpose of this survey is to offer a unified framework for offline scheduling with rejection by presenting an up-to-date survey of the results in this field. Moreover, we highlight the close connection between scheduling with rejection and other fields of research such as scheduling with controllable processing times and scheduling with due date assignment, and include some new results which we obtained for open problems.

Keywords Deterministic scheduling · Job rejection · Order acceptance · Sourcing decisions · Optimization and complexity

D. Shabtay (✉) · N. Gaspar · M. Kaspi
Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer Sheva, Israel
e-mail: dvirs@bgu.ac.il

N. Gaspar
e-mail: nufarg@bgu.ac.il

M. Kaspi
e-mail: moshe@bgu.ac.il

1 Introduction

Scheduling problems have been extensively studied in the literature under the assumption that all jobs have to be processed. However, in many practical cases, the manufacturer does not have to process all the jobs and thus a higher level decision of partitioning the set of jobs into a set of *accepted* and a set of *rejected* jobs has to be made prior to the scheduling decision. Then, the set of accepted jobs has to be efficiently scheduled on the machines so as to minimize a given predefined scheduling criterion. According to Cesaret et al. (2012), an important application of scheduling with rejection arises in make-to-order production systems with limited production capacity and tight delivery requirements, where simultaneous job rejection and scheduling decisions have to be made for maximizing the total revenue. In such systems, accepting orders without considering their impact on production capacity may cause overload in some periods, which in turn may delay some of the orders beyond their due dates. To be able to use production capacity more efficiently and preserve high quality of service (QoS) to customers of accepted jobs, the manufacturer has to determine which orders to accept and how to schedule them to maximize total revenue. Guerrero and Kern (1988) claim that accepting orders without considering their possible costly impact on capacity can mean that the firm is actually paying for the privilege of accepting an order. To make the right decision of whether to accept or reject a job, the firm has to take into account the trade-off between the revenue brought in by the order and the amount of resources that should be diverted to its processing.

Another important application of scheduling with rejection occurs in scheduling with an outsourcing option. Outsourcing has become a megatrend in many industries, most particularly in logistics and supply chain management.

Managers are increasingly under pressure to make the right sourcing decision, as business consequences can be significant, resulting in, for example increased costs, disrupted service, and even business failure (McGovern and Quelch 2005). According to the *Outsourcing Institute*, outsourcing can reduce and control operating costs, free internal resources for other purposes, gain access to world-class capabilities and encourage the sharing of risks (http://www.outsourcing.com).

Firms tend to consider sales/sourcing decisions to be at a higher hierarchal decision level than the actual scheduling decisions themselves. Although higher and lower level decision-making processes are tightly connected, traditional scheduling problems have been extensively studied under the assumption that any higher level decision has already been made and that the set of jobs to be processed in the shop is predefined to the scheduler. The field of scheduling with rejection provides tools for efficiently coordinating sales/sourcing and production scheduling decisions by combining them in a single model.

The aim of this study is to offer a unified framework for *offline* scheduling with rejection by providing an up-to-date survey of the results in this field. In addition, we highlight the close connection between scheduling with rejection and other fields of research such as scheduling with controllable processing times and scheduling with due date assignment, and present new results that we have obtained for some open problems.

A general definition of an offline scheduling problem with rejection may be stated as follows: n-independent, non-preemptive jobs, $J = \{J_1, J_2, \ldots, J_n\}$, are available for processing at time zero on a set of m machines $M = \{M_1, M_2, \ldots, M_m\}$ arranged according to a specific and predefined machine environment (such as identical, uniform, or unrelated machines in parallel, flow-shop, job-shop, or open-shop environments), where O_{ij} is the *operation* of job J_j on machine M_i. The input for a problem of scheduling with rejection includes the following two parameters: p_{ij}, which is the *processing time* of job J_j on machine M_i for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ (in a single machine environment the subscript i is omitted so that p_j is the processing time of job J_j), and e_j which is the *rejection cost* of job J_j. Additional parameters, depending on the specific objective function, include d_j, the *due date* of job J_j; r_j, the *release date* or the time in which job J_j arrives in the system; and w_j, a *weight* or significance attributed to job J_j (note that all parameters are assumed to be positive integers).

A solution S of a scheduling problem with rejection is defined by a partition of set J into two subsets—set A, which refers to the set of accepted jobs, and set $\overline{A}$, which refers to the set of rejected jobs—as well as by a schedule of set A on the m machines. The quality of a solution is measured by two criteria: F_1, a *scheduling criterion* that depends on the jobs' completion times, and,

$$F_2 = \mathrm{RC} = \sum_{J_j \in \overline{A}} e_j, \tag{1}$$

which is the *total rejection cost*. Some possible F_1 criteria include $\sum_{J_j \in A} C_j$, $\sum_{J_j \in A} W_j$, $\sum_{J_j \in A} E_j$, $\sum_{J_j \in A} T_j$, $\sum_{J_j \in A} U_j$ and $f_{\max}(A)$, where C_j is the *completion time* of job J_j; $W_j = C_j - p_j$ is the *waiting time* of job J_j; $L_j = C_j - d_j$ is the *lateness* of job J_j; $T_j = \max\{0, L_j\}$ is the *tardiness* of job J_j; $E_j = \max\{0, -L_j\}$ is the *earliness* of job J_j; U_j is the tardiness indicator variable for job J_j, i.e., $U_j = 1$ if $C_j > d_j$ and $U_j = 0$ if $C_j \leq d_j$, and $f_{\max}(A) = \max_{J_j \in A}\{f_j(C_j)\}$ with a non-decreasing function f_j for $j = 1, \ldots, n$.

Throughout this survey we use the standard three-field notation $X|Y|Z$ introduced by Graham et al. (1979) to describe scheduling problems with rejection. The X field describes the machine environment, the Y field defines the job processing characteristics and constraints (when referring to a scheduling problem with rejection, a rej entry in the Y field is included) and the Z field contains the optimization criteria.

Since scheduling with rejection is essentially a problem with two criteria, concepts from the theory of bicriteria scheduling are commonly used when dealing with such problems. Moreover, since most problems in this field are $\mathcal{NP}$-hard the use of approximation algorithms is common as well. Thus, in Sect. 1.1 we review the main concepts used in bicriteria scheduling [we refer the reader to T'kindt and Billaut (2006) and to Hoogeveen (2005) for a more detailed discussion], and in Sect. 1.2 we present the definitions of those approximation algorithms that are commonly used to solve $\mathcal{NP}$-hard scheduling problems.

1.1 Basic concepts of bicriteria scheduling

Let us consider the case where one criterion, say F_1, is far more important than the other criterion. In such a case one may consider using the *lexicographical approach* for optimization. According to this approach, among all solutions which minimize F_1, the one that minimizes F_2 is chosen. For scheduling with rejection, if F_1 is indeed the dominant criterion then using the lexicographical approach will yield a non-balanced solution for which $A = \emptyset$ and $\overline{A} = J$. In contrast, if F_2 is the dominant criterion then this approach will yield the opposite non-balanced solution in which all jobs are accepted. Thus, in most cases this approach is not useful for solving scheduling problems with rejection. However, no matter what method is used for selecting the most appropriate solution, only*Pareto-optimal* solutions are of interest. The definition of a Pareto-optimal solution is as follows (see, e.g., Hoogeveen 2005):

Definition 1 A solution S is called Pareto-optimal (or non-dominated) with respect to criteria F_1 and F_2 if there does not exist another solution S' such that $F_1(S') \leq F_1(S)$ and $F_2(S') \leq F_2(S)$, with at least one of these inequalities being strict.

One commonly used approach to select the most appropriate Pareto-optimal solution is the *a priori optimization* approach where the two criteria are aggregated into a given composite objective function $G(F_1, F_2)$ so that the objective is to find a solution S which minimizes $G(F_1, F_2)$. In scheduling with rejection, researchers mainly use the linear function for which $G(F_1, F_2) = G_l(F_1, F_2) = F_1 + F_2$. However, other types of non-decreasing functions can be used as well. There is a close relation between the optimal solutions which minimizes $G(F_1, F_2)$ and the set of Pareto-optimal solutions as given by the following theorem:

Theorem 1 *If $G(F_1, F_2)$ is a non-decreasing function of both F_1 and F_2 then there exists a Pareto-optimal solution which minimizes $G(F_1, F_2)$.*

Moreover, if $G(F_1, F_2)$ is a linear function, then the optimum is attained at an *extreme point* which we define below. The corresponding schedule is referred to as an *extreme schedule*.

Theorem 2 *An extreme point is a vertex of the efficient frontier, which is defined as the lower envelope of the convex hull of the set of Pareto-optimal points.*

We note that the efficient frontier is a piecewise-linear convex function, where each breakpoint is Pareto-optimal and each Pareto-optimal point is located either on or above this function.

In many real-life problems, there is a constraint on the value of one out of the two criteria. For example, the firm may want to provide high QoS to customers of accepted jobs by meeting their due dates. This may result in the need to solve the problem of minimizing RC subject to $F_1 = L_{\max}(A) \leq K = 0$. Another example arises when there is a capacity limitation and the machines are available only for K units of time. Then the firm may consider solving the problem of minimizing RC subject to $F_1 = C_{\max}(A) \leq K$. These types of problems may include more than a single optimal solution, some of which may be weak Pareto-optimal solutions.

Definition 2 A solution S is called a weak Pareto-optimal solution if it is not a Pareto-optimal solution and if there does not exist another schedule S' such that both $F_1(S') < F_1(S)$ and $F_2(S') < F_2(S)$.

Although additional problems can be defined (see, e.g., T'kindt and Billaut 2006, pp. 121–122), the literature on scheduling with rejection focuses on the following four problems.

- The first problem, which we denote by P1, uses the special case of the *a priori optimization* approach for which the aggregated function is linear and the objective is to find a solution which minimizes the total integrated cost, $G_l(F_1, F_2) = F_1 + F_2$. Using the scheduling notation introduced in T'kindt and Billaut (2006), this problem is referred to as $X|rej|F_1 + F_2$.
- The second problem, which we denote by P2, is to minimize F_1 subject to $F_2 \leq R$, where R is a given upper bound on the total rejection cost. Using the scheduling notation introduced in T'kindt and Billaut (2006), this problem can also be referred to as $X|rej|\epsilon(F_1/F_2)$. According to T'kindt and Billaut (2006), problem P2 is also called the ϵ-constraint problem with respect to the total rejection cost (F_2).
- The third problem, which we denote by P3, is to minimize F_2 subject to $F_1 \leq K$, where K is a given upper bound on the value of the scheduling criterion. Using the scheduling notation introduced in T'kindt and Billaut (2006), this problem can also be referred to as $X|rej|\epsilon(F_2/F_1)$. According to T'kindt and Billaut (2006), problem P3 is also called the ϵ-constraint problem with respect to the scheduling criterion (F_1).
- The fourth problem, which we denote by P4, is to identify a Pareto-optimal solution for each Pareto-optimal point. Using the scheduling notation introduced in T'kindt and Billaut (2006), this problem can also be referred to as $X|rej|\#(F_1/F_2)$.

It should be noted that solving problem P4 also solves problems P1–P3 as a by-product. Note also that the decision version (DV) of problem P2 is identical to that of problem P3, and is defined below.

Definition 3 DV*:* Given parameters K and R, is there a solution with $F_1 \leq K$ and $F_2 \leq R$?

The fact that both the P2 and the P3 problems share the same DV implies that either both or none of them is $\mathcal{NP}$-hard.

One method to solve the P4 problem is either to solve the corresponding P2 problem for any possible integer value of $R \in [0, \sum_{j=1}^{n} e_j]$ or to solve the corresponding P3 problem for any possible integer value of K (i.e., to solve a series of ϵ-constraint problems). Using this approach yields a set of Pareto and weak Pareto-optimal points, and by eliminating the weak Pareto points we end up with the entire set of Pareto-optimal points.

Most scheduling problems with rejection are $\mathcal{NP}$-hard. However, even $\mathcal{NP}$-hard problems are usually classified into two major subsets based on the existence of a pseudo-polynomial algorithm, which is defined below.

Definition 4 A pseudo-polynomial time algorithm is an algorithm for which running time is bounded by a polynomial of the size of the input, given that the input is written in

unary notation, i.e., if its running time is polynomial in the numeric value of the input.

An $\mathcal{NP}$-hard problem is considered $\mathcal{NP}$-hard in the ordinary sense (and not in the strong sense) if there exists a pseudo-polynomial time algorithm for its solution.

We note here that in most cases problems P2–P4 are $\mathcal{NP}$-hard. Moreover, the number of Pareto-optimal solutions is usually exponential in the number of jobs. For example, even for the simple case of a single machine with $F_1 = C_{max}$ and $p_j = e_j = j$ for $j = 1, \ldots, n$, any partition of set J into A and $\overline{A}$ yields a Pareto-optimal solution and thus there are 2^n Pareto-optimal solutions. However, one can easily group up a set of Pareto-optimal solutions which corresponds to the same specific Pareto-optimal point. For example, for the problem mentioned above the three different solutions with $\overline{A} = \{J_1, J_4\}$, $\overline{A} = \{J_2, J_3\}$, or $\overline{A} = \{J_5\}$ (and $A = J \setminus \overline{A}$ for each of these solutions) are all Pareto-optimal solutions. However, a single Pareto-optimal point of $F_1 = \frac{n(n+1)}{2} - 5$ and $F_2 = 5$ corresponds to each of these Pareto-optimal solutions. Thus, it is unlikely that there is a pseudo-polynomial time algorithm that can determine the entire set of Pareto-optimal solutions. As a result, the literature on scheduling with rejection is restricted to the definition of problem P4 where only a single Pareto-optimal solution is required for each Pareto-optimal point. As an outcome of this definition, the $1|rej|\#(C_{\max}(A)$, RC) problem, for example, is solvable in pseudo-polynomial time, although there are an exponential number of Pareto-optimal solutions. Theorem 3 below shows that there is a strong connection between problems P2–P4 in terms of the existence of a pseudo-polynomial time algorithm. This theorem is derived by a slight modification of Property 6 in T'kindt and Billaut (2006) with regard to the number of non-dominated solutions for a bicriteria scheduling problem.

Theorem 3 *If either problem P2 or P3 is ordinary $\mathcal{NP}$-hard, then the entire set of problems P2–P4 is ordinary $\mathcal{NP}$-hard.*

Proof Let us assume that there exists a pseudo-polynomial time algorithm for problem P2 with a running time of $O(f(n))$. The fact that the value of F_2 is upper bounded by $\sum_{j=1}^{n} e_j$, which is a numeric value of the input, implies that one can use the ϵ-constraint method to solve the P4 problem in $O(f(n) \sum_{j=1}^{n} e_j)$ time which is also a pseudo-polynomial time. This pseudo-polynomial time algorithm can then be used to solve problems P1 and P3 in pseudo-polynomial time as well. Since for the entire set of problems we survey here the F_1 value is also upper bounded by a numeric value of the input, we can say that if *DV* is an $\mathcal{NP}$-complete problem and if there exists a pseudo-polynomial time algorithm for at least one of the problems P2, P3, or P4, then problems P2–P4 are all ordinary $\mathcal{NP}$-hard. □

1.2 Approximation algorithms

Since large instances of $\mathcal{NP}$-hard problems cannot be optimally solved in reasonable time, a good heuristic algorithm that provides a feasible (but not necessarily optimal) solution in polynomial time is required. An important subclass of heuristic algorithms are approximation algorithms. The definition of an approximation algorithm for solving a problem which includes a single objective function (which may also be an aggregated function similar to our P1 problem) is presented below (the definition is restricted to minimization problems).

Definition 5 Let A be a heuristic algorithm with a solution value of $F_A(\mathcal{I})$ to a minimization problem with instance $\mathcal{I}$, and let $F^*(\mathcal{I})$ be the value of the optimal solution. Algorithm A is a ρ-approximation algorithm if for any instance $\mathcal{I}$ of the problem we have

$$\frac{F_A(\mathcal{I})}{F^*(\mathcal{I})} \leq \rho. \tag{2}$$

Definition 5 indicates that the uniqueness of an approximation algorithm is that it guarantees that the solution value it generates is at most ρ times the value of the optimal solution. Two common and important types of approximation algorithms are the polynomial time approximation scheme (PTAS), and the fully polynomial time approximation scheme (FPTAS), as defined below.

Definition 6 An approximation A_ε is said to be a polynomial time approximation scheme if the running time of the approximation is polynomial for n but not for ε, and the following ratio exists for every instance $\mathcal{I}$ of the problem:

$$\frac{F_{A_\varepsilon}(\mathcal{I})}{F^*(\mathcal{I})} \leq 1 + \varepsilon, \tag{3}$$

where $F_{A_\varepsilon}(\mathcal{I})$ denotes the approximation solution value for a given instance $\mathcal{I}$ and an ε value.

Definition 7 An approximation A_ε is said to be a *fully* polynomial time approximation scheme if the running time of the approximation is polynomial for both n and ε, and the ratio presented in Eq. (3) exists for any instance $\mathcal{I}$ of the problem.

Papadimitriou and Yannakakis (2000) consider an approximate version of the Pareto curve, the so-called ε-approximate Pareto curve, which is defined below.

Definition 8 Given $\varepsilon > 0$ and an instance $\mathcal{I}$, the Pareto ε-approximation set $P_\varepsilon(\mathcal{I})$ is a set in which for any Pareto-optimal solution $S \in P(\mathcal{I})$, there is a solution $S_\varepsilon \in P_\varepsilon(\mathcal{I})$ satisfying

$$\frac{F_1(S_\varepsilon)}{F_1(S)} \leq 1 + \varepsilon \text{ and } \frac{F_2(S_\varepsilon)}{F_2(S)} \leq 1 + \varepsilon, \tag{4}$$

where $P(\mathcal{I})$ is the entire set of Pareto-optimal solutions for instance $\mathcal{I}$.

Now we can easily convert Definitions 6 and 7 to define a PTAS and an FPTAS for obtaining the Pareto ε-approximation set P_ε (or a single solution S_ε within this set) by replacing the ratio in (3) by (4).

1.3 Organization of the paper

In several cases, we can directly derive results for scheduling problems with rejection from the analysis of equivalent scheduling problems from other fields in scheduling. Thus, in Sect. 2 we describe three such fields in scheduling: scheduling with controllable processing times, scheduling parallel machines with costs, and scheduling with due date assignment, all of which are highly connected to the field of scheduling with rejection. In Sect. 3, we present a survey of results for single machine problems, which is divided into subsections, based on the scheduling criterion used for F_1. Section 4 surveys multiple-machine scheduling problems with rejection. Section 4 is divided into subsections according to different machine configurations. Concluding remarks along with suggestions for future research are presented in the last section.

