Complexity in Numerical Optimization, 203-253
P.M. Pardalos, Editor

A Classification of Static Scheduling Problems

Jeffrey W. Herrmann
Chung-Yee Lee
Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611

Jane L. Snowdon
IBM Corporation, CIM Technology Center, 051-1/2103, Boca Raton, FL 33432

Abstract

In the last four decades, scheduling problems have received much attention by researchers. Recently, the Just-in-Time concept has inspired a renewed interest in scheduling, especially among industry practitioners. Although a number of papers have reviewed this field, this paper presents an easy-to-use reference guide of static scheduling problems with complexity results for practitioners, students, and others. Every attempt has been made to be complete; however, this survey is not exhaustive. This paper includes both regular and non-regular measures of performance, separate sections on dual criteria and multicriteria problems, and a discussion of recent developments in heuristic search approaches to these problems.

Keywords: Complexity, production planning and scheduling, sequencing, heuristic search.

1 Introduction

This paper presents a survey of static scheduling problems in an easy-to-use reference guide. To the best of the authors' knowledge, the only paper to provide such a review is Lageweg et al. (1982). This paper differs from Lageweg et al. in the following way: instead of being organized by complexity results, this paper classifies problems by their common characteristics. In addition, recent results and more sophisticated heuristics are included.

The paper has seven tables listing a number of deterministic machine scheduling problems and their algorithms and complexities. The paper divides the problems into three primary categories: one-machine, parallel-machine, and shop problems; it also covers four additional topics: dual-criteria problems, resource-constrained problems, stochastic scheduling problems, and heuristic searches, called in this paper smart-and-lucky searches.

Recently, some researchers have applied heuristic searches, such as simulated annealing, tabu search, and genetic algorithms, to scheduling problems. These searches are smart enough to escape most local optima; still, they must be lucky to find the global optimum. Some papers have shown promising results and research is continuing. This paper includes some of these studies in the tables.

Works on scheduling problems include the following books: Conway, Maxwell, and Miller (1967); Baker (1992); Rinnooy Kan (1976); Lenstra (1985); French (1982); Dempster, Lenstra, and Rinnooy Kan (1982); and Morton (1992). Surveys have been done by Graves (1981) and Lawler, Lenstra, Rinnooy Kan, and Shmoys (1989), which was a major source of information for some of these tables. Baker and Scudder (1990) also provided information on the earliness-tardiness problems. Fundamental papers on simulated annealing include Kirkpatrick et al. (1983), Cerny (1983), and Aarts and Van Laarhoven (1985); on tabu search, Glover (1989, 1990); and on genetic algorithms Holland (1975), Liepins and Hilliard (1989), Goldberg (1986), and Davis (1991).

2 Problem description

In deterministic machine scheduling, a set of m machines must process a set of n jobs, and all problem data is known in advance. The machine environment, job characteristics, objective function, and notation for the deterministic machine scheduling problem are defined in this section.

2.1 Machine environment

The first element of the problem description is the machine environment. A job may consist of one or more operations. If each job has only one operation, the environment is a single-machine problem or a parallel machine problem, where the job may be processed by any of the machines. Parallel machines may be identical, uniform, or unrelated machines.

Identical machines process jobs with the same speed.
Uniform machines have machine-dependent speeds.
Unrelated machines have machine-and-job-dependent speeds.

For shops, each job has a fixed sequence of operations requiring different machines. A shop may be a flow shop or a job shop. In a **job shop**, jobs may have different operation sequences. In a **flow shop**, all jobs have the same operation sequence.

2.2 Job characteristics

Each problem has a set of job characteristics, which may occur in any combination. **Preemption** (abbreviated **pmtn**) refers to environments in which a job's processing may be interrupted and later resumed (possibly on another machine). The jobs to be processed can have **precedence constraints** (abbreviated **prec**), that is, some jobs cannot be started until others are completed. The graph of these constraints may resemble a **tree** , where each job will have either a maximum of one successor (**intree**) or a maximum of one predecessor (**outtree**). The jobs may not be available until their individual **release dates** into the shop. Jobs can have **deadlines** which must be met or a **common due date** (all jobs are due at the same time). The jobs or operations can have unit or equal **processing requirements.**

2.3 Objective function

The objective function to be maximized or minimized is the third element of a problem description. This may be a sum of variables or the maximum of some variable or function. Typical objective functions include the following performance measures: **flowtime** is the sum of completion times; **makespan** is the maximum completion time. **Lateness** is the difference between the due date and completion time; this value can be positive (and thus is a measure of **tardiness**) or negative (**earliness**).

