2011

Order Acceptance and Scheduling: A Taxonomy and Review

Susan A. Slotnick
Cleveland State University, S.SLOTNICK@csuohio.edu

Follow this and additional works at: http://engagedscholarship.csuohio.edu/bus_facpub

Part of the Business Administration, Management, and Operations Commons

Publisher's Statement
NOTICE: this is the author's version of a work that was accepted for publication in European Journal of Operational Research. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in European Journal of Operational Research, 212 (2011), 10.1016/j.ejor.2010.09.042

Original Published Citation

This Article is brought to you for free and open access by the Monte Ahuja College of Business at EngagedScholarship@CSU. It has been accepted for inclusion in Business Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.
Order acceptance and scheduling: A taxonomy and review

Susan A. Slotnick*

Department of Operations and Supply Chain Management, Cleveland State University, 2121 Euclid Avenue, Cleveland, OH 44115, USA

1. Introduction

Over the past 20 years, the topic of order acceptance has attracted considerable attention from those who study scheduling and those who practice it. In a firm that strives to align its functions so that profit is maximized, the coordination of capacity with demand may require that business sometimes be turned away. Which orders are accepted and which are rejected may depend on the strategic direction of the firm, the current status of capacity already allocated, and the profitability of the order in question. In particular, there is a trade-off between the revenue brought in by a particular order, and all of its associated costs of processing, which may include delay costs for other orders, as well as any penalties incurred if this order is delivered after its agreed-upon due date.

When cost of capacity and per-order revenue must be reconciled, there are a number of ways that firms respond. They may expand capacity by permanent or temporary means, the latter including running extra shifts, diverting production resources, or subcontracting. Negotiation or renegotiation of delivery dates and pricing are other options. Related research areas include capacity rationing, lead-time estimation, revenue management, and due-date setting. The present paper focuses on research that approaches this reconciliation by considering two decisions: which orders to accept and how to schedule them.

This paper provides a review and a taxonomy of the literature on order acceptance and scheduling from a problem-oriented perspective. Diverse methodologies have been applied to this problem, including (but not limited to) mathematical programming, metaheuristics, queuing theory, simulation, algorithm development and decision analysis. Literature reviews in research papers are typically limited to the methodological area that is the topic of the analysis in that work. For example, a paper using integer programming is not likely to include a work on algorithm development in its literature review, and vice versa. So a major contribution of this paper is to facilitate future work on this topic, by providing a map of what has been done and how. Bringing the diverse streams of research together in one discussion will also lay the path for integrative studies that build on ideas and methodologies from various disciplines.

The most complete review to date of the research on this topic is part of a chapter on due-date management policies by Keskinocak and Tayur in the Handbook of Quantitative Supply Chain Analysis (Keskinocak and Tayur, 2004). Section 6 of that chapter discusses order acceptance in the context of due-date management, including pricing decisions. The current paper builds on this basis by developing a detailed taxonomy of the problem and extending the scope of inquiry to a diverse set of methodologies.

This paper is organized as follows. Section 2 defines the problem, briefly discusses related topics, and presents prevalent themes and the taxonomic structure that informs the rest of the analysis. Section 3 uses the taxonomy to frame the discussion of the literature, while relating individual studies to themes that cut across problem categories. The outlook for future research appears in Section 4.

2. Problem definition and taxonomy

2.1. Problem definition and scope

The problem of order acceptance and scheduling (OAS) is defined as the joint decision of which orders to accept for processing
and how to schedule them. The decision-maker is faced with a collection or stream of orders whose combined processing requirements would exceed available capacity, and has the option of rejecting some of those orders. If all orders must be accepted, the problem reduces to the scheduling decision. If no scheduling is required, the problem is analogous to the knapsack problem (deterministic or stochastic), or admission control in queueing theory (stochastic). Because the research reviewed here is motivated, not by an underlying mathematical model, but by a real-world decision problem, the diversity of the approaches (objective functions and methods) precludes one general, formal definition. Generic forms for the most common models are presented in Section 3.2.1 and referred to as appropriate in subsequent sections.