2 The connection between scheduling with rejection and other classes of scheduling problems

This section discusses scheduling with controllable processing times, scheduling parallel machines with costs, and scheduling with due date assignment, all of which, as mentioned, are highly connected to the field of scheduling with rejection. Below, we show how scheduling problems with rejection can be viewed either as equivalent to or as special cases of problems from each of these fields. Then, using these connections, we show throughout the survey how we can derive results for scheduling problems with rejection based on known results for scheduling problems from these closely related fields.

2.1 Scheduling with controllable processing times

When scheduling with controllable processing times (see Shabtay and Steiner 2007, for a survey paper in this field) the scheduler has to decide whether to divert more resources (such as manpower, heating, etc.) into a certain job in order to shorten (compress) its processing time. More formally, the processing time of job J_j on machine M_i is assumed to be a non-increasing function of the *amount of resource*, u_{ij}, allocated to the processing of operation O_{ij} for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The resource may be used in either *continuous* or *discrete* quantities. For the first case, the processing time is determined by the amount of a divisible resource, such as gas or electricity, that is allocated and therefore can vary continuously. For the second case, the amount of a divisible resource is indivisible, such as manpower and supporting equipment, and therefore the processing time has only a finite number of possible values. We include $dscr$ in the Y field of the three-field notation to indicate that the resource is used in discrete quantities. Otherwise, the resource is assumed to be divisible. In most studies on scheduling with controllable processing times, researchers assume that the job processing time is a bounded linear function of the amount of resource allocated to the processing of the job, i.e., the *resource consumption function* is of the form

$$p_{ij}(u_{ij}) = \overline{p}_{ij} - a_{ij}u_{ij}, \quad 0 \le u_{ij} \le \overline{u}_{ij} \le \overline{p}_{ij}/a_{ij}, \tag{5}$$

where $\overline{p}_{ij}$ is the *non-compressed processing time* for job J_j on machine M_i, $\overline{u}_{ij}$ is the *upper bound* on the amount of resource that can be allocated to perform job J_j on machine M_i, and a_{ij} is the positive *compression rate* of job J_j on machine M_i for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. In addition, a resource allocation cost function $c_{ij}(u_{ij})$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ is associated with the allocation of u_{ij} units of resource to operation O_{ij}.

Similarly to scheduling with rejection, scheduling with controllable processing times can be viewed as a bicriteria problem in which the first criterion, F_1, is a *scheduling criterion* that depends on the jobs' completion times while the second criterion,

$$F_2 = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}(u_{ij}),$$

is the total resource allocation cost function. The most commonly used function is the linear one for which

$$F_2 = \mathrm{TRAC} = \sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}u_{ij}, \tag{6}$$

where v_{ij} is the cost of one unit of resource allocated to operation O_{ij}. We include a lin entry in the Y field of the three-field notation to indicate that we are dealing with a scheduling problem with controllable processing times that has a linear type of resource consumption function (as given by (5)) and a linear resource allocation cost function (as given by (6)).

Theorem 4 *The DV of the $Rm|rej|\#(F_1, RC)$ problem polynomially reduces to the DV of the $Rm|lin, dscr|\#(F_1, TRAC)$ problem, subject to the condition that the contribution of any job with zero processing time to the value of the scheduling criterion, F_1, equals zero.*

Proof Given an instance $\mathcal{I} = \{\mathbf{p}, \mathbf{e}\}$ of the $Rm|rej|\#(F_1,\mathrm{RC})$ problem where $\mathbf{p} = (p_{ij})$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ and $\mathbf{e} = (e_1, \ldots, e_n)$, we construct an instance $\widetilde{\mathcal{I}}$ for the $Rm|lin, dscr|\#(F_1, \mathrm{TRAC})$ problem with $a_{ij} = \overline{p}_{ij}$, $\overline{u}_{ij} = 1$ and $v_{ij} = e_j$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

First, we show that if there exists a solution, S, for the $Rm|rej|\#\,(F_1, \mathrm{RC})$ problem with $F_1 \leq K$ and $\mathrm{RC} \leq R$, then there exists a schedule for the corresponding instance of the $Rm|lin, dscr|\#\,(F_1, \mathrm{TRAC})$ problem with $F_1 \leq K$ and $\mathrm{TRAC} \leq R$. Let A and $\overline{A}$ be the set of accepted and rejected jobs in S, respectively. Moreover, let $J_{i,[j]} \in A$ be the job in the jth position in the processing sequence on machine M_i for $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$, where n_i is the number of jobs assigned to machine M_i in S and $\sum_{i=1}^{m} n_i = |A| \leq n$. We construct the following solution, $\widetilde{S}$, for the $Rm|lin, dscr|\#\,(F_1, \mathrm{TRAC})$ problem from S: If $J_j \in A$, we set $u_j = 0$ and schedule job $J_{i,[j]}$ in the jth position in the processing sequence on machine M_i. Moreover, if $J_j \in \overline{A}$ in S, we set $u_j = 1$, and all the jobs belonging to set $\overline{A}$ in S are scheduled in $\widetilde{S}$ in any arbitrary sequence on any machine after the jobs from A have been completed. The fact that $u_j = 1$ for any $J_j \in \overline{A}$ and that $u_j = 0$ for any $J_j \in A$ implies that $\sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}u_{ij}(\widetilde{S}) = \sum_{J_j \in \overline{A}} v_{ij}(\widetilde{S}) = \sum_{J_j \in \overline{A}} e_j(S) \leq R$. Moreover, the fact that the completion time of any job $J_{i,[j]}$ with $u_{i[j]} = 0$ in $\widetilde{S}$ equals the completion time of the same job in S implies that if the contribution of any job with zero processing time to the scheduling criterion, F_1, equals zero, then $F_1(\widetilde{S}) = F_1(S) \leq K$.

Next, we show that if there exists a schedule, $\widetilde{S}$, for the corresponding instance of the $Rm|lin, dscr|\#\,(F_1, \mathrm{TRAC})$ problem with $F_1 \leq K$ and $\mathrm{TRAC} \leq R$, then a solution S for the $Rm|rej|\#\,(F_1, \mathrm{RC})$ problem with $F_1 \leq K$ and $\mathrm{RC} \leq R$ can be found. Let $J_{i,[j]}$ be the job in the jth position in the processing sequence on machine M_i for $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$ and n_i be the number of jobs assigned to machine M_i in $\widetilde{S}$ with $\sum_{i=1}^{m} n_i = n$. We construct the following solution, S, for the $Rm|rej|\#\,(F_1, \mathrm{RC})$ problem from $\widetilde{S}$: we set $A = \{J_{i,[j]} \in J | u_{i[j]} = 0\}$ and $\overline{A} = \{J_{i,[j]} \in J | u_{i[j]} = 1\}$. Moreover, we set the processing sequence on each of the m machines in S to be the same as in $\widetilde{S}$ despite the fact that rejected jobs are not scheduled at all. The fact that in $\widetilde{S}$ for any $J_{i,[j]}$ with $u_{i[j]} = 1$ we have that $v_{ij}u_{ij} = e_j$ and for any $J_{i,[j]}$ with $u_{i[j]} = 0$ we have that $v_{ij}u_{ij} = 0$ implies that $\sum_{J_j \in \overline{A}} e_j(S) = \sum_{i=1}^{m}\sum_{j=1}^{n} v_{ij}u_{ij}(\widetilde{S}) \leq R$. In addition, the fact that for any $J_{i,[j]}$ with $u_{i[j]} = 1$ we have that $p_{ij} = \overline{p}_{ij} - \overline{p}_{ij}u_{ij} = 0$ in $\widetilde{S}$ implies that although this job is scheduled in $\widetilde{S}$ and not in S, the completion time of any $J_{i,[j]}$ with $u_{i[j]} = 0$ in $\widetilde{S}$ equals the completion time of the same job in S. This further implies that if the contribution of any job with zero processing time to the scheduling criterion, F_1, equals zero, then $F_1(\widetilde{S}) = F_1(S) \leq K$. □

Since for $F_1 = \sum_{J_j \in A} C_j$ the contribution of a job with zero processing time to the scheduling criterion, F_1, equals zero we can conclude from Theorem 4 that the $Rm|rej|\#(\sum_{J_j \in A} C_j, \mathrm{RC})$ problem polynomially reduces to the $Rm|lin, dscr|\#(\sum_{j=1}^{n} C_j, \mathrm{TRAC})$ problem. However, since for $F_1 = L_{\max}(A)$, the contribution of a job with zero processing time to the scheduling criterion, F_1, may not be zero, we can conclude that the $Rm|rej|\#(L_{\max}(A), \mathrm{RC})$ problem does not polynomially reduce to the $Rm|lin, dscr|\#\,(L_{\max}(A), \mathrm{TRAC})$ problem.

The following *all-or-none* property is common for many $Rm|lin|F_1 + \mathrm{TRAC}$ problems with controllable processing times and a *continuous* type of linear resource consumption function as given by (5).

Definition 9 A scheduling problem with controllable processing times follows the *all-or-none* property if there exists an optimal solution for this problem in which each processing operation is either fully compressed (that is, $u_{ij} = \overline{u}_{ij}$ and thus $p_{ij}(u_{ij}) = \overline{p}_{ij} - a_{ij}\overline{u}_{ij}$), or not compressed at all (that is, $u_{ij} = 0$ and thus $p_{ij}(u_{ij}) = \overline{p}_{ij}$).

The following theorem shows that under some restricted conditions the DV of the $Rm|rej|F_1 + \mathrm{RC}$ problem polynomially reduces to the DV of the $Rm|lin|F_1 + \mathrm{TRAC}$ problem. The proof is omitted as it is similar to that of Theorem 4.

Theorem 5 *The DV of the $Rm|rej|F_1 + \mathrm{RC}$ problem polynomially reduces to the DV of the $Rm|lin|F_1 + \mathrm{TRAC}$ problem, subject to the conditions that the $Rm|lin|F_1 + \mathrm{TRAC}$ problem satisfies the above all-or-none property and that the contribution of any job with zero processing time to the value of the scheduling criterion, F_1, equals zero.*

The two conditions in Theorem 5 hold, for example, for the $Rm|lin|\sum_{j=1}^{n} C_j + \sum_{j=1}^{n} v_j u_j$ problem and thus the $Rm|rej|\sum_{J_j \in A} C_j + \mathrm{RC}$ problem polynomially reduces to this problem.

2.2 Scheduling parallel machines with costs

When scheduling parallel machines with costs, the processing of job J_j on machine M_i incurs a cost c_{ij} for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Thus, similar to scheduling with rejection, the quality of a solution is measured by two criteria: F_1 is the scheduling criterion and F_2 is the total processing cost. It is easy to see that a scheduling problem with rejection on a set of m parallel machines is a special case of scheduling parallel machines with costs on $m + 1$ machines in which $c_{ij} = 0$ for any $i = 1, \ldots, m$ and $j = 1, \ldots, n$ and $c_{ij} = e_j$ for $i = m + 1$ and $j = 1, \ldots, n$.

2.3 Scheduling with due date assignment

Meeting due dates has always been one of the most important objectives in scheduling. While traditional scheduling models consider due dates as given by exogenous decisions, in a more flexible and integrated system they are determined by taking into account the system's ability to meet the quoted delivery dates. For this reason, numerous studies have viewed

due date assignment as part of the scheduling process and show how the ability to control due dates can be a major factor for improving system performance. A diversity of due date assignment methods have been studied in the literature (see Gordon et al. 2002a,b, 2004; Kaminsky and Hochbaum 2004, for extensive surveys). We briefly present two of the more frequently used ones below.

- The *common* due date assignment method (referred to as CON), in which all jobs are assigned the same due date; that is $d_j = d$ for $j = 1, \ldots, n$, and $d \geq 0$ is an unrestricted decision variable.
- The *unconstraint* due date assignment method (usually referred to as DIF), in which each job can be assigned a *different* due date and there are no constraints on the due date values.

When studying a scheduling problem with assignable due dates, researchers have usually restricted their analysis to the objective of minimizing an integrated objective function of the following type:

$$F_1 + \gamma \sum_{j=1}^{n} d_j,$$

where F_1 is the scheduling (due date related) criterion and $\gamma \sum_{j=1}^{n} d_j$ is the due date assignment cost.

Let us first consider the single machine scheduling problem where the due dates are assignable according to the CON method and $F_1 = \sum_{j=1}^{n} w_j U_j$. De et al. (1991) present the following lemma.

Lemma 1 *The optimal common due date for any job sequence is given by* $d = C_{[l]} = \sum_{j=1}^{l} p_{[j]}$, *where* $C_{[l]}$ *is the completion time of a job in some position* $l (l \in \{0, 1, \ldots, n\})$ *in the sequence where* $C_{[0]} \stackrel{def}{=} 0$.

Based on Lemma 1, we can derive the following theorem.

Theorem 6 *The* $1|rej|C_{\max}(A) + RC$ *problem is equivalent to the* $1|CON| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ *problem.*

Proof Let us first show that the DV of the $1|rej|C_{\max}(A) +$ RC problem polynomially reduces to the DV of the $1|\text{CON}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem. To do so, given an instance $\mathcal{I} = \{\mathbf{p}, \mathbf{e}\}$ for the $1|rej|C_{\max}(A) +$ RC problem, where $\mathbf{p} = (p_1, \ldots, p_n)$ and $\mathbf{e} = (e_1, \ldots, e_n)$, we construct an instance $\widetilde{\mathcal{I}} = \{\mathbf{p}, \mathbf{w}\}$ for the $1|\text{CON}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem with $w_j = \gamma n e_j$ for $j = 1, \ldots, n$.

We now show that if there exists a solution, S, for the $1|rej|C_{\max}(A) +$ RC problem with $C_{\max}(A) + \text{RC} = \sum_{J_j \in A} p_j + \sum_{J_j \in \overline{A}} e_j \leq K$, then there exists a solution, $\widetilde{S}$, for the corresponding instance of the $1|\text{CON}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem with $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j \leq \gamma n K$. Let A and $\overline{A}$ be the set of accepted and rejected jobs in S, respectively. We define solution $\widetilde{S}$ as follows. We schedule the jobs in A before the jobs in $\overline{A}$ on the single machine. Moreover, we set $d = \sum_{J_j \in A} p_j$ and schedule the jobs in A and $\overline{A}$ in any arbitrary sequence. It follows that the jobs in $\overline{A}$ are tardy while all the jobs in A are completed no later than the common due date, d. Thus, we have that $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j = \sum_{J_j \in \overline{A}} w_j + \gamma n \sum_{J_j \in A} p_j = \gamma n \sum_{J_j \in \overline{A}} e_j + \gamma n \sum_{J_j \in A} p_j = \gamma n (\sum_{J_j \in \overline{A}} e_j + \sum_{J_j \in A} p_j) \leq \gamma n K$.

Next, we show that if there exists a schedule, $\widetilde{S}$, for the corresponding instance of the $1|\text{CON}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem with $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j \leq \gamma n K$, then a solution S for the $1|rej|C_{\max}(A) + \text{RC}$ problem with $C_{\max}(A) + \text{RC} = \sum_{J_j \in A} p_j + \text{RC} \leq K$ can be constructed. Due to Lemma 1, we may assume, without loss of generality, that the common due date coincides with the completion of some job in solution $\widetilde{S}$ (or is set to be equal to zero). Let A be the set of jobs which are completed no later than the common due date, d, in solution $\widetilde{S}$, and let $\overline{A} = J \setminus A$. We define solution S by accepting the jobs in A and rejecting the jobs in $\overline{A}$. The fact that for $\widetilde{S}$ we have that $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j = \sum_{J_j \in \overline{A}} w_j + \gamma n \sum_{J_j \in A} p_j = \gamma n (\sum_{J_j \in \overline{A}} e_j + \sum_{J_j \in A} p_j) \leq \gamma n K$ implies that $\sum_{J_j \in \overline{A}} e_j + \sum_{J_j \in A} p_j \leq K$.

In a similar way, we can easily prove that the DV of the $1|\text{CON}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem polynomially reduces to the DV of the $1|rej|C_{\max}(A)+\text{RC}$ problem by constructing an instance $\widetilde{\mathcal{I}} = \{\mathbf{p}, \mathbf{e}\}$ for the $1|rej|C_{\max}(A) +$ RC problem where $e_j = \frac{w_j}{\gamma n}$, given an instance $\mathcal{I} = \{\mathbf{p}, \mathbf{w}\}$ for the $1|\text{CON}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem. □

Let us next consider the $1|\text{CON}| \sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ problem. It is easy to prove that there exists an optimal schedule for the $1|\text{CON}| \sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ problem which satisfies the property in Lemma 1. Based on this property, we derive the following theorem.

Theorem 7 *The* $1|CON| \sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ *problem polynomially reduces to the* $1|rej| \sum_{J_j \in A} w_j C_j + RC$ *problem.*

Proof Given an instance $\mathcal{I} = \{\mathbf{p}, \mathbf{w}\}$ for the $1|\text{CON}| \sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ problem, where $\mathbf{p} = (p_1, \ldots, p_n)$ and $\mathbf{w} = (w_1, \ldots, w_n)$, we construct an instance $\widetilde{\mathcal{I}} = \{\mathbf{p}, \mathbf{w}, \mathbf{e}\}$ for the $1|rej| \sum_{J_j \in A} w_j C_j + \text{RC}$ problem where $e_j = \gamma n p_j$ for $j = 1, \ldots, n$.

We now show that if there exists a solution, S, for the $1|\text{CON}| \sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ problem with $\sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j \leq K$, then there exists a solution, $\widetilde{S}$, for the corresponding instance of the $1|rej| \sum_{J_j \in A} w_j C_j + \text{RC}$ problem with $\sum_{J_j \in A} w_j C_j + \sum_{J_j \in \overline{A}} e_j \leq K$. Due to Lemma 1, we may assume, without loss of generality, that the common due date coincides with the completion of some job in solution S. Let $\overline{A}$ be the set of jobs that are

scheduled no later than the common due date d in solution S, and let $A = J \setminus \overline{A}$. We define solution $\widetilde{S}$ by accepting the jobs in A and rejecting the jobs in $\overline{A}$. Moreover, we sequence the jobs in A in the same order as they are scheduled in solution S. Thus, for schedule $\widetilde{S}$ we have that $\sum_{J_j \in A} w_j C_j + \sum_{J_j \in \overline{A}} e_j = \sum_{j=1}^{l} w_{[j]} \sum_{i=1}^{j} p_{[i]} + \gamma n \sum_{J_j \in \overline{A}} p_j$, where $l = |A|$ and $[j]$ is the index of the jth job to be processed *tardy* by the single machine in solution S. The fact that for $\widetilde{S}$ we have that $\sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j \leq K$ implies that $\sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j = \sum_{J_j \in A} w_j T_j + \gamma n \sum_{J_j \in \overline{A}} p_j = \sum_{j=1}^{l} w_{[j]} (C_{[j]} - \sum_{J_j \in \overline{A}} p_j) + \gamma n \sum_{J_j \in \overline{A}} p_j = \sum_{j=1}^{l} w_{[j]} \sum_{i=1}^{j} p_{[i]} + \gamma n \sum_{J_j \in \overline{A}} p_j = \sum_{J_j \in A} w_j C_j + \sum_{J_j \in \overline{A}} e_j \leq K$.