One section of the Table 1 focuses on single-machine **earliness-tardiness** problems. Earliness-tardiness problems have been the focus of much recent study, since they model some aspects of the Just-in-Time philosophy, in which it is desirable that jobs finish close to their due dates. This deviation from the due date is the sum of the earliness and tardiness, since each quantity is a positive deviation in one of two directions. In addition, researchers have studied different performance measures, such as the square of the deviation, the penalty cost of the deviation, and the cost of delivering the jobs.

2.4 Notation

The problem descriptions include a number of abbreviations and symbols that represent characteristics and functions or variables associated with deterministic scheduling problems. Included here is a list of these symbols and their meaning.

m	the number of machines.
n	the number of jobs.
J_j	job j, where $j = 1, \ldots, n$.
M_i	machine i, where $i = 1, \ldots, m$.
α_j	the unit earliness penalty of job j.
α	a common unit earliness penalty, all $\alpha_j = \alpha$.
β_j	the unit tardiness penalty of job j.
β	a common unit tardiness penalty, all $\beta_j = \beta$.
C_j	the completion time of job j.
C_{ij}	the completion time of operation i of job j.
C_{max}	the makespan, the maximum of C_j over all jobs j.
d_j	the due date of job j.
D_j	the deadline for the completion time of job j.
e_j	the start of a due date window for job j.
E_j	the earliness of job j; $E_j = max\ \{d_j - C_j, 0\}$.
E_{max}	the maximum earliness of a set of jobs.
f_j	a regular (nondecreasing) function of a job's completion time.
f_{max}	the maximum of the f_j over all jobs j.
I_{ij}	the idleness of operation i of job j; $I_{ij} = C_{i+1,j} - p_{i+1,j} - C_{ij}$.
L_j	the lateness of job j equals $C_j - d_j$.
L_{max}	the maximum lateness.
O_{ij}	the ith operation of job j.
p_j	the processing time of job j.
p_{ij}	the processing time of operation i of job j.
Q	the number of batch deliveries.
r_j	the release date of job j.
T_j	the tardiness of job j; $T_j = max\ \{C_j - d_j, 0\}$.
T_{max}	the maximum tardiness of a set of jobs; this also refers to the minimum possible T_{max}.
U_j	$U_j = 1$ if job j is tardy; otherwise $U_j = 0$.
U_{min}	the minimum possible number tardy.
w_j	the weight associated with job j.

The following definitions are used only in Table 6 for stochastic scheduling problems:

exp	exponentially distributed
LEPT	longest expected processing time first
SEPT	shortest expected processing time first
$E[C_{max}]$	expected makespan
$E[\Sigma C_j]$	expected flowtime
$E[\Sigma U_j]$	expected number of tardy jobs
$E[\Sigma f_j]$	expected sum of f_j
iid	identically, independently distributed

3 Table organization

Tables 1 through 8 list a number of machine scheduling problems in these areas: one-machine problems, dual-criteria problems, parallel-machine problems, shop problems, resource-constrained problems, stochastic scheduling problems, and heuristic searches. The problems in each table are grouped by some common characteristic or objective function. Each of these tables is composed of three columns, as described below.

Some of these tables deserve additional comment. Table 2 includes some previously-studied **dual-criteria** and **multicriteria** single-machine scheduling problems. In the dual-criteria problems, the primary objective is used as a constraint on the feasible schedules, and the secondary criteria is minimized over this more limited set. In multicriteria problems, the aim is to find efficient solutions that cannot be dominated by any other solution in all criteria simultaneously.

In Table 5, **resource-constrained project scheduling**, jobs may require the use of a part of some limited resource during job execution. In Table 6, **stochastic scheduling**, some problem data may be unknown at the beginning. Usually, this occurs in processing times that are random variables (traditionally exponentially distributed). Table 7 includes a number of papers that use **smart-and-lucky** searches to solve scheduling problems. Table 8 lists some problems that have a **class scheduling** structure; that is, the jobs to be scheduled are grouped into job classes, where a setup is performed when the machine switches from one job class to another.

These tables form an easy-to-use reference guide for practitioners, students, and others. Every attempt has been made to be complete; however, this survey is not exhaustive.