Interested readers will find a guide to related research areas that are outside the scope of this survey in Table 1. This table also includes selected references to three specific topics in scheduling research that are relevant to this paper: minimizing the (weighted) number of late jobs, interval scheduling and due-window scheduling (see Table 2).

### 2.2. Taxonomy

References to the papers discussed here were found by searching standard bibliographic databases, augmented by an examination of the cited references in each paper for research that incorporates order acceptance as well as scheduling. Table 2 displays the five major categories of papers covered in this review, and their salient characteristics. The category of deadlines includes models that do not allow lateness. Adjustable processing times refer to processing times that are compressible (at a cost) as well as deteriorating processing times.

### 2.3. Research themes (Table 3)

Research on OAS is diverse in terms of objectives, solution methods, and problem characteristics. Within this diversity, there are related streams of research, and themes that connect papers from different disciplines and methodologies. Table 3 shows the major themes of OAS research in the past 20 years. Note that some papers are listed more than once, since they fit the criteria for multiple categories.

The first four themes refer to problem characteristics. The fourth and fifth themes, coordinating decisions in a hierarchical structure and use of myopic or greedy methods, includes deterministic and stochastic models, with single and multiple machines. The last two themes are important for the choice of solution methods, as well as for the implementation of decision models in practical situations. Authors using different objectives and methods find that it is more effective to combine the selection and scheduling decisions. The last theme evaluates the trade-off between the cost of information for OAS, and the benefit provided by acquiring that information.

The next section presents the details of the papers included in each category. The discussion is organized around the taxonomy (Table 2), and relates papers, when relevant, to the prevalent themes in Table 3.

### 3. Review of the OAS literature

#### 3.1. Conceptual papers

**Guerrero and Kern (1988)** provide a rationale for rejecting some orders, develop a framework and provide an example for making that decision in the context of the Master Production Schedule (MPS) and Final Assembly Schedule (FAS) for manufacturing. Kern and Guerrero (1990) formulate a mixed-integer linear programming model that minimizes costs of penalties for late or failed deliveries, inventory and setup (order) cost.

**Alarcoi et al. (2009)** develop a conceptual framework for order promising, which includes acceptance or rejection of orders. Focusing on the specific context of collaborative selling networks, the authors categorize previous research with regard to how specific
papers have helped to define various aspects of the order promising problem, and present a conceptual framework to guide the design of order promising methods.

As mentioned above, the recent survey by Keskinocak and Tayur (2004) covers various aspects of due-date management, including a section on “Due-Date Management with Price and Order Decisions.” An introductory subsection puts this topic in the perspective of the due-date management and lead-time setting literature, and their next two subsections discuss in detail those papers for which due-dates or quoted lead times, sometimes in conjunction with pricing decisions, have an effect on demand or customer orders.

3.2. The deterministic single-machine problem (Table 4)

3.2.1. Overview and generic problems

For revenue or profit maximization, the generic problem for deterministic single-machine OAS can be formulated as:
The processing order of greedy algorithms, and develop a generalization of the orders one at a time and inspecting the effect of each addition on the order of the problem, with multiple resource constraints and any number of projects is to be chosen. Aspvall et al. (1995) present an optimal dynamic programming (DP) procedure for the unconstrained problem when a time of order is . Gupta et al. (1992) present an optimal dynamic procedure. An insight of the paper is that for this type of problem, if optimal scheduling is not straightforward (e.g., WSPT for weighted lateness, Rothkopf sequence for NPV), then procedures that jointly select and schedule produce higher quality solutions than those that use separate heuristics for those two decisions (see Table 3). Rom and Slotnick (2009) develop a genetic algorithm for OAS with weighted tardiness that performs better than previous heuristics in terms of solution quality, though its running time is longer.