Next, we show that if there exists a schedule, $\widetilde{S}$, for the corresponding instance of the $1|rej| \sum_{J_j \in A} w_j C_j$ + RC problem with $\sum_{J_j \in A} w_j C_j$ + RC $= \sum_{j=1}^{l} w_{[j]} \sum_{i=1}^{j} p_{[i]} + \sum_{J_j \in \overline{A}} e_j \leq K$, where $l = |A|$ and $[j]$ is the index of the jth job to be processed by the single machine in solution S, then a solution S for the $1|\text{CON}| \sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ problem with $\sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j \leq K$ can be constructed. Let A and $\overline{A}$ be the set of accepted and rejected jobs in $\widetilde{S}$, respectively. We define solution S as follows: we schedule the jobs in $\overline{A}$ before the jobs in A on the single machine. Moreover, we set $d = \sum_{J_j \in \overline{A}} p_j$ and schedule the jobs in A in the same order as in schedule $\widetilde{S}$. The jobs in set $\overline{A}$ are scheduled in any arbitrary sequence. The fact that for schedule $\widetilde{S}$ we have that $\sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j = \sum_{J_j \in A} w_j T_j + \gamma n \sum_{J_j \in \overline{A}} p_j = \sum_{J_j \in A} w_j T_j + \sum_{J_j \in \overline{A}} e_j = \sum_{j=1}^{l} w_{[j]} (C_{[j]} - \sum_{J_j \in \overline{A}} p_j) + \sum_{J_j \in \overline{A}} e_j = \sum_{j=1}^{l} w_{[j]} \sum_{i=1}^{j} p_{[i]} + \sum_{J_j \in \overline{A}} e_j = \sum_{J_j \in A} w_j C_j + \sum_{J_j \in \overline{A}} e_j \leq K$ completes our proof. □

Finally, we consider the $1|\text{DIF}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem for which Shabtay and Steiner (2006) presented the following result.

Lemma 2 *There exists an optimal solution in which the jobs are partitioned into a set of early jobs* (A) *which are scheduled first in a non-decreasing order of* p_j*, i.e., in an SPT order, and a set of tardy jobs* $(\overline{A})$ *which are scheduled in any arbitrary sequence. Moreover, in this optimal solution,* $d_j = C_j$ *for any* $J_j \in A$ *and* $d_j = 0$ *for any* $J_j \in \overline{A}$.

Based on Lemma 2, the following theorem can be derived.

Theorem 8 *The* $1|DIF| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ *problem is equivalent to the* $1|rej| \sum_{J_j \in A} C_j + RC$ *problem.*

Proof Let us first show that the DV of the $1|rej| \sum_{J_j \in A} C_j$ + RC problem polynomially reduces to the DV of the $1|\text{DIF}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem. To do so, given an instance $\mathcal{I} = \{\mathbf{p}, \mathbf{e}\}$ for the $1|rej| \sum_{J_j \in A} C_j$ + RC problem where $\mathbf{p} = (p_1, \ldots, p_n)$ and $\mathbf{e} = (e_1, \ldots, e_n)$, we construct an instance $\widetilde{\mathcal{I}} = \{\mathbf{p}, \mathbf{w}\}$ for the $1|\text{CON}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem with $w_j = \gamma e_j$ for $j = 1, \ldots, n$.

We now show that if there exists a solution, S, for the $1|rej| \sum_{J_j \in A} C_j$ + RC problem with $\sum_{J_j \in A} C_j + \sum_{J_j \in \overline{A}} e_j \leq K$, then there exists a solution, $\widetilde{S}$, for the corresponding instance of the $1|\text{DIF}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem with $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j \leq \gamma K$. Let A and $\overline{A}$ be the set of accepted and rejected jobs in S, respectively. We define solution $\widetilde{S}$ as follows: we schedule the jobs in A first according to their processing order in S. Then, we schedule the jobs in $\overline{A}$ in any arbitrary order. Moreover, we set $d_j = C_j$ for any $J_j \in A$ and $d_j = 0$ for any $J_j \in \overline{A}$. Thus, under $\widetilde{S}$ we have that $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j = \gamma(\sum_{J_j \in \overline{A}} e_j + \sum_{J_j \in A} C_j) \leq \gamma K$.

Next, we show that if there exists a solution, $\widetilde{S}$, for the corresponding instance of the $1|\text{DIF}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem with $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j \leq \gamma K$, then a solution S for the $1|rej| \sum_{J_j \in A} C_j$ + RC problem with $\sum_{J_j \in A} C_j$ + RC $\leq K$ can be constructed. Without loss of generality, we may assume that $\widetilde{S}$ has the properties of Lemma 2, i.e., the jobs are partitioned into a set of early jobs (A) which are scheduled first according to the SPT order, followed by a set of tardy jobs $(\overline{A})$ which are scheduled in any arbitrary sequence. Moreover, we have that $d_j = C_j$ for any $J_j \in A$ and $d_j = 0$ for any $J_j \in \overline{A}$ in solution $\widetilde{S}$. Given $\widetilde{S}$, we define solution S for the $1|rej| \sum_{J_j \in A} C_j$ + RC problem as follows: we accept the jobs in A and reject the jobs in $\overline{A}$. Moreover, we schedule the jobs in A according to the SPT rule on the single machine. The fact that for $\widetilde{S}$ we have that $\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j = \sum_{J_j \in \overline{A}} w_j + \gamma \sum_{J_j \in A} C_j = \gamma(\sum_{J_j \in \overline{A}} e_j + \sum_{J_j \in A} C_j) \leq \gamma K$ implies that $\sum_{J_j \in \overline{A}} e_j + \sum_{J_j \in A} C_j \leq K$ in S and completes our proof that the DV of the $1|rej| \sum_{J_j \in A} C_j$ + RC problem polynomially reduces to the DV of the $1|\text{DIF}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem.

In a similar way, we can easily prove that the DV of the $1|\text{DIF}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem polynomially reduces to the DV of the $1|rej| \sum_{J_j \in A} C_j + \sum_{j \in \overline{A}} e_j$ problem by constructing an instance $\widetilde{\mathcal{I}} = \{\mathbf{p}, \mathbf{e}\}$ for the $1|rej| \sum_{J_j \in A} C_j + \sum_{j \in \overline{A}} e_j$ problem with $e_j = \frac{w_j}{\gamma}$, given an instance $\mathcal{I} = \{\mathbf{p}, \mathbf{w}\}$ for the $1|\text{DIF}| \sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem. □

To conclude this subsection, we wish to emphasize that the set of connections presented here is only a subset of a wider set of connections that can be demonstrated between scheduling problems with rejection and scheduling problems with due date assignment. We believe that analyzing these connections could be the subject of a more extensive research study.

3 Single machine scheduling with rejection

This section is divided into subsections based on the scheduling criterion used for F_1. In Sect. 3.1, we review the results for problems with $F_1 = f_{\max}$ while problems with $F_1 = \sum_{J_j \in A} w_j C_j$ are reviewed in Sect. 3.2. Section 3.3 is devoted to a wide class of problems in which the scheduling criterion, F_1, can be represented by (or reduced to) a special unified formulation that includes *positional penalties*. In the last subsection, we review the results for all other scheduling criteria.

3.1 The maximum penalty criterion ($F_1 = f_{\max}$)

3.1.1 The makespan criterion ($F_1 = C_{\max}$)

De et al. (1991) present an $O(n)$ time procedure to solve the $1|\text{CON}|\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem. The fact that the $1|rej|C_{\max}(A) + \text{RC}$ problem is equivalent to the $1|\text{CON}|\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem (see Theorem 6) implies that the $1|rej|C_{\max}(A) + \text{RC}$ problem is solvable in $O(n)$ time as well. Cao et al. (2006), Zhang et al. (2009c), and Zhang et al. (2010) show that the $1|rej|\epsilon(C_{\max}(A)/\text{RC})$ problem is equivalent to the 0–1 knapsack problem (or other equivalent problems). Using this equivalence, various well-known algorithms such as pseudo-polynomial time algorithms and FPTASs for obtaining optimal and approximate solutions for the 0–1 knapsack problem can be used for solving the P2 problem as well (for an elaboration of various methods for solving the knapsack problem, see, e.g., Kellerer et al. 2004). Moreover, any pseudo-polynomial time algorithm that solves the 0–1 *knapsack problem* problem can be used as a subroutine for solving the $1|rej|\#(C_{\max}(A), \text{RC})$ problem in pseudo-polynomial time. This results in the following proposition.

Proposition 1 *The P2–P4 problems with $F_1 = C_{\max}(A)$ on a single machine are all ordinary $\mathcal{NP}$-hard.*

Cao and Zhang (2007) study the P1 problem with release dates, i.e., the $1|rej, r_j|C_{\max}(A) + \text{RC}$ problem. They prove that it is $\mathcal{NP}$-hard and design a PTAS for its solution but leave open the question of whether this problem is strongly or ordinary $\mathcal{NP}$-hard. Zhang et al. (2009b) solve this open question by providing two different pseudo-polynomial time algorithms to solve the $1|rej, r_j|C_{\max}(A) + \text{RC}$ problem. The first algorithm requires $O(n\sum_{j=1}^{n} e_j)$ time while the second requires $O(n(r_{\max} + \sum_{j=1}^{n} p_j))$ time, where $r_{\max} = \max_{j=1,\ldots,n}\{r_j\}$. Based on these two algorithms, they show that the $1|rej, r_j, e_j = e|C_{\max}(A) + \text{RC}$ and the $1|rej, r_j, p_j = p|C_{\max}(A) + \text{RC}$ problems are solvable in polynomial times of $O(n^2)$ and $O(n^3)$, respectively. Finally, they provide a 2-approximation algorithm and an FPTAS with a running time of $O(n^3/\varepsilon)$ for the solution of the $1|rej, r_j|C_{\max}(A) + \text{RC}$ problem.

In another study, Zhang et al. (2009c) analyze the P2 problem with release dates, i.e., the $1|rej, r_j|\epsilon(C_{\max}(A)/\text{RC})$ problem. They directly derive its $\mathcal{NP}$-hardness from the result in Proposition 1. In addition, they provide a dynamic programming (DP) procedure and an FPTAS for its solution, with running times of $O(n(\sum_{j=1}^{n} p_j + r_{\max}))$ and $O(\log_{1+\varepsilon}^{np_{\max}r_{\max}})$, respectively. In a different works, Zhang et al. (2010) remark that the above DP can solve the $1|rej, r_j, p_j = p|\epsilon(C_{\max}(A)/\text{RC})$ problem in $O(n^3)$ time. They also mention that finding a dual DP procedure for the P2 problem type would not be difficult, and that its running time would be $O(n\sum_{j=1}^{n} e_j)$. Furthermore, they note that this dual procedure can serve as an $O(n^2)$ time optimization algorithm for the special case of identical rejection penalties.

Cheng and Sun (2007) study two scheduling problems in which the job processing time is linearly deteriorating, i.e., the job processing time is a linear function of its starting time. The problems differ from each other by their linear deterioration function. In the first problem, there is a unified release date of $r_j = t_0$ for $j = 1, \ldots, n$ and the job processing time is given by $p_j = b_j t$, where b_j is the deteriorating rate of the processing time for job J_j and t is the starting time of job J_j. In the second problem, however, the job processing time is given by $p_j = a_j + b_j t$, where a_j is the basic processing rate. They show that even though the problems without rejection are polynomially solvable, adding the option of rejection makes them $\mathcal{NP}$-hard. The proof is done by a reduction from the *subset product problem*. Moreover, they prove that the problems are only ordinary $\mathcal{NP}$-hard by providing $O(n\sum_{j=1}^{n} e_j)$ time pseudo-polynomial time algorithms for their solution. These algorithms can also serve as a $O(n^2)$ time optimization algorithm to solve the special case of identical rejection costs. Then, they use the pseudo-polynomial time algorithm to generate an FPTAS for the first problem with a running time of $O(\frac{n^2}{\varepsilon}\sum_{j=1}^{n}\log(1 + b_j))$. Moreover, they show that the special case of the first problem in which $b_j = b$ for $j = 1, \ldots, n$ is solvable in $O(n^2)$ time.

Lu et al. (2009) study the $1|rej, r_j, p\text{-batch}, l \le b|C_{\max}(A) + \text{RC}$ problem, i.e., the bounded single machine parallel-batch scheduling problem with release dates, where p-batch refers to parallel-batch processing, and $l \le b$ implies that the number of jobs to be processed in a single batch, l, is upper bounded by a fixed integer, b, with $b < n$. Under the p-batch assumption, the processing time of each batch is derived from the processing time of the longest job within the batch. They show that this problem is strongly $\mathcal{NP}$-hard and provide a 2-approximation algorithm for its solution. They also present an $O(nlb^l p_{\max}^{l-1}(\sum_{j=1}^{n} e_j)(\sum_{j=1}^{n} p_j)^l)$

Table 1 Summary of complexity results for the makespan criterion

Global notation	P1		P2–P4	References
$1\|rej\|C_{\max}$, RC	$O(n)$		ONPH	De et al. (1991) and Cao et al. (2006)
$1\|rej, r_j\|C_{\max}$, RC		ONPH		Cao and Zhang (2007), Zhang et al. (2009b,c)
$1\|rej, r_j, e_j = e\|C_{\max}$, RC		$O(n^2)$		Zhang et al. (2009b, 2010)
$1\|rej, r_j, p_j = p\|C_{\max}$, RC		$O(n^3)$		Zhang et al. (2009b, 2010)
$1\|rej, r_j = t_0, p_j = b_j t\|C_{\max}$, RC	ONPH			Cheng and Sun (2007)
$1\|rej, r_j = t_0, p_j = b_j t, e_j = e\|C_{\max}$, RC	$O(n^2)$			Cheng and Sun (2007)
$1\|rej, p_j = a_j + b_j t\|C_{\max}$, RC	ONPH			Cheng and Sun (2007)
$1\|rej, p_j = a_j + bt\|C_{\max}$, RC	$O(n^2)$			Cheng and Sun (2007)
$1\|rej, p_j = a_j + b_j t, e_j = e\|C_{\max}$, RC	$O(n^2)$			Cheng and Sun (2007)
$1\|rej, r_j, p$-batch, $l \le b\|C_{\max}$, RC	SNPH			Lu et al. (2009)
$1\|rej, r_j = r, p$-batch, $l \le b\|C_{\max}$, RC	$O(n^2)$			Lu et al. (2009)
$1\|rej, r_j, p$-batch$\|C_{\max}$, RC	ONPH			Lu et al. (2008)
$1\|rej, r_j, p$-batch, $e_j = e\|C_{\max}$, RC	$O(n^3)$			Lu et al. (2008)

time algorithm to solve this problem which is then converted to a PTAS which runs in $O(nlb^l(2n^2/\varepsilon)^{2l})$ time, where $p_{\max} = \max_{j=1,\ldots,n}\{p_j\}$ and l are the number of different release dates. They further point out that for a constant number of different release dates the first algorithm can be considered as a pseudo-polynomial time algorithm and the second as an FPTAS. A polynomial time algorithm of $O(n^2)$ time for solving the case of identical release dates is also presented. Lu et al. (2008) study the unbounded version of the problem and show that it is $\mathcal{NP}$-hard. They also provide a pseudo-polynomial time algorithm which runs in $O(n^2\sum_{j=1}^{n} e_j)$ time, a 2-approximation algorithm, and an FPTAS with a running time of $O(n^4/\varepsilon)$. Moreover, they show that the special case in which jobs have the same rejection penalty can be solved in polynomial time of $O(n^3)$.

The complexity results that we survey in this subsection are summarized in Table 1. Note that ONPH and SNPH are used to indicate that a problem is ordinary or strongly $\mathcal{NP}$-hard, respectively.

3.1.2 *The maximal lateness and maximal tardiness criteria ($F_1 = L_{\max}(A)$ and $F_1 = T_{\max}(A)$)*

Sengupta (2003) was the first to address scheduling problems with rejection and either the maximal lateness or maximal tardiness criteria. Using a reduction from the *Partition Problem*, he shows that the $1|rej|L_{\max}(A)$ + RC and the $1|rej|T_{\max}(A)$ + RC problems are both ordinary $\mathcal{NP}$-hard. In addition, he provides two alternative pseudo-polynomial time algorithms for solving both problems with a running time of $O(n\sum_{j=1}^{n} e_j)$ and $O(n\sum_{j=1}^{n} p_j)$. This directly implies that both problems are solvable in polynomial time of $O(n^2)$ if either $p_j = p$ or $e_j = e$ for $j = 1,\ldots,n$. An FPTAS for solving the $1|rej|T_{\max}(A)$ + RC problem is also provided. The following proposition immediately follows from the fact that the DV problem (see Definition 3) with $F_1 = C_{\max}(A)$ is known to be an $\mathcal{NP}$-complete problem (see Cao et al. 2006; Zhang et al. 2009c, 2010) and is a special case of the DV problem with $F_1 = L_{\max}(A)$ or with $F_1 = T_{\max}(A)$.

Proposition 2 *Zhang et al. (2010) Problem DV with $F_1 = L_{\max}(A)$ or with $F_1 = T_{\max}(A)$ is $\mathcal{NP}$-complete.*

We note here that the first pseudo-polynomial time algorithm provided by Sengupta (2003), which was designed to solve the P1 problem, can easily be used to solve the P2 problem as well, as it finds the value of the optimal scheduling criterion for a given total rejection cost value. Moreover, the second pseudo-polynomial algorithm provided by Sengupta (2003) can be used to solve the P3 problem type since it finds an optimal total rejection cost under an upper bound on the maximal lateness value. Thus, the P1–P3 problems are all solvable in $O(n^2)$ time while problem P4 is solvable in $O(n^3)$ time if either $p_j = p$ or $e_j = e$ for $j = 1,\ldots,n$. Khuller and Mestre (2008) present a faster $O(n\log n)$ time optimization algorithm for the solution of the $1|rej, e_j = e|\epsilon(L_{\max}(A)/\text{RC})$ and the $1|rej, e_j = e|\epsilon(T_{\max}(A)/\text{RC})$ problems. We mention here that, although not explicitly stated, the algorithm by Khuller and Mestre can be used to solve the related P4 problems in $O(n^2\log n)$ time.