3.1 Column one: Problem description

The problem description is given in the first column. The standard description of a scheduling problem includes three elements: machine environment (denoted by x), job characteristics (denoted by y), and objective function (denoted by z); this description is expressed in a three- field classification: $x/y/z$.

The first field is the machine environment. The symbol may be the number 1, which denotes a single-machine problem. Parallel machine problems are denoted by the letters **P**, **Q**, or **R**, as follows:

P: identical machines
Q: uniform machines
R: unrelated machines

Two additional letters identify multi-operation (shop) problems:

J: job shop
F: flow shop

In any of these cases, a number appearing after the symbol denotes that the number of machines is fixed at this value (for example, **F3** denotes a three-machine flow shop).

The second field contains any special job characteristics, which may occur in any combination. If the second field is blank, the jobs are assumed to have individual due dates and be immediately available, non-preemptive, and without precedence constraints between jobs.

The objective function to be minimized is the third field. This can be a sum of variables or the maximum of some variable or function. The following examples may serve to clarify this notation:

$1/r_j/L_{max}$	One-machine problem with jobs that have unequal release dates; the objective function is the maximum lateness.
$1/r_j, p_j = p/L_{max}$	Same as above but processing times are identical.
$1//\Sigma C_J$	One-machine problem with all jobs available at time 0; the objective function is the sum of completion times, i.e., flowtime.
P/pmtn/C_{max}	Identical parallel machine problem with preemption; the objective function is the maximum completion time, i.e., makespan.
F2//C_{max}	Two-machine flowshop problem; the objective function is makespan.
J//$Cmax$	General jobshop problem; the objective function is makespan.

3.2 Column two: Complexity

The second column of the table contains information about the solution for the problem. For example, if the problem is **NP-hard** (or strongly NP-hard), then **NP** as well as an error bound for an approximate solution can be listed. For the polynomial problem, this column can include the complexity of the algorithm as well as a brief description of, or an abbreviated name for, the algorithm. Common algorithms are **SPT (shortest processing time)** and **EDD (earliest due date)**. The following examples may serve to clarify the notation in this column:

NP	The problem is NP-hard.
P	The problem is polynomially solvable.
$O(n^2)$	The complexity of the algorithm is proportional to the square of the number of jobs n.
$O(nlogn)$	The complexity of the algorithm is proportional to n times the log of n.
NP(2)	The problem is NP-hard, and the heuristic has a relative error bound of 2.
EDD	The algorithm used to solve the problem optimally is the earliest due date procedure.
ERD	The algorithm used to solve the problem optimally is the earliest release date procedure.
WSPT	Weighted shortest processing time: namely, sequence jobs by ratio of processing time to weight, smallest first.
DP	Problem solved with dynamic program.
LP	Problem solved with linear program.
LB	Algorithm finds lower bound on solution.

3.3 Column three: Reference

The last column of the table lists the papers that address the problem or establish the results listed in column two. Refer to the bibliography for a complete reference.

Table 1. The Single Machine Deterministic Scheduling Problem

Problem description	Complexity	Reference
MINIMAX CRITERIA		
$1/prec/f_{max}$	$O(n^2)$	Lawler(1973)
$1/pmtn, r_j, prec/f_{max}$	$O(n^2)$	Baker et al. (1983)
DELIVERY TIME MODEL: MINIMIZE MAXIMUM LATENESS; RELEASE DATES		
$1//L_{max}$	$O(n \log n)$ EDD	Jackson (1955)
$1//L_{max} - L_{min}$	branch-and-bound	Gupta and Sen (1984b)
	DP $O(n \log n\ MS)$	Liao and Huang (1991)
$1/r_j/L_{max}$	NP(2) extended Jackson	Simons (1978)
$1/r_j/L_{max}$	NP(3/2)	Potts (1980b)
$1/r_j/L_{max}$	NP(4/3)	Hall and Shmoys (1988)
$1/r_j/L_{max}$	NP	Lenstra, Rinnooy Kan, Brucker (1977)
$1/r_j, d_j = d/L_{max}$	$O(n \log n)$ ERD	Jackson (1955)
$1/r_j, p_j = p/L_{max}$	P	Simons (1978)
$1/r_j$, prec$/L_{max}$	NP (elegant enumeration)	Simons (1978)
FLOWTIME PROBLEMS		
$1//\Sigma w_j C_j$	WSPT	Smith (1956)
$1//\Sigma C_j$	SPT	Smith (1956)
$1/prec, p_j = 1/\Sigma C_j$	NP	Lawler (1978)
		Lenstra, Rinnooy Kan (1978)
$1/r_j/\Sigma C_j$	NP	Lenstra, Rinnooy Kan, Brucker (1977)
	asymptotic algorithms	Gazmuri (1985)
	heuristics	Liu and MacCarthy (1991)
$1/r_j$, pmtn$/\Sigma C_j$	P	Baker (1974)
$1/r_j$, pmtn$/\Sigma w_j C_j$	NP	Labetoulle et al. (1984)
$1/prec/\Sigma w_j C_j$	polynomial decomposition	Sidney and Steiner (1986)
$1/D_j/\Sigma C_j$	P extension of Smith (1956)	Lenstra, Rinnooy Kan, Brucker (1977)