Nobibon and Leus (2011) consider order selection when there are “firm planned orders as well as potential orders.” After providing complexity (non-approximability) results, the authors present two MILP procedures and two B&B algorithms, which include features from Slotnick and Morton (2007) and Rom and Slotnick (2009). A computational study compares the performance of the procedures under various scenarios.

Oğuz et al. (2010) include sequence-dependent setup times in a model that maximizes profit (revenues of accepted orders minus total weighted tardiness penalties) and a two-level due-date structure: a preferred due-date , after which a tardiness penalty is incurred, until the strict deadline , after which the customer will not take the order. This work adds the strict deadline and sequence-dependent setup times to objective (1) and the problem studied by Slotnick and Morton (2007). The problem is formulated as an MILP and solved optimally for ten- and some fifteen-order problems. Larger problems are solved heuristically by Simulated Annealing (SA) and two constructive heuristics. A computational study shows that SA yields an average 9% deviation and a maximum 21% deviation compared to the LP bound for up to 50-order problems. Running time of the SA procedure increases as problem size grows, but two constructive heuristics for 100- and 300-order problems perform well.

\[
\max \sum_{j=1}^{n} x_j [Q_j - K_j], \quad \text{for weighted lateness and } K_j = w_j(C_j - d_j) \quad \text{for weighted tardiness, where } w_j \text{ is the customer weight (proportional lateness discount), } d_j \text{ is the due date, and } C_j \text{ is the completion time of order } j \text{ (i.e. } C_j = \sum_{k=x}^{n} p_k, \text{ where } p_k \text{ is the processing time of order } k). \text{ Abusing notation, } K \text{ without a subscript will be used to represent aggregate costs in the subsequent discussion.}
\]

\[
\min \sum_{j=1}^{n} (1 - x_j) R_j + x_j K_j \quad \text{where } R_j \text{ are costs of rejection. If rejection costs are exactly equal to lost revenue (which is sometimes but not always the case), then (2) can be rewritten as the following maximization problem:}
\]

\[
\max \sum_{j=1}^{n} \left[ Q_j - (1 - x_j) R_j - x_j K_j \right], \quad \text{for weighted lateness and } K_j = w_j(C_j - d_j) \quad \text{for weighted tardiness, where } w_j \text{ is the customer weight (proportional lateness discount), } d_j \text{ is the due date, and } C_j \text{ is the completion time of order } j \text{ (i.e. } C_j = \sum_{k=x}^{n} p_k, \text{ where } p_k \text{ is the processing time of order } k). \text{ Abusing notation, } K \text{ without a subscript will be used to represent aggregate costs in the subsequent discussion.}
\]

\subsection{3.2.2. Maximizing revenue}

A case study of a textile mill (Stern and Avivi, 1990) models a multiple-machine environment as a single machine by allowing preemption and concurrent processing. The model does not permit late orders, and so scheduling is from earliest to latest due-date (EDD) which minimizes maximum tardiness (Jackson, 1955). A 0–1 integer program subject to a nonlateness constraint is analogous to (1), where \( K = 0 \). The authors present an optimal procedure and two heuristics that employ greedy/myopic methods.

Research on OAS and project selection, where the objective is to maximize the Net Present Value (NPV) of the total return on a single processor, use an objective analogous to (1), where \( K = 0 \) and \( Q \) is discounted. When the NPV is an exponential function of completion time, there is a known result to sequence the orders (Rothkopf, 1966; Rothkopf and Smith, 1984), i.e., nonincreasing order of the index \( NPV_i / [1 - (1 + r)^{-i}] \), where \( NPV_i \) is the Net Present Value of order \( j \), the discount factor is \( 1 + r \) and \( p_j > 0 \) is the processing time of order \( j \). Gupta et al. (1992) present an optimal dynamic programming (DP) procedure for the unconstrained problem when a specified number of projects is to be chosen. Aspvall et al. (1995) study the same problem, develop an analogous scheduling rule, and present an optimal method for selection and scheduling. Kyparisis et al. (1996) develop two heuristics for an extension of this problem, with multiple resource constraints and any number of orders.