The classical definition of an approximation algorithm is that of a worst case relative error (see Definitions 5–7) which is a worst case factor by which the objective function value of the output solution differs from the optimal

Table 2 Summary of complexity results for the maximal lateness and maximal tardiness criteria

Global notation	P1	P2	P3	P4	References
$1\|rej\|L_{\max}$, RC			ONPH		Proposition 2, Theorem 3, and Sengupta (2003)
$1\|rej\|T_{\max}$, RC			ONPH		Proposition 2, Theorem 3, and Sengupta (2003)
$1\|rej, e_j = e\|L_{\max}$, RC	$O(n^2)$	$O(n \log n)$	$O(n^2)$	$O(n^2 \log n)$	Sengupta (2003) and Khuller and Mestre (2008)
$1\|rej, e_j = e\|T_{\max}$, RC	$O(n^2)$	$O(n \log n)$	$O(n^2)$	$O(n^2 \log n)$	Sengupta (2003) and Khuller and Mestre (2008)
$1\|rej, p_j = p\|L_{\max}$, RC		$O(n^2)$		$O(n^3)$	Sengupta (2003)
$1\|rej, p_j = p\|T_{\max}$, RC		$O(n^2)$		$O(n^3)$	Sengupta (2003)
$1\|r_j, rej\|L_{\max}$, RC			SNPH		Zhang et al. (2010) and Theorem 3
$1\|rej, r_j = t_0, p_j = b_j t\|L_{\max}$, RC	ONPH				Cheng and Sun (2007)
$1\|rej, r_j = t_0, p_j = b_j t\|T_{\max}$, RC	ONPH				Cheng and Sun (2007)

objective value. However, it seems that this way of measuring the quality of a heuristic algorithm is inappropriate for problems in which the optimal objective function value may be negative as in the case of the maximal lateness criterion. Thus, Sengupta suggests using an alternative notion of approximation, called *ε-optimization approximation*, which can accommodate such problems into an approximation framework. According to Sengupta (2003) and Orlin et al. (2000), a feasible solution S^* for a problem with input parameters e_j is said to be *ϵ-optimal* if S^* is optimal for a problem with *ε-perturbed* costs e'_j, i.e., costs e'_j satisfying the conditions $(1-\varepsilon)e_j \le e'_j \le (1+\varepsilon)e_j$ for all $e_j \ge 0$ and $(1+\varepsilon)e_j \le e'_j \le (1-\varepsilon)e_j$ for all $e_j < 0$. An *ε-optimization approximation scheme* provides an ε-optimal feasible solution for any $\varepsilon > 0$. An algorithm that produces an ε-optimization approximation is referred to as a polynomial time ε-optimization approximation scheme (PTEOS) if its running time is polynomial in the size of the instance. If, in addition, the running time is also polynomial in $1/\varepsilon$, the algorithm is called a fully polynomial time ε-optimization approximation scheme (FPTEOS). Sengupta (2003) presents an *PTEOS* for solving the $1|rej|L_{\max}(A)$ + RC problem in $O(n^{O(\log e_{\max}/\varepsilon)})$ time and an *FPTEOS* for solving the $1|rej|L_{\max}(A) + \prod_{J_j \in \overline{A}} e_j$ problem with a running time of $O(\frac{n}{\varepsilon}\sum_{j=1}^{n} \log e_j)$.

Zhang et al. (2010) study the $1|rej|\epsilon(L_{\max}(A)/\text{RC})$ problem and provide a DP procedure with a running time of $O(nR\sum_{j=1}^{n} p_j)$ for its solution. In addition, they note that when adding release dates, the resulting $1|r_j, rej|\epsilon(L_{\max}(A)/\text{RC})$ problem becomes strongly $\mathcal{NP}$-hard.

Cheng and Sun (2007) study the $1|rej, r_j = t_0, p_j = b_j t|L_{\max}(A)$+RC and the $1|rej, r_j = t_0, p_j = b_j t|T_{\max}(A)$ + RC problems. They note that both problems are $\mathcal{NP}$-hard due to the $\mathcal{NP}$-hardness of the same problem with $F_1 = C_{\max}(A)$, and provide a DP algorithm to solve the problems in $O(n\prod_{j=1}^{n}(1+b_j)\sum_{j=1}^{n} e_j)$ time. Then, they design an FPTAS for the $1|rej, r_j = t_0, p_j = b_j t|T_{\max}(A)$ + RC problem with a running time of $O\left(\frac{n^4}{\varepsilon^3}\sum_{j=1}^{n}\log(1+b_j)\log^2(\sum_{j=1}^{n} e_j)\right)$.

The complexity results presented in this subsection are summarized in Table 2.

3.2 Total-weighted completion time and total-weighted lateness criteria ($F_1 = \sum_{J_j \in A} w_j C_j$ and $F_1 = \sum_{J_j \in A} w_j L_j$)

The fact that $\sum_{j=1}^{n} w_j L_j = \sum_{j=1}^{n} w_j C_j - \sum_{j=1}^{n} w_j d_j$, and that $\text{WD} = \sum_{j=1}^{n} w_j d_j$ is a constant independent of the job schedule implies that the two problems, $1\,||\sum_{j=1}^{n} w_j C_j$ and $1\,||\sum_{j=1}^{n} w_j L_j$, are *equivalent*. However, with job rejection the problems are equivalent only when we consider the P1 problem variation. For this variation, we have that

$$\begin{aligned} G_l(F_1, F_2) &= F_1 + F_2 \\ &= \sum_{J_j \in A} w_j L_j + \sum_{J_j \in A} e_j \\ &= \sum_{J_j \in A} w_j C_j - \sum_{J_j \in A} w_j d_j + \sum_{J_j \in \overline{A}} e_j \\ &= \sum_{J_j \in A} w_j C_j - WD + \sum_{J_j \in \overline{A}} (e_j + w_j d_j), \end{aligned}$$

which implies that the $1|rej|\sum_{J_j \in A} w_j L_j$ + RC problem is equivalent to the $1|rej|\sum_{J_j \in A} w_j C_j$ + RC problem with a rejection cost of $\widetilde{e}_j = e_j + w_j d_j$ for $j = 1, \ldots, n$.

It is well known that the $1\,||\sum_{j=1}^{n} w_j C_j$ and the $1\,||\sum_{j=1}^{n} w_j L_j$ problems are solvable in $O(n \log n)$ time by sorting the jobs in a non-decreasing order of p_j/w_j, i.e., according to the WSPT order (see Smith (1956)). However, these problems become harder to solve with job rejection. Ghosh (1997) and Engels et al. (2003) study the equivalent P1 problem variations $1|rej|\sum_{J_j \in A} w_j L_j$ + RC and $1|rej|\sum_{J_j \in A} w_j C_j$ + RC, respectively. Due to the equivalence between the problems, it is not surprising that in both cases similar complexity

Table 3 Summary of relevant complexity results for the total-weighted completion time objective

Global notation	P1	P2–P4	References
$1\|rej\|\sum w_jC_j$, RC	ONPH[a]	ONPH	Ghosh (1997), Engels et al. (2003), Cao et al. (2006), and Theorem 3
$1\|rej, p_j = p\|\sum w_jC_j$, RC	$O(n^2)$[a]		Ghosh (1997) and Engels et al. (2003)
$1\|rej\|\sum C_j$, RC	$O(n^2)$[a]	ONPH	Ghosh (1997), Engels et al. (2003), Zhang et al. (2010), and Shabtay et al. (2012)
$1\|rej, r_j = t_0, p_j = b_jt\|\sum w_jC_j$, RC	ONPH		Cheng and Sun (2007)
$1\|rej, p_j = a_j + bt\|\sum C_j$, RC	$O(n^2)$		Cheng and Sun (2007)
$1\|rej$, p-batch, $l > n\|\sum C_j$, RC	$O(n^3 \log n)$		Li and Feng (2010)

[a]This result is applicable for $F_1 = \sum w_jL_j$ as well

Table 4 A list of scheduling criteria that fit the unified model

Scheduling criteria (F_1)	Positional penalty function
$C_{\max}$	$\xi_j(k) = 1$[a]
$\sum C_j$	$\xi_j(k) = j$[b]
$\delta_1\sum\sum\|C_i - C_j\| + \delta_2\sum C_j$	$\xi_j(k) = \delta_1 j(k-j) + \delta_2 j$[c]
$\delta_1\sum\sum\|W_i - W_j\| + \delta_2\sum W_j$	$\xi_j(k) = (j-1)(k-j+1)$[c]
$\alpha\sum E_j + \beta\sum T_j + \gamma d\,\|A\|$[d]	$\xi_j(k) = \begin{cases}\beta j & \text{for } j \le k-l\\ \alpha(k-j)+k\gamma & \text{for } j > k-l\end{cases}$[c]
$\alpha\sum E_j + \beta\sum T_j + \gamma_1\underline{d}\,\|A\| + \gamma_2 D\,\|A\|$[e]	$\xi_j(k) = \begin{cases}\beta j & \text{for } j \le k-l_2\\ \gamma_2 k & \text{for } k-l_2+1 \le j \le k-l_1\\ \gamma_1 k+\alpha(k-j) & \text{for } j > k-l_1.\end{cases}$[c]
$\alpha\sum E_j + \beta\sum T_j + \gamma\sum d_j^{\mathrm{f}}$	$\xi_j(k) = \begin{cases}\beta(j-1)+\gamma & \text{for } j \le k-l+1\\ \alpha(k-j+1)+\gamma(k+1) & \text{for } j > k-l+1.\end{cases}$[c]
$\alpha\sum E_j + \beta\sum T_j + \gamma\sum d_j^{\mathrm{g}}$	$\xi_j(k) = \xi_j = j\min(\beta,\gamma)$ for $j = 1,\ldots,k$[b]

[a]$\xi_j(k)$ is both position and k-independent
[b]$\xi_j(k)$ is k-independent and a monotonous function of j
[c]There exists a special case in which $\xi_j(k)$ is k-independent and is a monotonous function of j. l, l_1, and l_2 are constant values that can be obtained in constant time
[d]The due date (d) is assignable
[e]The scheduler has to assign a single time window $[\underline{d}, \underline{d}+D]$ for the completion time of each job
[f]All jobs are given equal slacks; that is, $d_j = p_j + s$, where $s \ge 0$ is a decision variable
[g]Each job can be assigned a different due date with no restrictions

results are obtained. They both provide an $\mathcal{NP}$-hard proof; a pseudo-polynomial time algorithm for solving these problems in $O(n\sum_{j=1}^{n} w_j)$ time, which can serve as an $O(n^2)$ time algorithm for the case of equal weights; an additional pseudo-polynomial time algorithm that can solve both problems in $O(n\sum_{j=1}^{n} p_j)$ time, which can also serve as an $O(n^2)$ time algorithm for the case of equal processing times; and an FPTAS with a running time of $O(n^2/\varepsilon)$. In addition to the above results, Engels et al. (2003) provide general techniques for designing approximation algorithms for scheduling with rejection based on reducing a problem with rejection to a potentially more complex scheduling problem without rejection. For example, they show how they can reduce an instance $\mathcal{I}$ of the $1|rej|\sum_{J_j\in A} w_jC_j +$ RC problem in an approximation-preserving manner to an instance of the $R||\sum_{J_j\in A} w_jC_j$ problem with $n+1$ unrelated machines working in parallel. Then, based on known approximation results for the $R||\sum_{J_j\in A} w_jC_j$ problem given by Chudak (1999) and Schulz and Skutella (1997), they conclude that the $1|rej|\sum_{J_j\in A} w_jC_j +$ RC problem has a 3/2-approximation algorithm. Using the same technique, and based on the approximation results by Schulz and Skutella (1997) for the $R|r_j|\sum_{J_j\in A} w_jC_j$ problem, they further conclude that the $1|rej, r_j|\sum_{J_j\in A} w_jC_j +$ RC problem has a 2-approximation algorithm. Another general technique that Engels et al. (2003) provide to obtain approximation results for scheduling problems with rejection based on the same problem without rejection is derived by exploiting the similarity between the LP relaxations of these problems. By applying this technique, they design a 4.5-approximation algorithm for the $1|rej, prec, r_j|\sum_{J_j\in A} w_jC_j +$ RC problem which runs in pseudo-polynomial time, where prec in the Y field implies that there are arbitrary precedence constraints between jobs. Moreover, they provide a polynomial

time $(4.5+\varepsilon)$-approximation algorithm for the same problem for any $\varepsilon > 0$.

We point out that the $1|\text{DIF}|\sum_{j=1}^{n} w_j U_j + \gamma \sum_{j=1}^{n} d_j$ problem is equivalent to the $1|rej|\sum_{J_j \in A} C_j$ + RC problem (see Theorem 8) and thus is solvable in $O(n^2)$ time as well (see also Koulamas 2010). Moreover, the algorithms designed by Engels et al. (2003) to solve the $1|rej|\sum_{J_j \in A} w_j C_j$ + RC problem can also be used to solve the $1|\text{CON}|\sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ problem, since it forms a special case of the $1|rej|\sum_{J_j \in A} w_j C_j$ + RC problem (see Theorem 7). The complexity status of the $1|\text{CON}|\sum_{j=1}^{n} w_j T_j + \gamma \sum_{j=1}^{n} d_j$ problem is, however, still an open question.

Slotnick and Morton (1996) provide an exact branch-and-bound (B&B) optimization algorithm to solve the $1|rej|\sum_{J_j \in A} w_j L_j$ + RC problem which they then convert into a heuristic algorithm by applying a *beam search* to eliminate nodes in the search tree. This heuristic runs in $O(n^4)$ time. They also present an additional *myopic heuristic* which runs in $O(n^4)$ time and evaluate its performance by performing a large scale computational study.

Cheng and Sun (2007) show that the $1|rej, r_j = t_0, p_j = b_j t|\sum_{J_j \in A} w_j C_j$ + RC problem is ordinary $\mathcal{NP}$-hard by providing a pseudo-polynomial time algorithm for its solution which runs in $O(n\sum_{j=1}^{n} e_j)$ time. Then, they use this algorithm to generate an FPTAS with a running time of $O(\frac{n^2}{\varepsilon^2}\log^2 \sum_{j=1}^{n} e_j)$. Cheng and Sun (2007) derive the $\mathcal{NP}$-hardness of the $1|rej, p_j = a_j + b_j t|\sum_{J_j \in A} w_j C_j$ + RC problem from the $\mathcal{NP}$-hardness of the same problem where rejection is not allowed (see Bachman et al. 2002) and show that for the special case in which $b_j = b$ and $w_j = 1$ for $j = 1, \ldots, n$, the problem is solvable in $O(n^2)$ time.

Cao et al. (2006) show that problem DV with $F_1 = \sum_{J_j \in A} w_j C_j$ is $\mathcal{NP}$-complete on a single machine. They also provide a pseudo-polynomial time algorithm with a running time of $O(n^3 p_{\max}^2 w_{\max})$ and an FPTAS with a running time of $O((1+\varepsilon)^2 n \log p_{\max} \log(n^2 p_{\max} w_{\max}))$ for the solution of the $1|rej|\epsilon(\sum_{J_j \in A} w_j C_j/\text{RC})$ problem where $w_{\max} = \max_{j=1,\ldots,n}\{w_j\}$. Both Zhang et al. (2010) and Shabtay et al. (2012) independently prove that a less general DV problem with $F_1 = \sum_{J_j \in A} C_j$ is $\mathcal{NP}$-complete as well. Shabtay et al. (2012) also design a pseudo-polynomial time algorithm and an FPTAS for the solution of the $1|rej|\#(\sum_{J_j \in A} C_j, \sum_{J_j \in \overline{A}} e_j)$ problem.

Moghaddam et al. (2010) present and test a multi-objective simulated annealing algorithm to solve the $1|rej|\#(\sum_{j \in A} w_j C_j, \sum_{j \in \overline{A}} e_j)$ problem. By comparing the results obtained by their algorithm with the set of exact Pareto-optimal solutions obtained from a complete enumeration, they conclude that the algorithm is reasonably good.

Li and Feng (2010) study the P1 version of the unbounded single machine parallel-batch processing scheduling problem with $F_1 = \sum_{J_j \in A} C_j$ and provide a DP procedure that solves the problem in $O(n^3 \log n)$ time.

Phillips et al. (2000) study the $1|rej, r_j, e_j = e|\epsilon(\sum_{J_j \in A} F_j/\text{RC})$ problem, where $F_j = C_j - r_j$ is the flow time of job J_j, and via linear programming (LP) relaxation and a rounding procedure obtain a polynomial time approximation algorithm that for any $\alpha\,(0 \le \alpha \le 1)$ produces a schedule for which at least $n\alpha/(1+\alpha)$ jobs are accepted and the value of F_1 is upper bounded by $(1/(1-\alpha) + o(1))\text{OPT}$, where OPT is the minimal value of F_1 when all jobs are accepted. They also show how this result can be extended to obtain similar approximation algorithms for *any* $F_1 = \sum_{J_j \in A} f_j(C_j)$ where f_j is a non-negative non-decreasing function of C_j for $j = 1, \ldots, n$. However, in contrast to the algorithm for $F_1 = \sum_{J_j \in A} F_j$ which runs in polynomial time, the algorithm for the more general scheduling criterion of $F_1 = \sum_{J_j \in A} f_j(C_j)$ runs in pseudo-polynomial time.

The complexity results presented in this subsection are summarized in Table 3.

3.3 A set of scheduling criteria that includes positional penalties ($F_1 = \sum_{j=1}^{k} \xi_j(k) p_{[j]}$)

Shabtay et al. (2012) provide a *unified* bicriteria analysis for a large set of single machine scheduling problems with job rejection whose scheduling criterion, F_1, can be either represented by or reduced to a special case of the following unified model:

$$F_1 = \sum_{j=1}^{k} \xi_j(k) p_{[j]}, \tag{7}$$

where $k = |A|$ and $\xi_j(k)$ is a *positional, k*-dependent, *and job-independent penalty* for *any* job scheduled in the jth position *from the end* in π, where π represents the permutation in which the jobs in set A are scheduled on the single machine. Here $[j]$ represents the job that is in the jth position *from the end* in π for $j = 1, 2, \ldots, k = |A|$. Table 4 below presents a subset of single machine scheduling problems of which scheduling criterion can be either represented by or reduced to the unified format in (7) with their specific $\xi_j(k)$ value.