FLOWTIME PROBLEMS, cont.

$1/D_j/\Sigma w_j C_j$	NP	Lenstra, Rinnooy Kan, Brucker (1977)
$1/r_j$, pmtn, $D_j/\Sigma C_j$	NP	Du and Leung (1988b)
$1/D_j/\Sigma w_j C_j$	NP	Potts and Van Wassenhove (1983)
$1/r_j/\Sigma w_j C_j$	NP	Hariri and Potts (1983)
$1/r_j$, clustered jobs $/\Sigma w_j C_j$	P	Posner (1986)
$1/\text{prec}/\Sigma w_j C_j$	branch-and-bound	Potts (1985b)
1/preventive maintenance/ΣC_j	NP	Rinnooy Kan
	NP(2/7)	Lee and Liman

NUMBER OF TARDY JOBS

$1//\Sigma w_j U_j$	NP	Karp (1972)
$1//\Sigma w_j U_j$	NP	Lawler and Moore (1969)
$1//\Sigma U_j$	P	Moore (1968)
$1/D_j, D_j \geq d_j/\Sigma U_j$	NP	Lawler (1982b)
$1/p_j < p_k$ implies $w_j \geq w_k/\Sigma w_j U_j$	$O(n \log n)$	Lawler (1976b)
$1/r_j/\Sigma U_j$	NP	Lawler (1982b)
	dominance properties	Erschler et al. (1983)
$1/r_j$, pmtn$/\Sigma U_j$	$O(n^5)$	Lawler (1990)
$1/r_j$, pmtn$/\Sigma w_j U_j$	$O(n^3(\Sigma w_j)^2)$	Lawler (1990)
$1/r_j, r_j < r_k$ implies $d_j \leq d_k/\Sigma U_j$	$O(n^2)$	Kise, Ibaraki, Mine (1978)
	$O(n \log n)$	Lawler (1982b)
1/pmtn, $r_j, (r_j, d_j)$ nested$/\Sigma w_j U_j$	$O(n \log n)$	Lawler (1982b)
1/pmtn, $r_j, r_j < r_k$ implies $p_j \leq p_k$ and $w_j \geq w_k/\Sigma w_j U_j$	$O(n \log n)$	Lawler (1982b)
$1/p_j = 1/\Sigma U_j$	$O(n)$	Monma (1982)
$1/p_j = 1$, prec$/\Sigma U_j$	NP	Garey and Johnson (1976)
$1//\Sigma w_j U_j$	NP, branch and bound	Villarreal and Bulfin (1983)
	NP: LB in $O(n \log n)$	Potts and Van Wassenhove (1988)
	DP: $O(n\Sigma w_j)$	Sahni (1976)