A common feature of these algorithms is the idea of adding orders one at a time and inspecting the effect of each addition on the objective function value. That is, they are greedy/myopic procedures. Alidaee et al. (2001) study this problem in the general context of greedy algorithms, and develop a generalization of the dynamic programming algorithm for the earlier project selection problem (Gupta et al., 1992; Aspvall et al., 1995), where the objective function value depends on the completion time of each project.

\subsection{3.2.3. Profit objective with time-related penalties}

The above papers maximize NPV without explicitly considering costs. Variants of the single-machine problem with static arrivals, where the objective is the maximization of profit, arise from the use of various costs, including costs of lateness, tardiness, earliness, setup, compression, outsourcing, production and holding. Slotnick and Morton (1996) study OAS with the objective of maximizing profit, that is, revenue minus weighted lateness penalties (1). Because Weighted Shortest Processing Time order (WSPT) is the optimal sequence for weighted lateness/flowtime (Smith, 1956), scheduling is straightforward, and so optimal procedures and heuristics can be developed that are not burdened with a high computational effort of scheduling.

Their results include a myopic property of the problem, that is, the original set of orders can be partitioned into a subset of orders that are definitely included in the optimal solution, and a subset of those that may be rejected. Specifically, if the index \( \pi_i = \min_j \{ p_j, w_j(C_j - d_j) - Q_j \} \) is negative, order \( j \) belongs to the optimal solution. This property reduces the search space for a Branch-and-Bound method, and is the basis of high-quality and high-speed heuristics, including a myopic procedure. Ghosh (1997) extends this work, showing that OAS with lateness penalty is NP-hard, and presents two pseudo-polynomial time algorithms (based on De et al. (1993), discussed in Section 3.3) that solve the problem optimally, and a Fully Polynomial-Time Approximation Scheme (FPTAS). Lewis and Slotnick (2002) employ the myopic property to extend the OAS model to multiple periods, using a dynamic programming approach for an optimal benchmark, and various heuristics for quicker solutions and larger problems. Their results suggest that incurring a cost of information, which may involve keeping historical records or accurate forecasting of future demand, is advantageous when customers and order types are heterogeneous (see Table 3).
Yang and Geunes (2003, 2007) extend methods used for the Throughput Maximization Problem (TMP) to OAS, maximizing profit while considering costs of tardiness, processing time compression and extension of the scheduling horizon. The objective function of their most general problem corresponds to (1) with costs including tardiness, compression, and extension. An MILP solves small problems; heuristics with separate selection and scheduling (priority dispatching) are developed and tested for larger complex problems, using as benchmark a heuristic with a proven worst-case ratio.

### 3.2.4. Lead-time and due-date setting models

Models that develop reliable lead-time quotation (that is, late deliveries not allowed) may include the option to reject orders, in order to manage capacity. Keskincak et al. (1997, 2001) present online and offline models for lead-time quotation in which orders may be rejected, customers will leave if lead-time is too long, and accepted orders must be delivered on time. The objective is to maximize revenue, which is a decreasing function of the lead-time quotation (analogous to (1), with appropriate changes to \( Q \) to reflect the relationship with lead time). Models are developed for scenarios with one or two customer types; online or offline; and immediate or delayed lead-time quotation. Results from the case with unit processing times lead to insights about the general problem, including bounds and competitive analysis (measuring the performance of an online algorithm by comparing it with an optimal offline algorithm, modeled as an MIP).

Charnsiriaskkul et al. (2004) develop an MIP formulation and use numerical analysis to examine simultaneous order acceptance, scheduling and due-date setting decisions for a manufacturer that can choose lead-times and reject orders, with a two-level due-date structure, as in Öguz et al. (2010). Prices are exogenous, order pre-emption is allowed, demand is deterministic, and there are negligible setup costs. The objective function is analogous to (1), where \( K \) is expanded to include production and holding costs as well as tardiness penalties, and the decision variable \( x \) is defined as units of capacity per order. Computational studies compare the benefits of lead-time flexibility with the flexibility to deliver partial orders. Charnsiriaskkul et al. (2006) add pricing decisions, where order rejection may be caused by a price higher than the customer will accept. The objective function here includes price selection, and corresponding order quantity, in the revenue term. Numerical studies comparing two heuristics based on LP relaxation with an MIP lead to insights about the usefulness of pricing, inventory and lead-time flexibility in different environments.