Shabtay et al. (2012) provide an $O(n^3)$ time optimization algorithm for the *unified* $1|rej|\sum_{j=1}^{k} \xi_j(k) p_{[j]}$ + RC problem which is based on a DP procedure. Thereby they both extend and improve the results of Mosheiov and Sarig in (2009) who present an $O(n^4)$ time algorithms for solving two special cases of the unified model. The first is when $F_1 = \sum_{J_j \in A} E_j + \beta \sum_{J_j \in A} T_j + \gamma d\,|A| + \sum_{J_j \in \overline{A}} e_j$ and the common due date (d) is a decision variable, and the

Table 5 Summary of complexity results for the set of objectives which include positional penalties

[a] ξ_j is a monotonous function of j
[b] For any pair of jobs J_l and J_m: if $p_l \le p_m$ then $e_l \ge e_m$

Global notation	P1	P2–P4
$1\|rej\|\sum_{j=1}^{k}\xi_j(k)p_{[j]}$, RC	$O(n^3)$	ONPH
$1\|rej\|\sum_{j=1}^{k}\xi_j p_{[j]}$, RC[a]	$O(n^2)$	ONPH
$1\|rej\|\xi\sum_{j=1}^{k}p_{[j]}$, RC	$O(n)$	ONPH
$1\|rej$, agreeable[b]$\|\sum_{j=1}^{k}\xi_j(k)p_{[j]}$, RC	$O(n^2\log n)$	
$1\|rej$, agreeable[b]$\|\sum_{j=1}^{k}\xi_j p_{[j]}$, RC[a]	$O(n\log n)$	

second is when $F_1 = \alpha\sum_{J_j\in A}E_j+\beta\sum_{J_j\in A}T_j+\gamma_1\underline{d}\,|A|+\gamma_2 D\,|A|+\sum_{J_j\in\overline{A}}e_j$ and the scheduler has to assign a single desired time window, $[\underline{d},\underline{d}+D]$, for the completion time of each job (lines 5–6 of Table 4 show how these two problems can be viewed as special cases of the unified model). Shabtay et al. (2012) also show that the complexity of the DP can be reduced to $O(n^2)$ time if $\xi_j(k)=\xi_j$, i.e., $\xi_j(k)$ is k-independent. This result subsumes the result by Engels et al. (2003) who suggest an $O(n^2)$ time optimization algorithm for the $1|rej|\sum_{J_j\in A}C_j+\text{RC}$ problem (see line 3 in Table 3). Moreover, Shabtay et al. show that if $\xi_j(k)=\xi$ for $k=1,\ldots,n$ and $j=1,\ldots,k$, i.e., $\xi_j(k)$ is both position and k-independent, then the resulting $1|rej|\xi\sum_{j=1}^{k}p_{[j]}+\text{RC}$ problem can be solved in linear time. This result subsumes the result by De et al. (1991) who provide an $O(n)$ time optimization algorithm for the $1|rej|C_{\max}(A)+\text{RC}$ problem (see line 1 in Table 1).

Additional results obtained by Shabtay et al. (2012) for the unified model include a proof that problem DV is $\mathcal{NP}$-complete for *any* $F_1=\sum_{j=1}^{k}\xi_j p_{[j]}$, a pseudo-polynomial time optimization algorithm and an FPTAS for the solution of the general $1|rej|\#(\sum_{j=1}^{k}\xi_j(k)p_{[j]},\text{RC})$ problem, and two polynomial time optimization algorithms for the case where the following condition (which they refer to as an agreeable condition) holds, namely, for any l and m, the fact that $p_l\le p_m$ implies also that $e_l\ge e_m$. Note that the conditions that $e_j=e$ and/or $p_j=p$ are special cases of this agreeable condition.

Table 5 summarizes the complexity results in this section which were obtained by Shabtay et al. (2012).

3.4 Other scheduling criteria

Zhang et al. (2010) derive the $\mathcal{NP}$-hardness of the $1|rej|\epsilon(\sum_{J_j\in A}U_j/\text{RC})$ problem directly from the $\mathcal{NP}$-hardness of the $1|rej|\epsilon(C_{\max}(A)/\text{RC})$ problem and provide a DP algorithm for solving the more general $1|rej|\epsilon(\sum_{J_j\in A}w_jU_j/\text{RC})$ problem which runs in $O(nR\sum_{j=1}^{n}p_j)$ time. According to Theorem 3, by applying the ε- constraint method, this algorithm can be used to solve the more general $1|rej|\#(\sum_{J_j\in A}w_jU_j,\text{RC})$ problem in pseudo-polynomial time as well. This together with the fact that the $1||\sum_{J_j\in A}w_jU$ problem is known to be $\mathcal{NP}$-hard (see Karp 1972; Sahni 1976) implies that the entire set of P1–P4 problems with $F_1=\sum_{J_j\in A}w_jU_j$ are ordinary $\mathcal{NP}$-hard.

Steiner and Zhang (2011) study a scheduling problem where due dates are assignable according to the DIF method and the objective is to minimize a cost function which includes penalties due to tardy jobs and due date assignment. Moreover, for each job J_j, there is a lead time, A_j, which the customer considers to be acceptable and thus there is no penalty for assigning the due date to be no greater than A_j. However, if $d_j > A_j$ the due date assignment cost is $\alpha\max\{0,d_j-A_j\}$, where α is a positive integer that reflects the per unit of time cost of exceeding the acceptable lead time. The problem studied by Steiner and Zhang is referred to as the $1|\text{DIF}|\sum_{j=1}^{n}w_jU_j+\alpha\sum_{j=1}^{n}\max\{0,d_j-A_j\}$ problem. They show that this problem is equivalent to the $1|rej|\sum_{J_j\in A}T_j+\text{RC}$ problem, and then provide a pseudo-polynomial time optimization algorithm to solve the $1|rej|\sum_{J_j\in A}T_j+\text{RC}$ problem in $O(n^4(\sum_{j=1}^{n}p_j)^3)$ time. It is pointed out that this algorithm can solve the special case in which $p_j=p$ for $j=1,\ldots,n$ in $O(n^7)$ time. Moreover, they provide an FPTAS for solving the $1|rej|\sum_{J_j\in A}T_j+\text{RC}$ problem in $O\left(n^{10}\log n+n^{10}/\varepsilon+n^2\log n\log e_{\max}\right)$ time, where $e_{\max}=\max_{j=1,\ldots,n}\{e_j\}$.

Slotnick and Morton (2007) study the $1|rej|\sum_{J_j\in A}w_jT_j+\text{RC}$ problem and provide several heuristics and an exact B&B optimization algorithm that is based on linear (integer) relaxation. By performing a large scale computational experiment, the results obtained by each heuristic are compared to the results obtained by the B&B algorithm for small and medium size instances. For larger instances, where the exact B&B algorithm is not practical, they compare the results obtained by each heuristic to those obtained by a *beam search* heuristic. Rom and Slotnick (2009) use a genetic algorithm to solve the same problem. They conduct a computational study and show that for the entire set of tested instances the genetic algorithm performs better than an alternative myopic heuristic in terms of the objective function value, but at a cost of increased computational time. Nobibon and Leus (2011) study a different variants of the $1|rej|\sum_{J_j\in A}w_jT_j+\text{RC}$ problem for which the set of jobs, J, is partitioned into two disjoint subsets. The first, F,

consists of the firm's planned orders which must be accepted, while its complement $\overline{F}$ contains the *optional* jobs which may be either accepted or rejected. Following the scheduling notation introduced in T'kindt and Billaut (2006), we refer to this problem by $1|rej, \overline{F} \subseteq J|\sum_{J_j \in A} w_j T_j + \text{RC}$. Note that the $1|rej, \overline{F} \subseteq J|\sum_{J_j \in A} w_j T_j + \text{RC}$ problem may be viewed as a special case of the $1|rej|\sum_{J_j \in A} w_j T_j + \text{RC}$ problem for which the jobs in set F have an infinite rejection cost. Moreover, the $1|rej|\sum_{J_j \in A} w_j T_j + \text{RC}$ problem may be viewed as a special case of the $1|rej, \overline{F} \subseteq J|\sum_{J_j \in A} w_j T_j + \text{RC}$ problem for which $J = \overline{F}$. Thus, the two problems are in fact *equivalent*. Nobibon and Leus prove that unless $\mathcal{P} = \mathcal{NP}$, there is no constant-factor approximation algorithm for the $1|rej|\sum_{J_j \in A} w_j T_j + \text{RC}$ problem. Moreover, they provide two mixed integer linear programming (MILP) formulations and design two exact B&B optimization algorithms for the $1|rej|\sum_{J_j \in A} w_j T_j + \text{RC}$ problem. By conducting a computational study, they compare the efficiency and quality of the results obtained using the four different approaches where a commercial integer programming (IP) solver is used during the computational study to solve the two MILP formulations. Yang and Geunes (2007) study an extended version of the $1|rej|\sum_{J_j \in A} w_j T_j + \text{RC}$ problem for which the job processing times are controllable by a special case of the linear model in Eq. (5) and the objective function includes a resource allocation cost in addition to tardiness and rejection costs. They provide two heuristic algorithms and apply a set of computational experiments demonstrating the effectiveness of the proposed heuristics.

Bilgintürk et al. (2007) study a scheduling problem for which each accepted order has a gain of e_j for being accepted (which is equivalent to having a penalty of e_j for being rejected). In addition to the gain, the instance for their problem includes the job release dates, due dates and deadlines, and a matrix $\mathbf{S} = (s_{ij})$ of sequence-dependent set-up times, where s_{ij} is the set-up time required before starting the processing of job (order) J_j, given that it is processed right after order J_i. According to their problem definition, the manufacturer may complete each accepted order J_j until its deadline, $\overline{d}_j$, where $\overline{d}_j \geq d_j$. If job $J_j \in A$, its revenue R_j is calculated as $R_j = \max\{0, e_j - w_j T_j\}$ where $w_j = e_j/(\overline{d}_j - d_j)$ such that the revenue decreases down to zero if job J_j is completed at its deadline. The objective is to partition set J into $A \cup \overline{A}$ such that $\sum_{J_j \in A} R_j$ is maximized. We refer to this problem as $1|rej, s_{ij}, r_j, \overline{d}_j|\sum_{J_j \in A} R_j$ (note that here the objective has to be maximized). The $1|rej, s_{ij}, r_j, \overline{d}_j|\sum_{J_j \in A} R_j$ problem is an extension of the $1|s_{ij}|w_j T_j$ problem which is known to be strongly $\mathcal{NP}$-hard (see Lawler et al. 1982). Bilgintürk et al. (2007) provide an MILP formulation for the $1|rej, s_{ij}, r_j, \overline{d}_j|\sum_{J_j \in A} R_j$ problem. Moreover, they suggest using a simulated annealing procedure to heuristically solve larger instances. Other alternative heuristics for the same problem appear in Oğuz et al. (2010) and Cesaret et al. (2012).

The complexity results presented in this subsection are listed in Table 6.

4 Multiple-machine problems with rejection

This section is devoted to multiple-machine environments. The division into subsections is based on the machine configuration.

4.1 Parallel machines

In a parallel machine environment, job J_j requires a single operation and may be processed on any of the m machines. Three different systems of parallel machines are considered in the literature.

- *Identical machines* ($X = Pm$), where the job processing time is machine-independent; that is $p_{ij} = p_j$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.
- *Uniform machines* ($X = Qm$), where machine M_i has a *speed* of s_i, i.e., the processing time p_{ij} of job J_j on machine M_i is equal to p_j/s_i.
- *Unrelated machines* ($X = Rm$), where the job processing time is machine dependent.

The rest of this subsection is organized as follows. In Sects. 4.1.1 and 4.1.2, we survey the results for scheduling parallel machines with rejection with $F_1 = f_{\max}$ and $F_1 = \sum_{J_j \in A} w_j C_j$, respectively, while in Sect. 4.1.3 we

Table 6 Summary of relevant complexity results for other scheduling criteria

Global notation	P1	P2–P4	References
$1\|rej\|\sum U_j$, RC		ONPH	Zhang et al. (2010) and Theorem 3
$1\|rej\|\sum w_j U_j$, RC	ONPH		Zhang et al. (2010) and Theorem 3
$1\|rej\|\sum T_j$, RC	ONPH		Steiner and Zhang (2011)
$1\|rej, p_j = p\|\sum T_j$, RC	$O(n^7)$		Steiner and Zhang (2011)

provide an original contribution which proposes general schemes for scheduling unrelated machines with rejection.

4.1.1 Maximum penalty criterion ($F_1 = f_{\max}$)

Bartal et al. (2000) study the problem of minimizing the makespan on a set of identical machines in parallel and provide an FPTAS for solving the $Pm|rej|C_{\max}(A) + \text{RC}$ problem for any fixed m. Moreover, they provide an (2-1/m)-approximation algorithm for an arbitrary m value with a running time of $O(n \log n)$. Zhang et al. (2009c) study the $Pm|rej, r_j|\epsilon\ (C_{\max}(A)/\text{RC})$ problem and design a pseudo-polynomial time algorithm and an FPTAS for its solution.

Li and Yuan (2010) study the $Pm|rej, p_j = a_j + b_j t|C_{\max}(A) + \text{RC}$ problem whose $\mathcal{NP}$-hardness is directly derived from the $\mathcal{NP}$-hardness of the single machine version of this problem (see line 7 in Table 1). They construct an FPTAS with a running time of $O(n^{2m+2} \log^{m+2}(\max\{n, 1\backslash\varepsilon, a_{\max}, 1 + b_{\max}, e_{\max}\})/\varepsilon^{m+1})$ to solve this problem.

Hoogeveen et al. (2003) examine preemptive scheduling on parallel, related and unrelated machines. They show that the problems $Pm|rej,\ pmtn|C_{\max}(A)+\text{RC}$, $Qm|rej,\ pmtn|C_{\max}(A)+\text{RC}$ and $Rm|rej,\ pmtn|C_{\max}(A)+\text{RC}$ are all ordinary $\mathcal{NP}$-hard for $m \geq 2$. However, when m (the number of machines) is variable, the $R|rej,\ pmtn|C_{\max}(A)+\text{RC}$ problem becomes strongly $\mathcal{NP}$-hard while the other problems remain ordinary $\mathcal{NP}$-hard. In addition, they design FPTASs for several problems and a 1.58-approximation algorithm for the $R|rej,\ pmtn|C_{\max}(A) + \text{RC}$ problem.

Cao and Yang (2009) study the bounded parallel-batch scheduling problem with release dates on a set of m identical machines in parallel and generate a PTAS for solving this strongly $\mathcal{NP}$-hard $Pm|rej, r_j, p\text{-batch}, k \leq b|C_{\max}(A) + \text{RC}$ problem with a running time of $O((4b)^{1/\varepsilon} n^{3/\varepsilon+4} (1/\varepsilon)^{2/\varepsilon+2})$, where $b < n$ is the maximal batch size. They also show that the special case where jobs arrive simultaneously, i.e., $r_j = r$ for $j = 1, \ldots, n$, is solvable in $O(n^2 \log n)$ time. Miao et al. (2010) study the unbounded parallel-batch scheduling problem on a set of m unrelated machines under the assumption that the job processing times are job-independent, that is, $p_{ij} = p_i$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ and give a pseudo-polynomial time algorithm for the solution of the $Rm|rej,\ p_{ij} = p_i,\ p\text{-batch}|C_{\max}(A)+\text{RC}$ problem in $O\left(mn^2 \sum_{j=1}^{n} e_j\right)$ time. Note that this algorithm can solve the $Rm|rej,\ p_{ij} = p_i,\ p\text{-batch},\ e_j = e|C_{\max}(A)+\text{RC}$ problem in polynomial time.

Next, we present some known results for problems of scheduling parallel machines with costs. Since scheduling with rejection on a set of m parallel machines is a special case of scheduling parallel machines with costs (see Sect. 2.2), these results can be adopted for scheduling with rejection as well. Lin and Vitter (1992) study the P4 problem variation of scheduling *unrelated* parallel machines with costs and construct an ε-approximation algorithm that, given a cost C and a makespan T, finds in polynomial time a solution with a cost of at most $(1 + \varepsilon)C$ and a makespan with a value of at most $(2 + 1/\varepsilon)T$ (but only if there is a schedule with a cost of at most C and a makespan of at most T). Improved ε-approximation algorithms were later presented by Jansen and Porkolab (1999) and Angel et al. (2001). The latter algorithm is based on the well-known rounding method for converting a pseudo-polynomial time DP algorithm into an FPTAS. Trick (1994) studies an extension of the P1 problem variation of scheduling unrelated parallel machines with costs and presents a 2.618-approximation algorithm for $F_1 = C_{\max}$. Shmoys and Tardos (1993) improve this result by providing a 2-approximation algorithm for the same problem. Moreover, they provide an $O(mn^2 \log n)$ time algorithm for the P4 problem variation that finds a solution with a cost of at most C and a makespan of a cost of at most $2T$.

To the best of our knowledge, only Sengupta (2003) considers a scheduling problem with rejection in order to minimize the maximal lateness and the maximal tardiness criteria on a set of parallel machines. He presents an $O(nm2^m \prod_{i=1}^{m}(\sum_{j=1}^{n} p_{ij}))$ time algorithm to solve the $Rm||L_{\max}(A) + \text{RC}$ and $Rm||T_{\max}(A) + \text{RC}$ problems.

The complexity results presented in this subsection are summarized in Table 7.

4.1.2 Total-weighted completion times criterion ($F_1 = \sum_{J_j \in A} w_j C_j$)

The $Pm|rej|\epsilon(\sum_{J_j \in A} w_j C_j/\text{RC})$ problem is $\mathcal{NP}$-hard due to the $\mathcal{NP}$-hardness of the same problem on a single machine (Cao et al. 2006). Zhang et al. (2009a) study this problem and provide a pseudo-polynomial and an FPTAS for its solution.

Next, we show that several new results can be obtained for scheduling problems on a set of m identical machines with $F_1 = \sum_{J_j \in A} C_j$ as a direct outcome of the results obtained in Shabtay et al. (2012). For a set of m identical machines in parallel, the sum of completion time under an optimal schedule can be represented by (see, e.g., Gurel and Akturk (2007))

$$F_1 = \sum_{J_j \in A} C_j = \sum_{j=1}^{k} \left\lceil \frac{j}{m} \right\rceil p_j. \tag{8}$$

The scheduling criterion in (8) has the same unified format that appears in (7) with $\xi_j(k) = \left\lceil \frac{j}{m} \right\rceil$ for $j = 1, \ldots, k$. Moreover, since for this case $\xi_j(k)$ is k-independent and is a monotonous function of j, the following results can be directly obtained from the analysis in Shabtay et al. (2012) (see Table 5 lines 2 and 5): the $Pm|rej|\sum_{J_j \in A} C_j + \text{RC}$

Table 7 Summary of relevent complexity results for the maximal penalty criterion on parallel machines

Global notation	P1	P2–P4	References
$Pm\|rej\|C_{\max}$, RC	ONPH		Bartal et al. (2000)
$Pm\|rej, r_j\|C_{\max}$, RC		ONPH	Zhang et al. (2009c) and Theorem 3
$Pm\|rej, r_j = r$, p-batch, $k \le b\|C_{\max}$, RC	$O(n^2 \log n)$		Cao and Yang (2009)
$Pm\|rej, p_j = a_j + b_j t\|C_{\max}$, RC	ONPH		Li and Yuan (2010)
$Rm\|rej, pmtn\|C_{\max}$, RC	ONPH		Hoogeveen et al. (2003)
$R\|rej, pmtn\|C_{\max}$, RC	SNPH		Hoogeveen et al. (2003)
$Rm\|rej, p_{ij} = p_i$, p-batch$\|C_{\max}$, RC	ONPH		Miao et al. (2010)
$Rm\|rej, p_{ij} = p_i$, p-batch, $e_j = e\|C_{\max}$, RC	$O(mn^3)$		Miao et al. (2010)
$Rm\|rej\|C_{\max}$, RC	ONPH		Angel et al. (2001)
$Rm\|\|L_{\max}$, RC	ONPH		Sengupta (2003)
$Rm\|\|T_{\max}$, RC	ONPH		Sengupta (2003)

problem is solvable in $O(n^2)$ time, the $Pm|rej|\#(\sum_{J_j \in A} C_j$, RC) is ordinary $\mathcal{NP}$-hard and the $Pm|rej, agreeable|\#\left(\sum_{J_j \in A} C_j, \text{RC}\right)$ is solvable in $O(n \log n)$ time ($agreeable$ in the Y field of the three-field notation implies that for any $l, m \in \{1, \ldots n\}$ if $p_l \le p_m$, the condition that $e_l \ge e_m$ holds as well).