OTHER SUMS

$1/p_j = 1/\Sigma f_j$	$O(n^3)$ weighted bipartite matching	Lawler et al. (1989)
$1//\Sigma f_j$	open: $O(n^3)$ for LB	Rinnooy Kan, Lageweg, Lenstra (1975)
$1/\text{prec}/\Sigma f_j$	DP	Steiner (1984)
$1//\Sigma T_j$	pseudopolynomial algorithm $O(n^4\ MS)$	Lawler (1977)
	polynomial approximation	Lawler (1982c)
$1//\Sigma T_j$	NP	Du and Leung (1989b)
	adjacent pairwise interchange	Fry et al. (1989)
	branch-and-bound	Potts and Van Wassenhove (1985)
	branch-and-bound	Sen and Borah (1991)
$1/\text{chain}, p_j = 1/\Sigma T_j$	NP	Leung and Young (1990)
$1/\text{prec}, p_j = 1/\Sigma T_j$	NP	Lenstra and Rinnooy Kan (1978)
$1/\text{pmtn} /\Sigma\ min\ \{T_j, p_j\}$	$O(n \log n)$	Potts and Van Wassenhove (1992)
$1//\Sigma\ min\ \{T_j, p_j\}$	NP	Potts and Van Wassenhove (1992)
$1//\Sigma w_j T_j$	NP	Lawler (1977)
		Lenstra, Rinnooy Kan, Brucker (1977)
	branch-and-bound	Potts and Van Wassenhove (1985)
	local precedence relationships	Rachamadugu (1987)
	analysis of local searches	Chang et al. (1990)
	heuristic decomposition	Chambers et al. (1991)
$1/r_j, p_j = 1/\Sigma w_j T_j$	$O(n^3)$ weighted bipartite matching	Lawler et al. (1989)
$1/\Sigma U_j = n/\Sigma w_j T_j$	WSPT	McNaughton (1959)
$1//\Sigma w_j C_j^2$	branch-and-bound	Townsend (1978)
	dominance properties	Gupta and Sen (1984a)
		Szwarc et al. (1988)
$1//\Sigma (C_j - \overline{C})^2$	heuristic	Vani and Ragchavachari (1987)
$1//\Sigma (C_j - \overline{C})^2$	DP $O(n^2\ MS)$ polynomial approximation	De et al. (1992)

MULTIPLE RELEASE-DEADLINE INTERVALS

$1/p_j = p/\Sigma U_j$	NP	Simons and Sipser (1984)
$1/p_j = 1/\Sigma U_j$	$O(n^3)$	Simons and Sipser (1984)
$1/p_j = 1/\Sigma C_j$	$O(n^4)$	Simons and Sipser (1984)
$1/p_j = 1/C_{max}$	$O(n^4)$	Simons and Sipser (1984)

EARLINESS-TARDINESS PROBLEMS
SINGLE-MACHINE COMMON DUE DATE PROBLEMS

$1/d_j = d/\Sigma(E_j + T_j)$	unrestricted d, $O(n \log n)$	Kanet (1981)
	restricted, NP	Hall, Kubiak, Sethi (1991)
	heuristic (1.5)	Liman and Lee (1991)
	branch and bound	Szwarc (1989)
$1/d_j = d/\Sigma(\alpha E_j + \beta T_j)$	unrestricted d, $O(n \log n)$	Bagchi, Chang, Sullivan (1987)
$1/d_j = d/\Sigma(E_j^2 + T_j^2)$	interchange heuristic	Eilon and Chowdhury (1977)
$1/d_j = d/\Sigma(\alpha E_j^2 + \beta T_j^2)$	enumerative search	Bagchi, Chang, and Sullivan (1987)
$1/d_j = d, \alpha_j = \beta_j$ $/\Sigma(\alpha_j E_j + \beta_j T_j)$	NP	Hall and Posner (1991)
$1/d_j = d, \alpha_j = \alpha p_j$ $/\Sigma(\alpha_j E_j + \beta_j T_j)$		Bagchi (1985)
$1/d_j = d/\Sigma(F(E_j) + G(T_j))$	$O(n\ MS)$	Kahlbacher (1993)
$1/d_j = d/\Sigma(F(E_j) + G(T_j))$	$O(n^2 p_{max}^2)$	Federgruen and Mosheiov (1991)

DIFFERENT DUE DATES

$1//\Sigma(\alpha_j E_j + \beta_j T_j)$	NP	Garey, Tarjan, Wilfong (1988)
	Interchange heuristic	Fry, Armstrong, Blackstone (1987)
	Filtered beam search	Ow and Morton (1989)
$1//\Sigma(\alpha_j E_j^2 + \beta_j T_j^2)$	branch-and-bound	Gupta and Sen (1983)

DUE DATE WINDOW ($E_j = max\ \{0, e_j - C_j\}, T_j = max\ \{0, C_j - d_j\}$)

$1/C_j \leq d_j/E_{max}$	allowable idle time, P no idle time, $O(n^2)$	Lee (1991)
$1//max \{E_{max}, T_{max}\}$	bisection search	Lee (1991)
$1/d_j - e_j = K$, agreeable α_j, β_j $/\Sigma (\alpha_j E_j + \beta_j T_j)$	Pseudopolynomial	Kraemer and Lee (1992)
$1/d_j - e_j = K$, agreeable α_j, β_j $/\Sigma (\alpha E_j + \beta T_j)$	$O(n \log n)$	Kraemer and Lee (1992)