### 3.2.5. Minimizing cost

All of the papers discussed above maximize revenue or profit, to reflect the disadvantage of rejecting orders. Another approach is to minimize costs, and include a cost of rejection, analogous to (2). Akkan (1997) minimizes the present value of lost revenue due to rejection, and inventory holding costs due to earliness. Arriving orders are inserted into an established schedule by heuristics that consider costs as well compaction and fragmentation of the schedule. A computational study demonstrates that including compaction improves the performance of a backward insertion heuristic; a heuristic that minimizes fragmentation when inserting a new order also performs well.

A problem from steel production motivates Chen et al. (2008), who use an objective function that minimizes cost, by including a “non-execution cost” when an order is rejected. The other costs are Early/Tardy and transition (setup) costs. The authors develop two heuristics: a GA and Extremal Optimization (EO), which eliminates and randomly replaces the worst components of a suboptimal solution. A hybrid of these two procedures dominates the genetic algorithm in a computational study.

Lee and Sung (2008) minimize a combination of completion time and outsourcing costs (i.e., rejection costs), with deteriorating (i.e. positionally dependent) processing times and an outsourcing cost constraint. They use Shortest Processing Time order (SPT) to schedule accepted orders, and addition/removal heuristics for accepting/rejecting. A computational study compares heuristics with an optimal procedure. Gordon and Strusevich (2009) develop a model with deteriorating processing times in which the decision variables include due dates and processing time. There are three decisions: accept/reject; assign due dates; schedule. Orders that cannot be completed by the due date are “discarded” by mutual agreement, with a penalty paid to the customer. Dynamic programming algorithms are developed for two variants of the model, using the two well-known due-date assignment methods CON and SLK (Baker and Bertrand, 1981).

### 3.3. The stochastic single-machine problem (Table 5)

Wester et al. (1992) employ simulation to study the relationship between order acceptance, production planning and scheduling, using four order selection approaches, while maximizing capacity utilization. Lateness is not allowed, there are multiple product types, and arrivals are stochastic. The authors find that the best approach is based on a knowledge of the current production schedule, with rescheduling when necessary, rather than estimates of capacity load or of the effect of accepting an order on the lateness of previously accepted orders. Adding selective order acceptance to a myopic procedure improves its performance. The authors analyze the value of detailed information that the firm should maintain for the purposes of order acceptance and scheduling, and the value of joint decision-making (see Table 3). Extensions of this model to multiple-machine scenarios are discussed in Section 3.4.