Li and Yuan (2010) study the $Pm|rej, p_j = b_j t, r_j = t_0|\sum_{J_j \in A} w_j C_j$ + RC problem whose $\mathcal{NP}$-hardness is directly implied from the $\mathcal{NP}$-hardness of the single machine version of the problem (see line 4 in Table 3). For this problem, they provide an FPTAS with a running time of $O(n^{2m+1} \log^m(1 + b_{\max})/\varepsilon^m)$. In addition, they provide a polynomial time solution algorithm for solving the $Pm|rej, p_j = a_j + bt|\sum_{J_j \in A} C_j$ + RC problem in $O(n^2)$ time.

Engels et al. (2003) provide a 3/2-approximation algorithm for the $R|rej|\sum_{J_j \in A} w_j C_j$ + RC problem and a 2-approximation algorithm for the $R|rej, r_j|\sum_{J_j \in A} w_j C_j$+ RC problem. Miao et al. (2010) study the unbounded parallel-batch scheduling problem on a set of m unrelated machines under the assumption that the job processing times are job-independent; that is, $p_{ij} = p_i$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. They also provide a pseudo-polynomial time algorithm for the solution of the $Rm|rej, p_{ij} = p_i$, p-batch$|\sum_{J_j \in A} w_j C_j$ + RC problem in $O(mn^3 p_m \sum_{j=1}^n e_j)$ time.

Alidaee and Ahmadian (1993) study the scheduling problem of minimizing the total completion time plus the total-weighted resource consumption, where the resource consumption function is given by the linear model in (5), and show that the resulting $Rm\,|lin|\sum_{j=1}^n C_j$ + TRAC problem is solvable in $O(n^3 m + n^2 m \log(nm))$ time by reducing it to an assignment problem. They also show that the same method can be used to solve the $Rm\,|lin, \text{CON}|\,\alpha \sum_{j=1}^n E_j + \beta \sum_{j=1}^n T_j$ + TRAC problem in $O(n^3 m + n^2 m \log(nm))$ time. Although not specifically stated, their results are applicable when the resource can be used in either discrete or continuous quantities. Since, for both problems, the contribution of a job with a zero processing time to the objective function value in an optimal schedule is zero, we can conclude from Theorem 5 that the $Rm\,|rej|\sum_{J_j \in A} C_j$ + RC problem is a special case of the $Rm\,|lin|\sum_{j=1}^n C_j$ + TRAC problem and that the $Rm\,|rej, \text{CON}|\,\alpha \sum_{J_j \in A} E_j + \beta \sum_{J_j \in A} T_j$+RC problem is a special case of the $Rm\,|lin, \text{CON}|\,\alpha \sum_{j=1}^n E_j + \beta \sum_{j=1}^n T_j$ + TRAC problem. This implies that these two problems are solvable in $O(n^3 m + n^2 m \log(nm))$ time as well.

The complexity results presented in this subsection are presented in Table 8.

4.1.3 *General optimization schemes for scheduling unrelated machines with rejection*

In this subsection, we present new results by providing general schemes for solving scheduling problems on unrelated parallel machines with rejection. These schemes offer unified pseudo-polynomial time solution methods for many $\mathcal{NP}$-hard scheduling problems that share two common properties: their scheduling criterion is *regular*, i.e., it is a non-decreasing function of the job completion times, and it is possible to index the jobs such that in an optimal schedule the jobs assigned to a given machine are scheduled in the order of their indices. These general schemes not only subsume several results that appear in the literature but also provide a solution method for problems that have not yet been studied.

Table 9 summarizes complexity results from the literature. Each of these results can be obtained by applying the unified optimization schemes presented in this subsection.

Rothkopf (1966) and Lawler and Moore (1969) suggest a general DP optimization algorithm for a fixed number of machines that can solve special cases of $Rm\,||\sum_{j=1}^n f_j$, for which f_j is a *regular* (non-decreasing) criterion for

Table 8 Summary of relevant complexity results for the total- weighted completion times objective on parallel machines

Global notation	P1	P2–P4	References
$Pm\|rej\|\sum w_jC_j$, RC	ONPH		Zhang et al. (2009a)
$Pm\|rej, p_{ij}=p_i, p\text{-batch}\|\sum w_jC_j$, RC	ONPH		Miao et al. (2010)
$Pm\|rej, p_j=b_jt, r_j=t_0\|\sum w_jC_j$, RC	ONPH		Li and Yuan (2010)
$Pm\|rej, p_j=a_j+bt\|\sum C_j$, RC	$O(n^2)$		Li and Yuan (2010)
$Pm\|rej\|\sum C_j$, RC	$O(n^2)$	ONPH	Shabtay et al. (2012)
$Pm\|rej$, agreeable[a]$\|\sum C_j$, RC	$O(n\log n)$		Shabtay et al. (2012)
$Rm\|rej\|\sum C_j$, RC	$O(n^3m+n^2m\log(nm))$		Alidaee and Ahmadian (1993)
$Rm\|rej$, CON$\|\alpha\sum_{J_j\in A}E_j+\beta\sum_{J_j\in A}T_j+\sum_{J_j\in\overline{A}}e_j$, RC	$O(n^3m+n^2m\log(nm))$		Alidaee and Ahmadian (1993)

[a]The conditions that $p_l\le p_m$ and $e_l\ge e_m$ are agreeable

$j=1,\ldots,n$ and when it is possible to index the jobs such that in an optimal schedule the jobs assigned to a given machine are scheduled in the order of their indices. The algorithm is described as follows.

Given an appropriate indexing $j=1,\ldots,n$ of the jobs, define $F_j(t_1,\ldots,t_m)$ as the minimum cost of a schedule for jobs $J_1,\ldots,J_j$ subject to the constraint that the last job on machine M_i is completed at time t_i for $i=1,\ldots,m$. Then, for the $\sum_{j=1}^n f_j$ criterion we have that

$$F_j(t_1,\ldots,t_m)=\max_{i=1,\ldots,m}\left\{F_{j-1}(t_1,\ldots,t_i-p_{ij},\ldots,t_m)+f_j(t_i)\right\}. \tag{9}$$

The initial conditions are

$$F_0(t_1,\ldots,t_m)=\begin{cases}0 & \text{if } t_i=0 \text{ for } i=1,\ldots,m\\ \infty & \text{otherwise}\end{cases} \tag{10}$$

and the optimal solution value is given by

$$F_n^*=\min(F_n(t_1,\ldots,t_m)\mid 0\le t_i\le C), \tag{11}$$

where $C=\max_{i=1,\ldots,m}(\sum_{j=1}^n p_{ij})$ is an upper bound on the completion time of any job in an optimal schedule. In general, these equations can be solved in $O(mnC^m)$ time. However, if the machines are uniform only $m-1$ of the $t_1,\ldots,t_m$ values are independent which means that for $m\ge 2$ uniform machines, the time complexity reduces to $O(mnC^{m-1})$. In both cases, however, the time complexity is pseudo-polynomial for a constant number of machines.

Below we show that three variants of the above optimization algorithm can solve the $Rm\,|rej|\sum_{J_j\in A}f_j+\text{RC}$, the $Rm\,|rej|\#(\sum_{J_j\in A}f_j,\text{RC})$ and the $Rm\,|rej|\#(f_{\max}(A),\text{RC})$ problems in pseudo-polynomial time. All of these algorithms are applicable for the case where f_j is a *regular* (non-decreasing) criterion for $j=1,\ldots,n$ and for when it is possible to index the jobs such that in an optimal schedule the jobs assigned to a given machine are scheduled in the order of their indices.

The $Rm\,|rej|\sum_{J_j\in A}f_j+\text{RC}$ problem The $Rm\,|rej|\sum_{J_j\in A}f_j+\text{RC}$ problem on m machines can be modeled as an $Rm\,||\sum_{j=1}^n f_j$ problem on $m+1$ machines where machine M_{m+1} is a dummy machine and all jobs that are assigned to this machine are rejected. Thus, by defining $p_{m+1,j}=0$ and $f_j(t_{m+1}=0)=e_j$ for $j=1,\ldots,n$, only states with $t_{m+1}=0$ have to be considered while implementing the recursion in (9). Therefore, the time complexity remains $O(mnC^m)$ for unrelated machines and $O(mnC^{m-1})$ for a uniform machines, and the following theorem holds.

Theorem 9 *The $Rm\,|rej|\sum_{J_j\in A}f_j+RC$ problem is solvable in $O(mnC^m)$ time and the $Qm\,|rej|\sum_{J_j\in A}f_j+RC$ problem is solvable in $O(mnC^{m-1})$ time for any f_j which is a regular criterion for $j=1,\ldots,n$ and when it is possible to index the jobs such that in an optimal schedule the jobs assigned to a given machine are scheduled in the order of their indices.*

Below, we present three different applications of Theorem 9. The first is for solving the $Qm\,|rej|\sum_{J_j\in A}w_jC_j+\text{RC}$ problem, which is known to be $\mathcal{NP}$-hard (see line 1 in Table 3). It should be noted that $\sum_{J_j\in A}w_jC_j$ is a regular criterion. In addition, it is known that in an optimal schedule the jobs on each machine are scheduled according to the WSPT order (see Smith 1956). It is easy to see that for identical or uniform parallel machines, the WSPT order of the jobs is the *same* no matter which machine is considered. Therefore, the WSPT order can be used as the common indexing of the jobs required by the DP algorithm defined by Eqs. (9–11). Moreover, this set of equations can be applied with $m+1$ machines and

Table 9 Related results from the literature that include pseudo-polynomial time algorithms for scheduling with rejection

Problem	Complexity	References	Indexing rule
$1\|rej, r_j\|C_{\max} + \text{RC}$	$O\left(n(r_{\max} + \sum_{j=1}^{n} p_j)\right)$	Zhang et al. (2009b)	Earliest release date (ERD)
$1\|rej, r_j\|\epsilon(C_{\max}/\text{RC})$	$O\left(n(r_{\max} + \sum_{j=1}^{n} p_j)\right)$	Zhang et al. (2010)	ERD
$Pm\|rej\|\epsilon(C_{\max}/\text{RC})$	$O\left(n(\sum_{j=1}^{n} p_j)^m\right)$	Zhang et al. (2009c)	Arbitrary
$Rm\|rej\|L_{\max} + \text{RC}$	$O\left(nm2^m \prod_{i=1}^{m}(\sum_{j=1}^{n} p_j)\right)$	Sengupta (2003)	Earliest due date (EDD)
$Rm\|rej\|T_{\max} + \text{RC}$	$O\left(nm2^m \prod_{i=1}^{m}(\sum_{j=1}^{n} p_j)\right)$	Sengupta (2003)	EDD
$1\|rej\|\epsilon\ (L_{\max}/\text{RC})$	$O\left(nR\sum_{j=1}^{n} p_j\right)$	Zhang et al. (2010)	EDD
$1\|rej\|\epsilon(\sum w_j U_j/\text{RC})$	$O\left(nR\sum_{j=1}^{n} p_j\right)$	Zhang et al. (2010)	EDD
$1\|rej\|\sum_{J_j\in A} w_j C_j + \text{RC}$	$O\left(n\sum_{j=1}^{n} p_j\right)$	Engels et al. (2003)	WSPT

$$f_j(t_i) = \begin{cases} w_j t_i & \text{for } i = 1, \ldots, m \\ e_j & \text{for } i = m+1 \end{cases} \quad \text{for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m \tag{12}$$

to solve the $Qm\,|rej|\sum_{J_j\in A} w_j C_j + \text{RC}$ problem. It is clear that $C = \max_{i=1,\ldots,m}\left\{\frac{1}{s_i} \times \sum_{j=1}^{n} p_j\right\}$ is an upper bound on the completion time of any job in the case of uniform machines and thus we have the following proposition.

Proposition 3 *The $Qm\,|rej|\sum_{J_j\in A} w_j C_j + RC$ is solvable in $O(mnC^{m-1})$ time for $m \geq 2$ and in $O(nC)$ time for $m = 1$, where $C = \max_{i=1,\ldots,m}\left\{\frac{1}{s_i} \times \sum_{j=1}^{n} p_j\right\}$.*

Note that the result obtained by Zhang et al. (2010) (see the last row of Table 9) can be directly derived as a special case of the result in Proposition 3.

The second application of Theorem 9 arises from the $Qm\,|rej, d_j = d|\sum_{J_j\in A} T_j + \text{RC}$ problem. Here again, the scheduling criterion, $\sum_{J_j\in A} T_j$, is regular. Moreover, since it is well known that there exists an optimal schedule for which the jobs on each machine are scheduled according to the SPT order, this order can be used as the common indexing of the jobs required by the DP algorithm defined by Eqs. (9–11). In addition, this set of equations can be applied with $m + 1$ machines and

$$f_j(t_i) = \begin{cases} \max\{0, t_i - d\} & \text{for } i = 1, \ldots, m \\ e_j & \text{for } i = m+1 \end{cases} \quad \text{for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m \tag{13}$$

to solve the $Qm\,|rej, d_j = d|\sum_{J_j\in A} T_j + \text{RC}$ problem. Thus, we have the following proposition as well.

Proposition 4 *The $Qm\,|rej, d_j = d|\sum_{J_j\in A} T_j + RC$ problem is solvable in $O(mnC^{m-1})$ time for $m \geq 2$ and in $O(nC)$ time for $m = 1$, where $C = \max_{i=1,\ldots,m}\left\{\frac{1}{s_i} \times \sum_{j=1}^{n} p_j\right\}$.*

Note that the $Pm\,|d_j = d|\sum_{i=1}^{n} T_i$ problem, which is a special case of the $Qm\,|rej, d_j = d|\sum_{J_j\in A} T_j + \text{RC}$ problem, is known to be $\mathcal{NP}$-hard for $m = 2$, since the problem of finding a schedule with value $\sum_{i=1}^{n} T_i = 0$ for the instance where $A = 1/2 \times \sum_{i=1}^{n} p_i$ is equivalent to the $\mathcal{NP}$-hard *multiprocessor scheduling problem* (see Garey and Johnson 1979). Therefore, the result in Proposition 4 implies that the $Qm\,|rej, d_j = d|\sum_{J_j\in A} T_j + \text{RC}$ problem is ordinary $\mathcal{NP}$-hard.

A third application of Theorem 9 arises from the $Rm\,|rej|\sum_{J_j\in A} w_j U_j + \text{RC}$ problem whose scheduling criterion is regular. Moreover, it is well known that there exists an optimal schedule for which the set of non-tardy jobs on each machine are ordered in a non-decreasing order of d_j, i.e., according to the EDD rule. Therefore, the DP algorithm defined by Eqs. (9–11) can be applied for solving the $Rm\,|rej|\sum_{J_j\in A} w_j U_j + \text{RC}$ problem. This can be done by assigning the early jobs to machines $M_1, \ldots, M_m$ according to the EDD rule and all rejected and late jobs to machine M_{m+1} in any arbitrary sequence. For this case, the set of equations defined by (9–11) can be applied with $m + 1$ machines,

$$f_j(t_i) = \begin{cases} 0 & \text{if } t_j \leq d_j \\ \infty & \text{if } t_j > d_j \end{cases} \quad \text{for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n \tag{14}$$

and

$$f_j(t_i) = \min(w_j, e_j) \text{ for } i = m+1 \text{ and } j = 1, \ldots, n \tag{15}$$

to solve the $Rm\,|rej|\sum_{J_j\in A} w_j U_j + \text{RC}$ problem. Thus, the following proposition holds.

Proposition 5 *The $Rm\,|rej|\sum_{J_j\in A} w_j U_j + RC$ problem is solvable in $O(mnC^m)$ time. Moreover, the $Qm\,|rej|\sum_{J_j\in A} w_j U_j + RC$ problem is solvable in $O(mnC^{m-1})$ time for $m \geq 2$.*

Note that the $1\,||\sum_{j=1}^{n} w_j U_j$ problem, which is a special case of the $Rm\,|rej|\sum_{J_j\in A} w_j U_j$ + RC problem, is known to be $\mathcal{NP}$-hard (see Karp 1972; Sahni 1976). Therefore, the result in Proposition 5 implies that the $Rm\,|rej|\sum_{J_j\in A} w_j U_j$ + RC problem is ordinary $\mathcal{NP}$-hard.

The $Rm\,|rej|\#(\sum_{J_j\in A} f_j, \mathrm{RC})$ problem Given an appropriate indexing $j = 1,\ldots,n$ of jobs, let us define $F_j(t_1,\ldots,t_m,E)$ as the minimum scheduling cost for a partial solution that includes jobs $J_1,\ldots,J_j$ subject to the constraint that the last job on M_i is completed at time t_i for $i = 1,\ldots,m$ and the total rejection cost is E. Then, all possible states can be calculated by the recursion in (16) as follows:

$$F_j(t_1,\ldots,t_m,E) = \min\begin{cases} \min_{i=1,\ldots,m}\{F_{j-1}(t_1,\ldots,t_i-p_{ij},\ldots,t_m,E)+f_j(t_i)\} \\ F_{j-1}(t_1,\ldots,t_m,E-e_j). \end{cases} \tag{16}$$

The initial conditions are

$$F_0(t_1,\ldots,t_m,E) = \begin{cases} 0 & \text{if } t_i = 0 \text{ for } i = 1,\ldots,m \text{ and } E = 0 \\ \infty & \text{otherwise} \end{cases} \tag{17}$$

and for any $0 \le E \le \sum_{j=1}^{n} e_j$, the minimal scheduling cost is given by

$$F_n^*(E) = \min(F_n(t_1,\ldots,t_m,\widetilde{E}) \mid \widetilde{E} \le E \text{ and } 0 \le t_i \le C), \tag{18}$$

where C is an upper bound on the completion time of any job in an optimal schedule. Note that by applying Eqs. (16–18) the entire set of weak Pareto-optimal and Pareto-optimal points are determined, which implies that for solving the $Rm\,|rej|\#(\sum_{J_j\in A} f_j, \mathrm{RC})$ problem, the weak Pareto-optimal solutions have to be eliminated. In general, Eqs. (16–18) can be solved in $O(mnC^m\overline{E})$ time where $\overline{E} = \sum_{j=1}^{n} e_j$. However, if the machines are uniform only $m-1$ of the $t_1,\ldots,t_m$ values are independent which means that for $m \ge 2$ uniform machines, the time complexity reduces to $O(mnC^{m-1}\overline{E})$. In both cases, however, the time complexity is pseudo-polynomial for a constant number of machines and the following theorem holds.