DELIVERY COSTS

$1/d_j = d$, agreeable α_j, β_j $/\Sigma (\alpha_j E_j + \beta_j T_j + \gamma_j U_j)$	$O(n\ MS + n^2 d)$	Lee, Danusaputro, Lin (1991)
$1/d_j = d/\Sigma (\alpha E_j + \beta T_j + \gamma U_j)$	NP, $O(n(d + p_{max}))$	Lee, Danusaputro, Lin (1991)

BATCH DELIVERIES

$1/\beta_j = 0/\Sigma\, \alpha_j E_j + f(Q)$	NP	Cheng and Kahlbacher
$1/\beta_j = 0/\Sigma\, \alpha E_j + f(Q)$	$O(n^2 \log n)$	Cheng and Kahlbacher
$1/d_j = 0$, two fixed deliveries$/\Sigma (\alpha E_j + \beta T_j)$	unrestricted, $O(n)$ restricted, NP	Chhajed (1991)
$1/d_j = d, \alpha \leq \beta$ $/\Sigma (\alpha E_j + \beta T_j) + KQ$	$O(n^2 MSd)$	Herrmann and Lee (1991)
$1/d_j = d, \alpha \leq \beta, p_j = p/\Sigma (\alpha E_j + \beta T_j) + KQ$	$O(n^2)$	Herrmann and Lee (1991)
$1/d_j = d, \alpha_j = 0$ $/\Sigma\, \beta T_j + KQ$	$O(n^2)$	Herrmann and Lee (1991)

ADDITIONAL PENALTIES, d A DECISION VARIABLE

$1/d_j = d/\Sigma(\alpha E_j + \beta T_j + \gamma(d - d_0)^+)$	enumeration	Panwalkar, Smith, Seidmann (1982)
$1//\Sigma(\alpha E_j + \beta T_j + \gamma(d - d_0)^+)$	SPT	Panwalkar, Smith, Seidmann (1982)
$1/d_j = d/\Sigma(\alpha E_j + \beta T_j + \gamma d)$	$O(n \log n)$	Panwalkar, Smith, Seidmann (1982)
$1/d_j = d/\Sigma(\alpha_j E_j + \beta_j T_j + \gamma d)$	V-shaped schedules optimal	Bector et al. (1989)
$1/d_j = d/\Sigma(\alpha E_j + \beta T_j + \theta C_j)$	$O(n \log n)$	Baker and Scudder (1990)

JOB DEADLINES

$1/D_j/\Sigma E_j$	NP, branch-and-bound	Ahmadi and Bagchi (1986)
1/order j, job ij, D_j $/\Sigma_j n_j\theta_j C_j + \Sigma_j \Sigma_i (\alpha_j E_{ij} + \beta_j T_{ij}$		Bagchi (1989)

Table 2. Dual-Criteria Deterministic Scheduling Problems

Problem description	Complexity	Reference
DEADLINES		
First, note that the following constraints are equivalent: $\Sigma U_j = 0$, $C_j \leq d_j$, D_j, $T_{max} = 0$.		
$1/C_j \leq d_j/\Sigma C_j$	$O(n \log n)$	Smith (1956)
$1/C_j \leq d_j + T_{max} /\Sigma C_j$	$O(n \log n)$	Heck and Roberts (1972)
$1/C_j \leq d_j/\Sigma w_j C_j$	counter-example to Smith's algorithm	Burns (1976)
	NP	Lenstra, Rinnooy Kan, Brucker (1977)
	branch-and-bound	Bansal (1980)
	pivoting heuristic $O(n^3)$	Miyazaki (1981)
	branch-and-bound, decomposition approach	Shanthikumar and Buzacott (1982)
	branch-and-bound	Potts and Van Wassenhove (1983)
	better branch-and-bound	Posner (1985)
	$p_i > p_j$ implies $w_i \leq w_j$ Smith's algorithm	Chand and Schneeberger (1986)
	w_j a convex (concave) function of p_j: Smith's algorithm	Chand and Schneeberger (1986)
	improved lower bound	Bagchi and Ahmadi (1987)
$1/C_j \leq d_j/\Sigma w_j E_j$	NP, dynamic goal programming, special cases	Chand and Schneeberger (1988)
	branch-and-bound	Ahmadi and Bagchi (1986)
MINIMAL NUMBER OF TARDY JOBS		
$1/\Sigma U_j = U_{min}/\Sigma C_j$	branch-and-bound	Emmons (1975)
$1/\Sigma U_j = U_{min}/T_{max}$	branch-and-bound	Shanthikumar (1983)
$1/\Sigma U_j = U_{min}/\Sigma T_j$	branch-and-bound	Vairaktarakis and Lee (1992)