Theorem 10 *The $Rm\,|rej|\#(\sum_{J_j\in A} f_j, RC)$ problem is solvable in $O(mnC^m\overline{E})$ time and the $Qm\,|rej|\#(\sum_{J_j\in A} f_j, RC)$ problem is solvable in $O(mnC^{m-1}\overline{E})$ time for any f_j which is a regular criterion for $j = 1,\ldots,n$ and when it is possible to index the jobs such that in an optimal schedule the jobs assigned to a given machine are scheduled in the order of their indices.*

The $Qm\,|rej|\#(\sum_{J_j\in A} w_j C_j, \mathrm{RC})$, the $Qm|rej, d_j = d|\#(\sum_{J_j\in A} T_j, \mathrm{RC})$, and the $Rm\,|rej|\#(\sum_{J_j\in A} w_j U_j, \mathrm{RC})$ problems can be used as examples for an application of Theorem 10. For the first problem, the WSPT order can serve as the common indexing of the jobs and thus we can solve the $Qm\,|rej|\#(\sum_{J_j\in A} w_j C_j, \mathrm{RC})$ problem by applying Eqs. (16–18) with the $f_j(t_i)$ values in (12). For the second problem, the SPT order can serve as the common indexing of the jobs and thus we can solve the $Qm\,|rej, d_j = d|\#(\sum_{J_j\in A} T_j, \sum_{J_j\in\overline{A}} e_j)$ problem by applying Eqs. (16–18) with the $f_j(t_i)$ values in (13). For the last problem, the EDD order can serve as the common indexing of the jobs and thus the problem can be solved by applying Eqs. (16–18) with the values in (14) and (15). The following proposition now holds.

Proposition 6 *The $Qm\,|rej|\#(\sum_{J_j\in A} w_j C_j, RC)$ and the $Qm\,|rej, d_j = d|\#(\sum_{J_j\in A} T_j, RC)$ problems are solvable in $O(mnC^{m-1}\overline{E})$ time for $m \ge 2$ and in $O(nC\overline{E})$ time for $m = 1$. Moreover, the $Rm\,|rej|\#(\sum_{J_j\in A} w_j U_j, RC)$ problem is solvable in $O(mnC^m\overline{E})$ time.*

Note that the result obtained by Zhang et al. (2010) (which appears in the seventh row of Table 9) can be directly derived as a special case of the result in Proposition 6.

The $Rm\,|rej|\#(f_{\max}(A), \sum_{J_j\in\overline{A}} e_j)$ problem Given an appropriate indexing $j = 1,\ldots,n$ of jobs, let us define $F_j(t_1,\ldots,t_m,l)$ as the total rejection cost for jobs $J_1,\ldots,J_j$ subject to the constraint that the last job on M_i is completed at time t_i for $i = 1,\ldots,m$ and the value of the maximal scheduling criterion is at most l. Then, the following recursion holds:

$$F_j(t_1,\ldots,t_m,l) = \min\begin{cases} \begin{cases} \min\limits_{i=1,\ldots,m} F_{j-1}(t_1,\ldots,t_i-p_{ij},\ldots,t_m,l) & \text{if } f_j(t_i) \le l \\ \infty & \text{otherwise} \end{cases} \\ F_{j-1}(t_1,\ldots,t_m,l)+e_j \end{cases} \tag{19}$$

where $f_j(t_i)$ is the scheduling criterion value for job J_j if completed on machine M_i at time t_i. The initial conditions are

$$F_1(0,0,\ldots,0,\ldots,0,l) = e_1 \text{ and} \tag{20}$$

$$F_1(t_1,\ldots,t_m,l) = \begin{cases} 0 & \text{if } t_i = p_{i1},\ f_i(p_{i1}) \le l \text{ and } t_{-i} = 0 \text{ for } i = 1,\ldots,m \\ \infty & \text{otherwise} \end{cases}; \tag{21}$$

for any $l_{\min} \le l \le l_{\max}$, where $l_{\min}$ and $l_{\max}$ are lower and upper bounds on the value of the maximal scheduling criterion, respectively. Then, for any $l_{\min} \le l \le l_{\max}$ the minimal total rejection cost can be given by

$$F_n^* = \min(F_n(t_1,\ldots,t_m,\widetilde{l}) \mid \widetilde{l} \le l \text{ and } 0 \le t_i \le C). \tag{22}$$

By applying Eqs. (19–22) for any $l_{\min} \le l \le l_{\max}$, the entire set of weak Pareto-optimal and Pareto-optimal points can be determined. Then, for solving the $Rm\,|rej|\#(f_{\max}(A), \mathrm{RC})$

problem, the weak Pareto-optimal solutions have to be eliminated. Since the solution of the $Rm\,|rej|\,\#(f_{\max}(A), \mathrm{RC})$ problem requires calculating $F_j(t_1, \ldots, t_m, l)$ for any $j = 1, \ldots n$, $l_{\min} \le l \le l_{\max}$, and $0 \le t_i \le C$ for $i = 1, \ldots, m$, each of which requires $O(m)$ time, the following theorem holds.

Theorem 11 *The $Rm\,|rej|\,\#(f_{\max}(A), RC)$ problem is solvable in $O(mn(l_{\max} - l_{\min})C^m)$ time and the $Qm\,|rej|\,\#(f_{\max}(A), RC)$ problem is solvable in $O(mn(l_{\max} - l_{\min})C^{m-1})$ time, for any f_j which is a regular criterion for $j = 1, \ldots, n$ and when it is possible to index the jobs such that in an optimal schedule the jobs assigned to a given machine are scheduled in the order of their indices.*

Below, we present three different applications of Theorem 11. First, the $Rm\,|rej|\,\#(L_{\max}(A), \mathrm{RC})$ and the $Rm\,|rej|\,\#(T_{\max}(A), \mathrm{RC})$ problems are considered. For these problems, the EDD order can be used as the common indexing for the jobs and thus these problems can be solved by applying Eqs. (19–22) with $f_j(t_i) = t_i - d_j$ for the $Rm\,|rej|\,\#(L_{\max}(A), \mathrm{RC})$ problem and with $f_j(t_i) = \max\{t_i - d_j, 0\}$ for the $Rm\,|rej|\,\#(T_{\max}(A), \mathrm{RC})$ problem. Moreover, if $d_j \le C$ for $j = 1, \ldots, n$, the condition that $l_{\max} - l_{\min} = O(C)$ holds for both problems and the following proposition holds as well.

Proposition 7 *The P1–P4 problems with either $F_1 = L_{\max}$ or with $F_1 = T_{\max}$ are all solvable in $O(mnC^{m+1})$ time on unrelated machines. This time complexity can be reduced to $O(mnC^m)$ if the scheduling system includes $m \ge 2$ uniform machines.*

Note that the results obtained by Sengupta (2003) (which appears in the fourth and the fifth rows of Table 9) can be directly derived as special cases of the result in Proposition 7.

Next, the $Rm\,|rej, r_j|\,\#(C_{\max}(A), \mathrm{RC})$ problem is considered. For this problem, it is known that there exists an optimal schedule for which the jobs on each machine are scheduled in a non-decreasing order of r_j, i.e., according to the ERD order. Therefore, the ERD order can serve as the common indexing of jobs required by the DP algorithm defined by Eqs. (19–22) with $f_j(t_i) = t_i$. The condition that $r_j \le t_i - p_{ij}$ has to be added to the condition that $f_j(t_i) \le l$ in (19) and in the initialization in (21) the condition that $t_i = p_{i1}$ has to be replaced with the condition that $t_{i1} = p_{i1} + r_1$. Moreover, for the $Rm|rej, r_j|\#(C_{\max}(A), \mathrm{RC})$ problem the upper bound is $C = r_{\max} + \max_{i=1,\ldots,m}\left\{\sum_{j=1}^{n} p_{ij}\right\}$, $l_{\min} = 0$ and $l_{\max} \le C$. Thus, the following proposition holds.

Proposition 8 *The $Rm|rej, r_j|\#(C_{\max}(A), RC)$ problem is solvable in $O(mnC^{m+1})$ time. This time complexity can be reduced to $O(mnC^m)$ if the scheduling system includes $m \ge 2$ uniform machines.*

Table 10 Summary of additional results for parallel machine systems

Global notation	P1–P4	References
$Qm\,\|rej\|\sum w_jC_j$, RC	ONPH	Propositions 3 and 6
$Qm\,\|rej, d_j = d\|\sum T_j$, RC	ONPH	Propositions 4 and 6
$Rm\,\|rej\|\sum_{J_j \in A} w_jU_j$, RC	ONPH	Propositions 5 and 6
$Rm\,\|rej\|\,L_{\max}$, RC	ONPH	Proposition 7
$Rm\,\|rej\|\,T_{\max}$, RC	ONPH	Proposition 7
$Rm\,\|rej, r_j\|\,C_{\max}$, RC	ONPH	Proposition 8

Note that the results obtained by Zhang et al. (2009c) and Zhang et al. (2010) (which appears in the first two rows of Table 9) can be directly derived as special cases of the result in Proposition 8.

The complexity results presented in this subsection are presented in Table 10.

4.2 The flow-shop, job-shop, and open-shop scheduling systems

In a flow-shop system, the machines are linearly ordered and the jobs have to follow the same route from the first to the last machine. In a job-shop scheduling system each job has a predefined route to follow through the machines. However, different jobs may have different routes. In an open-shop system, each job needs to be processed exactly once on each of the machines, where the route of each job is up to the scheduler's decision. We include an Fm, Jm, or Om entry in the X field when referring to a scheduling problem in a flow-shop, job-shop, or open-shop scheduling system, respectively.

The two-machine flow-shop problem with rejection has been studied by Shabtay and Gaspar (2012) and Choi and Chung (2011). While Shabtay and Gaspar (2012) study the entire set of P1–P4 problems, Choi and Chung focus on the P1 problem. Both works include a proof that the $F2\,|rej|\,C_{\max}(A) + \mathrm{RC}$ problem is $\mathcal{NP}$-hard and provide a pseudo-polynomial time algorithm for its solution. However, it seems that the algorithm by Shabtay and Gaspar (2012) is more efficient as it runs in $O(n\sum_{j=1}^{n} p_{2j})$ time while that of Choi and Chung (2011) requires $O(n\Pi_{i=1}^{2}\sum_{j=1}^{n} p_{ij})$ time. Shabtay and Gaspar also convert their pseudo-polynomial time algorithm into an FPTAS with a running time of $O(n^3/\varepsilon)$ time. Additional results for the P1 problem on two machines include 2-approximation algorithms with a running time of $O(n \log n)$ (Shabtay and Gaspar 2012) and an $O(n^4)$ time algorithm for solving the special case for which $p_{ij} = p_j + v_i$ for $i = 1, 2$ and $j = 1, \ldots, n$ (Choi and Chung 2011). With regard to the P4 problem, Shabtay and Gaspar provide an $O(nP_2E)$ time pseudo-polynomial time algorithm for the solution of the $\mathcal{NP}$-hard $F2\,|rej|\,\#(C_{\max}(A), \mathrm{RC})$ problem, where $P_2 = \sum_{j=1}^{n} p_{2j}$ and $E = \sum_{j=1}^{n} e_j$. They also

provide an ε-approximation algorithm that, given the existence of a schedule with a total rejection cost of at most R and a makespan of at most K, finds in $O(n^5/\varepsilon^2)$ time a solution with a total rejection cost of at most $(1+\varepsilon)R$ and a makespan value of at most $(1+\varepsilon)K$. Note that according to Definition 8, this ε-approximation algorithm finds a single solution S_ε within the Pareto ε-approximation set P_ε only if (K, R) is a Pareto-optimal solution.

It should be noted that Shabtay and Gaspar (2012) failed to observe that some earlier results concerning problems P2–P4 can be derived from earlier results obtained by Jozefowska et al. (1994) who studied the $X\left|d_j=d\right|\sum_{j=1}^{n}w_jU_j$ problem and which are based on the following easy-to-prove theorem

Theorem 12 *The $X\left|d_j=d\right|\sum_{j=1}^{n}w_jU_j$ problem is equivalent to the $X\left|rej\right|\epsilon(RC/C_{\max}(A))$ problem.*

These authors prove that the $X\left|d_j=d\right|\sum_{j=1}^{n}U_j$ problem is $\mathcal{NP}$-hard for $X\in\{F2, J2, O2\}$ and propose pseudo-polynomial time algorithms to solve the $F2\left|d_j=d\right|\sum_{j=1}^{n}w_jU_j$ problem, the $O2\left|d_j=d\right|\sum_{j=1}^{n}w_jU_j$ problem in $O(nd^2)$ time, and the $J2\left|d_j=d\right|\sum_{j=1}^{n}w_jU_j$ problem in $O(nd^3)$ time. T'kindt et al. (2007) showed that the $F2\left|\text{CON}\right|\epsilon\left(\sum_{j=1}^{n}U_j/d\right)$ problem is *equivalent* to the $F2\left|d_j=d\right|\sum_{j=1}^{n}U_j$ problem and extended the the pseudo-polynomial time algorithm by Jozefowska et al. (1994) to solve the $F2\left|\text{CON}\right|\#\left(\sum_{j=1}^{n}U_j, d\right)$ problem (and therefore also the $F2\left|rej\right|\#\left(C_{\max}(A), \text{RC}\right)$ problem) in $O(nD^2)$ time, where D is an upper bound on the makespan value. Note that this time complexity is neither dominated by nor dominate the time complexity of the algorithm proposed by Shabtay and Gaspar (2012) for the equivalent $F2\left|rej\right|\#(C_{\max}(A), \text{RC})$ problem. However, we note that when $e_j=1$, the algorithm by Shabtay and Gaspar (2012) runs faster than the one by T'kindt et al. (2007). T'kindt et al. (2007) present several B&B algorithms for solving the $F2\left|\text{CON}\right|\epsilon\left(d/\sum_{j=1}^{n}U_j\right)$ problem which are based on problem-dependent cuts and covering inequalities, as well as on an initial preprocessing phase that enables a drastic reduction in the problem size. They perform an experimental study and show that the preprocessing phase enables them to fix at least 76 % of the problem variable. Moreover, they show that the best B&B algorithm is capable of solving instances of the $F2\left|\text{CON}\right|\epsilon\left(d/\sum_{j=1}^{n}U_j\right)$ problem with up to the huge amount of 3,000 jobs in less than 180 s on average and instances of the $F2\left|\text{CON}\right|\#\left(\sum_{j=1}^{n}U_j, d\right)$ problem with up to 500 jobs in less than 750 s on average.

Gaspar and Shabtay (2010) study two different special cases of the flow-shop problem on m machines with rejection. The first is where the processing times are machine-independent, i.e., $p_{ij}=p_j$ for $i=1,\ldots,m$ and $j=1,\ldots,n$, which is commonly known as the *proportionate* flow-shop problem. This model of processing times has been well studied in the literature (see, e.g., Pinedo 2008; Shakhlevich et al. 1998; Choi et al. 2007) and is supported by various practical applications in industry (see, e.g., Panwalker et al. 1973). The second is where the shop produces identical jobs; that is $p_{ij}=p_i$ for $i=1,\ldots,m$ and $j=1,\ldots,n$ (see, e.g., Mosheiov 2003; Mosheiov and Oron 2005 for other studies on flow-shop scheduling problems with identical jobs). The results obtained by Gaspar and Shabtay (2010) for m proportionate machines include an $O(n\log n)$ time optimization algorithm for the $Fm|rej,\ p_{ij}=p_j|C_{\max}(A)+\text{RC}$ problem, a proof that the DV problem with $F_1=C_{\max}(A)$ is $\mathcal{NP}$-complete, a pseudo-polynomial time algorithm, and an ε-approximation algorithm for the solution of the $Fm\left|rej,\ p_{ij}=p_j\right|\#\left(C_{\max}(A), \text{RC}\right)$ problem. In addition, for the case of identical jobs, Gaspar and Shabtay present an $O(n)$ time optimization algorithm for the $Fm\left|rej,\ p_{ij}=p_i\right|C_{\max}(A)+\text{RC}$ problem, as well as an $O(n\log n)$ time optimization algorithm for the solution of the $Fm\left|rej,\ p_{ij}=p_i\right|\#\left(C_{\max}(A), \text{RC}\right)$ problem.

Hoogeveen et al. (2003) study the problem of scheduling preemptive jobs in open-shop scheduling systems. They show that the $Om|rej,\ pmtn|C_{\max}(A)+\text{RC}$ problem is ordinary $\mathcal{NP}$-hard for a fixed m value ($m\geq 2$). However, when m (the number of machines) is unknown, the $O|rej,\ pmtn|C_{\max}(A)+\text{RC}$ problem becomes strongly $\mathcal{NP}$-hard. In addition, they design a 1.58-approximation algorithm for the $O|rej,\ pmtn|C_{\max}(A)+\text{RC}$ problem.

The complexity results presented in this subsection are summarized in Table 11.

5 Concluding remarks and future research

We presented a survey of results for scheduling problems with rejection. In addition to known results from the literature, we included an original contribution by suggesting general schemes for solving scheduling problems on unrelated parallel machines with rejection in pseudo-polynomial time. Although the field of scheduling with rejection has attracted much attention in the last decade, there are still many challenges for future research some of which we list below.