MULTIPLE MACHINE MODELS

$P2/p_j = 1$, prec, $C_j \leq d_j/C_{max}$	$O(n^2 \log n)$	Garey and Johnson (1976)
$F/r_j, C_j \leq d_j$ $/\Sigma\, w_{ij} I_{ij}$	NP, branch-and-bound	Ahmadi and Bagchi (1992)
$F2/C_j \leq d_j/\Sigma\, I_{ij}$	NP, branch-and-bound	Ahmadi and Bagchi (1992)

MULTICRITERIA OBJECTIVE FUNCTIONS

$1//\Sigma\, C_j, T_{max}$	$O(MSn \log n)$	Van Wassenhove and Gelders (1980)
$1/a_j \leq p_j \leq b_j/T_{max}$, $\Sigma\, w_j(b_j - p_j)$	$O(n^2)$	Van Wassenhove and Baker (1982)
$1/a_j \leq p_j \leq b_j/max$ $g_j(C_j)$, $\Sigma\, w_j(b_j - p_j)$	$O(n^3)$	Van Wassenhove and Baker (1982)
$1//\Sigma\, C_j$, T_{max}, $\Sigma\, U_j$	branch-and-bound, heuristics	Nelson et al. (1986)
$1//\Sigma\, C_j$, $\Sigma\, U_j$	lower bound	Kiran and Unal (1991)
$1//\Sigma\, C_j$, $\Sigma\, T_j$, $\Sigma\, U_j$	branch-and-bound	Kao (1980)
$1//\Sigma\, C_j$, $\Sigma\Sigma\, \lvert C_j - C_i \rvert$	P	Bagchi (1989)
$1//\Sigma\, (C_j - \overline{C})^2, \overline{C}$	DP $O(n^2 MS)$	De et al. (1992)
$F//C_{max}$, T_{max}	branch-and-bound heuristics	Daniels and Chambers (1990)

Table 3. Parallel Machine Deterministic Scheduling Problems

Problem description	Complexity	Reference
MINIMUM SUM PROBLEMS FLOWTIME		
$R//\Sigma C_j$	$O(n^3)$	Horn (1973)
	Weighted bipartite matching	Bruno, Coffman, Sethi (1974)
$P//\Sigma C_j$	$O(n \log n)$	Conway, Maxwell, Miller (1967)
$Q//\Sigma C_j$	$O(n \log n)$	Horowitz and Sahni (1976)
UNIFORM MACHINES, UNIT PROCESSING TIMES		
$Q/p_j = 1/\Sigma f_j$	$O(n^3)$	Dessoukey et al. (1989)
$Q/p_j = 1/\Sigma w_j C_j$	$O(n \log n)$	Dessoukey et al. (1989)
$Q/p_j = 1/\Sigma T_j$	$O(n \log n)$	Dessoukey et al. (1989)
$Q/p_j = 1/\Sigma w_j U_j$	$O(n \log n)$	Dessoukey et al. (1989)
$P/p_j = 1/\Sigma U_j$	$O(n \log n)$	Lawler (1976a)
$Q/p_j = 1/f_{max}$	$O(n^2)$	Dessoukey et al. (1989)
$Q/p_j = 1/L_{max}$	$O(n \log n)$	Dessoukey et al. (1989)
$Q/p_j = 1/C_{max}$	$O(n \log n)$	Dessoukey et al. (1989)
IDENTICAL MACHINES		
$P2//\Sigma C_j$	$O(n)$	Assad (1985)
$P2/p_j = 1/\Sigma w_j C_j$	$O(n)$	Assad (1985)
P/increasing processing rate/ΣC_j	SPT	Meilijson and Tamir (1984)
P/r_j, increasing processing rate/ΣC_j	SPT	Huang (1986)
P/variable processing rate/C_{max}		Dror et al. (1987)
P/variable processing rate/ΣC_j	SPT on one machine	Dror et al. (1987)
$P2//\Sigma w_j C_j$	NP	Bruno, Coffman, Sethi (1974)
P2/tree/ΣC_j	NP	Sethi (1977)
P2/machine 2 has limited available time /ΣC_j	NP	Lee and Liman (1992)

IDENTICAL MACHINES, cont.