- In Sect. 2, we highlight the close connections between scheduling with rejection and other fields of research, particularly the field of scheduling with due date assignment for which we show that the $1|\text{CON}|\sum_{j=1}^{n}w_jT_j+\gamma\sum_{j=1}^{n}d_j$ problem *polynomially reduces* to the $1|rej|$

Table 11 Summary of complexity results for flow-shop, job-shop, and open-shop systems

Global notation	P1	P2–P4	References
$F2\,\|rej\|\,C_{\max}$, RC	ONPH		Jozefowska et al. (1994), Shabtay and Gaspar (2012), and Choi and Chung (2011)
$F2\,\|rej,\ p_{ij}=p_j+v_i\|\,C_{\max}$, RC	$O(n^4)$		Choi and Chung (2011)
$Fm\,\|rej,\ p_{ij}=p_j\|\,C_{\max}$, RC	$O(n\log n)$	ONPH	Gaspar and Shabtay (2010)
$Fm\,\|rej,\ p_{ij}=p_i\|\,C_{\max}$, RC	$O(n)$	$O(n\log n)$	Gaspar and Shabtay (2010)
$X\,\|rej\|\,C_{\max}$, RC$(X\in\{J2,O2\})$		ONPH	Jozefowska et al. (1994)
$Om\|rej,\ pmtn\|C_{\max}$, RC	ONPH		Hoogeveen et al. (2003)
$O\|rej,\ pmtn\|C_{\max}$, RC	SNPH		Hoogeveen et al. (2003)

$\sum_{J_j\in A} w_jC_j$ + RC problem, that the $1|rej|C_{\max}(A)$ +RC problem is *equivalent* to the $1|\text{CON}|\sum_{j=1}^{n} w_jU_j+\gamma\sum_{j=1}^{n} d_j$ problem, and that the $1|\text{DIF}|\sum_{j=1}^{n} w_jU_j+\gamma\sum_{j=1}^{n} d_j$ problem is *equivalent* to the $1|rej|\sum_{J_j\in A} C_j$ + RC problem. Other connections between these two closely related fields can be a subject for an extensive research study.

- In Sect. 4.1.3, we present general optimization schemes for solving scheduling problems with rejection on a set of unrelated machines in pseudo-polynomial time. An interesting question for future research is whether they can be converted into FPTASs.
- Lawler (1973) designed a very powerful polynomial time algorithm that solves the $1|prec|f_{\max}$ problem in $O(n^2)$ time where $f_{\max}=\max_{j=1,\ldots,n}\{f_j\}$ can be *any* regular criterion. However, it seems that all four variants of the more general $1|rej|f_{\max}(A)$, RC problem are $\mathcal{NP}$-hard even for the special cases of $f_{\max}(A)=L_{\max}(A)$ and $f_{\max}(A)=L_{\max}(A)$ (Sengupta 2003). An important objective for future research might be to see if the pseudo-polynomial time algorithms designed by Sengupta (2003) to solve all four variants of the $1|rej|L_{\max}(A)$, RC and the $1|rej|T_{\max}(A)$, RC problems can be extended to solve the more general $1|rej|f_{\max}(A)$, RC problem mentioned above in pseudo-polynomial time.
- A significant question for future research is whether the techniques provided by Engels et al. (2003) for designing approximation algorithms for scheduling with rejection, based on reducing a problem with rejection to a scheduling problem without rejection, can be used or extended to provide additional approximation results for scheduling problems with rejection. Another similar question is whether the pseudo-polynomial time approximation algorithm designed by Phillips et al. (2000) for the $|rej,r_j,e_j=e|\epsilon(\sum_{J_j\in A} f_j(C_j)/\text{RC})$ problem (where f_j can be *any* regular criterion) can be extended to provide similar approximation algorithms for problems on parallel machines.
- T'kindt et al. (2007) presented a B&B algorithm for solving the $F2\,|\text{CON}|\,\epsilon\left(d/\sum_{j=1}^{n} U_j\right)$ problem (which in equivalent to the $F2\,|rej|\,\epsilon\,(C_{\max}(A)/\text{RC})$ problem). They perform an experimental study and show that the algorithm is capable of solving instances with up to the huge amount of 3,000 jobs in less than 180 s on average. It is worthwhile to study if this approach can be adopted to design similar algorithms for other scheduling problems with rejection.
- There are many problems (mainly single machine problems) for which the P1 variation is solvable in polynomial time while variations P2–P4 are $\mathcal{NP}$-hard (e.g., the $1|rej|C_{\max}(A)$, RC and $1|rej|\sum_{j=1}^{k}\xi_j(k)p_{[j]}$, RC problems). We are currently working on designing a general approach to heuristically solve problem variations P2–P4 based on solving a series of P1 problem variations.
- By taking a closer look at Tables 1–11, which summarize the complexity results appear in the literature, one can easily observe that in many cases, the complexity of either the P1 problem variation or the P2–P4 problem variations is still an open question. For example, the P1 problem variation of the $1|rej|\sum_{J_j\in A} T_j$, RC problem is known to be ordinary $\mathcal{NP}$-hard (Steiner and Zhang 2011). However, the question of whether the P2–P4 variations are ordinary or strongly $\mathcal{NP}$-hard is still an open question. Another interesting example arises from the $O2\,|rej|\,C_{\max}(A)$, RC problem which is solvable in

$O(n \log n)$ time (see Gonzalez and Sahni 1976) if all jobs have to be processed and is known to be ordinary $\mathcal{NP}$-hard for the P2–P4 problem variations (Jozefowska et al. 1994). Thus, it is worth investigating whether the $O2\,|rej|\,C_{\max}(A) + \mathrm{RC}$ problem is solvable in polynomial time or not.

- There are many problems in the literature that have been extensively studied when rejection is not allowed, but yet have not been studied for the extended case where rejection is a valid option. One example is scheduling problems on a set of multipurpose machines (see Leung and Li 2008 for a survey paper). The two-machine flow-shop scheduling problem with a no-wait restriction and rejection also merits further analysis as do problems with batching options, precedence constraints and preemptions.

We hope that this survey will inspire new research on these open questions and will lead to further progress in the area of scheduling with rejection.

Acknowledgments We would like to thank the anonymous referees for their helpful and valuable comments which greatly helped us to improve the quality of this survey paper.

References

Alidaee, B., & Ahmadian, A. (1993). Two parallel machine sequencing problems involving controllable job processing times. *European Journal of Operational Research*, *70*(3), 335–341.

Angel, E., Bampis, E., & Kononov, A. (2001). A FPTAS for approximating the unrelated parallel machines scheduling problem with costs. *Lecture Notes in Computer Science*, *2161*, 194–205.

Bechman, A., Janiak, A., & Kovalyov, M. Y. (2002). Minimizing the total weighted completion time of deteriorating jobs. *Information Processing Letters*, *81*(2), 81–84.

Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J., & Stougie, L. (2000). Multiprocessor scheduling with rejection. *SIAM Journal of Discrete Mathematics*, *13*(1), 64–78.

Bilgintürk, Z., Oğuz, C., & Salman, S. (2007). Order acceptance and scheduling decisions in make-to-order systems. In *5th Multidisciplinary International Scheduling Conference: Theory and Applications, Paris, France*.

Cao, Z., Wang, Z., Zhang, Y., & Liu, S. (2006). On several scheduling problems with rejection or discretely compressible processing times. *Lecture Notes in Computer Science*, *3959*, 90–98.

Cao, Z., & Yang, X. (2009). A PTAS for parallel batch scheduling with rejection and dynamic job arrivals. *Theoretical Computer Science*, *410*, 2732–2745.

Cao, Z., & Zhang, Y. (2007). Scheduling with rejection and non-identical job arrivals. *Journal of Systems Science and Complexity*, *20*, 529–535.

Cesaret, B., Oğuz, C., & Salman, F. S. (2012). A tabu search algorithm for order acceptance and scheduling. *Computers and Operations Research*, *39*(6), 1197–1205.

Cheng, Y., & Sun, S. (2009). Scheduling linear deteriorating jobs with rejection on a single machine. *European Journal of Operational Research*, *194*(1), 18–27.

Choi, B. C., Yoon, S. H., & Chung, S. J. (2007). Minimizing maximum completion time in a proportionate flow shop with one machine of different speed. *European Journal of Operational Research*, *176*(2), 964–974.

Choi, B. C., & Chung, J. (2011). Two-machne flow shop schedulng problem with an outsourcing option. *European Journal of Operational Research*, *213*, 66–72.

Chudak, F. (1999). A min-sum 1.5-approximation algorithm for scheduling unrelated parallel machines. *Journal of Scheduling*, *2*(2), 73–77.

De, P., Ghosh, J. B., & Wells, C. E. (1991). Optimal delivery time quotation and order sequencing. *Decision Sciences*, *22*(2), 379–390.

Engels, D. W., Karger, D. R., Kolliopoulos, S. G., Sengupta, S., Uma, R. N., & Wein, J. (2003). Techniques for scheduling with rejection. *Journal of Algorithms*, *49*(1), 175–191.

Gaspar, N., & Shabtay, D. (2010). Various special cases of the multiple-machine flow-shop scheduling problem with rejection. Working Paper.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of $\mathcal{NP}$-completeness*. San Francisco: Freeman.

Ghosh, J. B. (1997). Job selection in a heavily loaded shop. *Computers and Operations Research*, *24*(2), 141–145.

Gonzalez, T., & Sahni, S. (1976). Open shop scheduling to minimize finish time. *Journal of Association of Computing Machinery*, *23*(4), 665–679.

Gordon, V., Proth, J. M., & Chu, C. B. (2002). A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research*, *139*(1), 1–25.

Gordon, V., Proth, J. M., & Chu, C. B. (2002). Due date assignment and scheduling: SLK, TWK and other due date assignment models. *Production Planning and Control*, *13*(2), 117–132.

Gordon, V., Proth, J. M., & Strusevich, V. A. (2004). Scheduling with due date assignment. In J. Y.-T. Leung (Ed.), *Handbook of scheduling* (pp. 21-1-21-22). Boca Raton, FL: CRC Press.

Graham, R. L., Lawler, E. L., & Lenstra, J. K. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, *4*, 287–326.

Guerrero, H. H., & Kern, G. M. (1988). How to more effectively accept and refuse orders. *Production and Inventory Management*, *29*(4), 59–63.

Gurel, S., & Akturk, M. S. (2007). Scheduling parallel CNC machines with time/cost trade-off considerations. *Computers and Operations Research*, *34*(9), 2774–2789.

Hoogeveen, H., Skutella, M., & Woeginger, G. J. (2003). Preemptive scheduling with rejection. *Mathematical Programming*, *94*(3), 361–374.

Hoogeveen, H. (2005). Multicriteria scheduling. *European Journal of Operational Research*, *167*, 592–623.

Jansen, K., & Porkolab, L. (1999). Improved approximation schemes for scheduling unrelated parallel machines. In *STOC '99 Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, Atlanta* (pp. 408–417).

Józefowska, J., Jurisch, B., & Kubiak, W. (1994). Scheduling shops to minimize the weighted number of late jobs. *Operations Research Letters*, *16*, 277–283.

Kaminsky, P., & Hochbaum, D. (2004). Due date quotation models and algorithms. In J. Y.-T. Leung (Ed.), *Handbook of scheduling* (pp. 20-1–20-22). Boca Raton, FL: CRC Press.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller & J. W. Thatcher (Eds.), *Complexity of computer computations* (pp. 85–103). New York: Plenum Press.

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). *Knapsack problems*. Berlin: Springer.

Khuller, S., & Mestre, J. (2008). An optimal incremental algorithm for minimizing lateness with rejection. *Lecture Notes in Computer Science*, *5193*, 601–610.

Koulamas, C. (2010). A faster algorithm for a due date assignment problem with tardy jobs. *Operations Research Letters*, *38*(2), 127–128.

Lawler, E. L., & Moore, J. M. (1969). A functional equation and its application to resource allocation and sequencing problems. *Management Science*, *16*(1), 77–84.

Lawler, E. L. (1973). Optimal sequencing of a single machine subject to precedence constraints. *Operations Research*, *26*, 544–546.

Lawler, E. L., & Lenstra, J. K. (1982). Recent developments in deterministic sequencing and scheduling: A survey. In J. K. Lenstra & A. H. G. Rinnooy Kan (Eds.), *Deterministic and stochastic scheduling* (pp. 35–73). Dordrecht: Dempster, Reidel.

Leung, J. Y. T., & Li, C. L. (2008). Scheduling with processing time restriction: A survey. *International Journal of Production Economics*, *116*(2), 251–262.

Li, X., & Feng, H. (2010). Minimize the sum of total completion time and total rejection penalties on a single batching machine. In *Proceeding of the International Conference on Information, Engineering* (pp. 200–202).

Lin, J. H., & Vitter, J. S. (1992). ε-Approximation algorithms with minimum packing constraint violation. In *STOC '92 Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing* (pp. 771–782).

Li, S., & Yuan, J. (2010). Parallel-machine scheduling with deteriorating jobs and rejection. *Theoretical Computer Science*, *411*, 3642–3650.

Lu, L., Cheng, T. C. E., Yuan, J., & Zhang, L. (2009). Bounded single-machine parallel-batch scheduling with release dates and rejection. *Computers and Operations Research*, *36*(10), 2748–2751.

Lu, L., Zhang, L., & Yuan, J. (2008). The unbounded parallel batch machine scheduling with release dates and rejection to minimize makespan. *Theoretical Computer Science*, *396*, 283–289.

Miao, C., Zhang, Y., & Wang, C. (2010). Bounded parallel-batch scheduling on unrelated parallel machines. *Lecture Notes in Computer Science*, *6124*, 220–228.

Moghaddam, A., Yalaoui, F., Amodeo, L., Karimi, B., & Jolai, F. (2010). Developing a technique for a bi-objective scheduling model with rejection by simulated annealing. In *MOSIM'10, Proceedings of the Eight International Conference of Modeling and Simulation, Hammamet*.

Mosheiov, G. (2003). Scheduling unit processing time jobs on an m-machine flow shop. *Journal of the Operational Research Society*, *54*(4), 437–441.

Mosheiov, G., & Oron, D. (2005). A note on flow-shop and job-shop batch scheduling with identical processing-time jobs. *European Journal of Operational Research*, *161*(1), 285–291.

Mosheiov, G., & Sarig, A. (2009). Scheduling and due-date assignment problems with job rejection. *Foundations of Computing and Decision Sciences*, *34*(3), 193–208.

McGovern, G., & Quelch, J. (2005). Outsourcing marketing. *Harvard Business Review*, *83*, 2–3.

Nobibon, F. T., & Leus, R. (2011). Exact algorithms for a generalization of the order acceptance and scheduling problem in a single-machine environment. *Computers and Operations Research*, *38*(10), 367–378.

Oğuz, C., Salman, S., & Bilgintürk, Z. (2010). Order acceptance and scheduling decisions in make-to-order systems. *International Journal of Production Economics*, *125*(1), 200–211.

Orlin, J. B., Schulz, A. S., & Sengupta, S. (2000). ε-Optimization schemes and L-Bit precision: Alternative perspectives in combinatorial optimization. In *STOC '00 Proceedings of the Thirty-Two Annual ACM Symposium on Theory of Computing*.

Panwalker, S. S., Dudek, R. A., & Smith, M. L. (1973). Sequencing research and the industrial problem. In S. E. Elmaghraby (Ed.), *Symposium on the theory of scheduling and its applications* (pp. 29–38). Berlin: Springer.

Papadimitriou, C. H., & Yannakakis, M. (2000). On the approximability of trade-offs and optimal access of web sources. In *Proceedings 41th Annual IEEE Symposium on Foundations of Computer Science, Redondo Beach, CA, USA* (pp. 86–92).

Phillips, C. A., Uma, R. N., & Wein, J. (2000). Off-line admission control for general scheduling problems. *Journal of Scheduling*, *3*, 365–381.

Pinedo, M. (2008). *Scheduling: Theory, algorithms and systems* (3rd ed.). Upper Saddle River, NJ: Prentice-Hall.

Rom, W. O., & Slotnick, S. A. (2009). Order acceptance using genetic algorithms. *Computers and Operations Research*, *36*(5), 1758–1767.

Rothkopf, M. H. (1966). Scheduling independent tasks on parallel processors. *Management Science*, *12*(5), 437–447.

Sahni, S. (1976). Algorithms for scheduling independent tasks. *Journal of the ACM*, *23*(1), 116–127.

Schulz, A. S., & Skutella, M. (1997). Random-based scheduling: New approximations and LP lower bounds. *Lecture Notes in Computer Science*, *1269*, 119–133.

Schulz, A. S., & Skutella, M. (1997). Scheduling-LPs bear probabilities: Randomized approximations for min-sum criteria. *Lecture Notes in Computer Science*, *1284*, 416–429.

Sengupta, S. (2003). Algorithms and approximation schemes for minimum lateness/tardiness scheduling with rejection. *Lecture Notes in Computer Science*, *2748*, 79–90.

Shabtay, D., & Steiner, G. (2006). Two due date assignment problems in scheduling a single machine. *Operations Research Letters*, *34*(6), 683–691.

Shabtay, D., & Steiner, G. (2007). A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, *155*(13), 1643–1666.

Shabtay, D., Gaspar, N., & Yedidsion, L. (2012). A bicriteria approach to scheduling a single machine with rejection and positional penalties. *Journal of Combinatorial Optimization*, *23*(4), 395–424.

Shabtay, D., & Gaspar, N. (2012). Two-machine flow-shop with rejection. *Computers and Operations Research*, *39*(5), 1087–1096.

Shakhlevich, N., Hoogeveen, H., & Pinedo, M. (1998). Minimizing total weighted completion time in a proportionate flow shop. *Journal of Scheduling*, *1*(3), 157–168.

Shmoys, D. B., & Tardos, E. (1993). An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, *62*, 461–474.

Slotnick, S. A., & Morton, T. E. (1996). Selecting jobs for a heavily loaded shop with lateness penalties. *Computers and Operations Research*, *23*(2), 131–140.

Slotnick, S. A., & Morton, T. E. (2007). Order acceptance with weighted tardiness. *Computers and Operations Research*, *34*, 3029–3042.

Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, *3*, 59–66.

Steiner, G., & Zhang, R. (2011). Revised delivery-time quotation in scheduling with tardiness penalties. *Operations Research*, *59*, 1504–1511.

T'kindt, V., & Billaut, J.-C. (2006). *Multicriteria scheduling: Theory, models and algorithms* (2nd ed.). Berlin: Springer.

T'kindt, V., & Della Croce, F. (2007). Enumeration of Pareto optima for a flowshop scheduling problem with two criteria. *INFORMS Journal of Computing*, *19*(1), 64–72.

Trick, M. A. (1994). Scheduling multiple variable-speed machines. *Operations Research*, *42*(2), 234–248.

Yang, B., & Geunes, J. (2007). A single resource scheduling problem with job-selection flexibility, tardiness costs and controllable processing times. *Computers and Industrial Engineering*, *53*, 420–432.

Zhang, S., Cao, Z., & Zhang Y. (2009a). Scheduling with rejection to minimize the total weighted completion time. In *ISORA'09* (pp. 111–114).

Zhang, L., Lu, L., & Yuan, J. (2009b). Single machine scheduling with release dates and rejection. *European Journal of Operational Research*, *198*(3), 975–978.

Zhang, L., Lu, L., & Yuan, J. (2010). Single-machine scheduling under the job rejection constraint. *Theoretical Computer Science*, *411*, 1877–1882.

Zhang, Y., Ren, J., & Wang, C. (2009c). Scheduling with rejection to minimize the makespan. *Lecture Notes in Computer Science*, *5573*, 411–420.