$P//\Sigma w_jC_j$	DP	Lawler and Moore (1969)
	DP	Lee and Uzsoy (1992)
	$NP((\sqrt{2}+1)/2)$	Kawaguchi and Kyan (1986)
$P//\Sigma T_j$	heuristic	Ho and Chang (1991)
$P/w_j = p_j/\Sigma w_jT_j$	NP	Arkin and Roundy (1991)
	$O(n^2MS)$	

IDENTICAL MACHINES, PREEMPTION

$P/\text{pmtn}/\Sigma C_j$	$O(n \log n)$	McNaughton (1959)
$P/\text{pmtn}/\Sigma w_jC_j$	NP	Du, Leung, Young (1989)
$P2/\text{pmtn, tree}/\Sigma C_j$	NP	Du, Leung, Young (1989)
$P2/\text{pmtn}, r_j/\Sigma C_j$	NP	Du, Leung, Young (1988)
$P2/\text{pmtn}, p_j = p, r_j$ $/\Sigma C_j$	$O(n \log n)$	Herrbach (1990)
$P2/\text{pmtn}, r_j/\Sigma U_j$	NP	Du, Leung, Wong (1989)
$P/\text{pmtn}/\Sigma U_j$	NP	Lawler (1983)

OTHER PREEMPTIVE MACHINES

$Q/\text{pmtn}/\Sigma C_j$	$O(n \log n + nm)$	Gonzalez (1977)
$Q2/\text{pmtn}/\Sigma U_j$	$O(n^3)$	Lawler and Martel (1989)
$Q2/\text{pmtn}/\Sigma w_jU_j$	$O(n^2 \Sigma w_j)$	Lawler and Martel (1989)
	polynomial approximation	
$Q/r_j, \text{pmtn}/\Sigma U_j$	$O(mn^3)$	Federgruen and Groenvelt (1986)
$Q/r_j, \text{pmtn}/L_{max}$	$O(mn^3 \log(np_{max}s_{max}))$	Federgruen and Groenvelt (1986)
$Q/r_j, \text{pmtn}/\Sigma w_jC_j$	$O(mn^3p_{max})$	Federgruen and Groenvelt (1986)
$R/r_j, \text{pmtn}/\Sigma U_j$	NP	Du and Leung (1991)

MINIMAX CRITERIA
NON-PREEMPTIVE PROBLEMS

$P//C_{max}$	NP	Garey and Johnson (1978)
	NP$(2 - 1/(2m))$	Graham (1966)
	NP$(4/3 - 1/(3m))$	Graham (1969)
	NP(1.22)	Coffman et al. (1978)
	NP(6/5)	Friesen and Langston (1986)
	NP(10/9 for 2 machines)	Lee and Massey(1988)
$P/\|\{p_j\}\| = k/C_{max}$	pseudopolynomial	Leung (1982)
P/communication delay/C_{max}	Error bound	Lee et al. (1988)
P/M_i not immediately available/C_{max}	NP (1.5 for LPT)	Lee (1991)
$P/r_j/L_{max}$	EDD $(((2m-1)/m)p_{max})$	Gusfield (1984)
$P/r_j, p_j = p/L_{max}$	$O(mn^2)$	Simons and Warmuth (1989)
$R//C_{max}$	NP	Lenstra, Shmoys, Tardos (1989)
	List scheduling	Davis and Jaffe (1981)
	NP(2)	Potts (1985a)

PREEMPTION, MAKESPAN

P/pmtn/C_{max}	$O(n)$	McNaughton (1959)
Q/pmtn/C_{max}	$O(mn^2)$	Horvath, Lam, Sethi (1977)
	$O(n + m\log m)$	Gonzalez and Sahni (1978b)
R/pmtn/C_{max}	LP	Lawler and Labetoulle (1978)
P/pmtn,r_j/C_{max}	$O(n^2)$	Horn (1974)
	$O(mn)$	Gonzalez and Johnson (1980)
Q/pmtn,r_j/C_{max}	$O(n\log n + mn)$	Labetoulle et al. (1984)

PREEMPTION, MAXIMUM LATENESS

P/pmtn/L_{max}	$O(n^2)$	Horn (1974)
	$O(mn)$	Gonzalez and Johnson (1980)