De et al. (1993) investigate order selection and scheduling for a single machine with random processing times and a random common due date. The authors develop optimal properties of scheduling, some of them myopic, which lead to dominance properties that accelerate computation. Pseudo-polynomial time exact and polynomial-time approximate algorithms are presented, as well as polynomial solutions for special cases, including the stochastic version of the problem studied by Gupta et al. (1992) and Aspvall et al. (1995).
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1973</td>
<td>Ono and Jones</td>
<td>Det</td>
<td>Max. profit</td>
<td>Simulation</td>
<td>Static</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1978</td>
<td>Jain et al.</td>
<td>Det</td>
<td>Max. throughput</td>
<td>LP, heuristic</td>
<td>Static</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>Pourbabi</td>
<td>Det</td>
<td>Max. profit</td>
<td>MILP</td>
<td>Static</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1992</td>
<td>Pourbabi</td>
<td>Det</td>
<td>Max. value</td>
<td>MILP, DP, heuristic</td>
<td>Static</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>Raaymakers</td>
<td>Det</td>
<td>Max. utility, service level</td>
<td>Workload, sched., makespan estimation</td>
<td>Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2000a</td>
<td>Raaymakers et al.</td>
<td>Det</td>
<td>Max. util., service level</td>
<td>Workload</td>
<td>Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>Snoek</td>
<td>Stoch</td>
<td>Max. profit</td>
<td>Neural net., Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>Ivalaescu et al.</td>
<td>Stoch</td>
<td>Max. service level, % accept, min. late</td>
<td>Workload, regression, scheduling</td>
<td>Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>Ivalaescu</td>
<td>Stoch</td>
<td>Max. % on time, util., min. tardy</td>
<td>Simulation, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>Nandi and Rogers</td>
<td>Stoch</td>
<td>Max. % accept, orders, cost</td>
<td>Simulation, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>Roundy et al.</td>
<td>Stoch</td>
<td>Max. % accept, orders, cost</td>
<td>Simulation, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>Moreira and Alves</td>
<td>Stoch</td>
<td>Min. cost</td>
<td>Simulation, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2006a</td>
<td>Ivalaescu et al.</td>
<td>Stoch</td>
<td>Max. % on time, utilization</td>
<td>Hybrid, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2006b</td>
<td>Ivalaescu et al.</td>
<td>Stoch</td>
<td>Max. % on time, utilization</td>
<td>Hybrid, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>Rogers and Nandi</td>
<td>Stoch</td>
<td>Max. profit</td>
<td>Simulation, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>Moreira and Alves</td>
<td>Stoch</td>
<td>Min. cost</td>
<td>Simulation, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>Mestry et al.</td>
<td>Det</td>
<td>Max. profit</td>
<td>Simulation, Dynamic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Stadje (1995) also develops optimal scheduling and selecting procedures for a stochastic version of the Gupta–Aspvall problem. The objective is to maximize total expected rewards by selecting and scheduling a fixed number of orders from an existing set. Processing times and the common due-date are random; initiation costs and termination rewards are deterministic. The machine is subject to random breakdowns (with probability of breakdown dependent on each order) that cause processing to terminate. Two optimal procedures which include myopic properties are presented for selection and one for scheduling.

Kate (1994, 1995) uses simulation studies to compare integrated and hierarchical approaches to OAS with random arrivals. When there are short lead-times or high utilization, a method based on the aggregate characteristics of orders already accepted is dominated by one that also includes production scheduling. Performance criteria include selectivity of orders, average lateness, fraction of tardy orders and average batch size. Kate (1995) develops MILP formulations, as well as heuristics, for the corresponding static Early/Tardy (E/T) scheduling problems. The integrated and hierarchical approaches perform better in computational studies than does a procedure that accepts orders randomly. As in previous papers (see Table 3), this author finds that it is worth using detailed information in certain circumstances, i.e., when capacity or lead-times are tight.

Dueñas and Hopp (1995) consider OAS in the context of setting due-dates when arrivals and processing times are stochastic and customer demand is sensitive to quoted lead times (rejection means setting a due-date beyond the customer’s tolerance). They develop models of optimal control-limit policies that maximize expected profit (revenue minus tardiness costs), including cases with infinite capacity and finite capacity with First-Come-First-Served order (FCFS), when lead-times are fixed by the market and when they are set by the firm. For the case when orders are not FCFS, they show that EDD is optimal with one customer class and linear or convex lateness penalties. Dueñas (1995) extends this work to multiple customer classes, developing heuristics for setting due dates. The results of a computational study suggest that information about customer preferences for lead times is advantageous to the decision making (see Table 3).

Carr and Dueñas (2000) consider two product classes, Make to Stock (MTS) and Make to Order (MTO), with random arrivals and processing times, preemption and no setups. The average profit per unit time is defined as the total revenue for both types of products, minus inventory costs for MTS and MTO, and penalties for shortages of MTS. There are two acceptance decisions: whether to accept an order for MTO, and what level of demand to satisfy for MTS. Optimal switching-curve policies for acceptance and scheduling are developed. A computational study investigates the performance of simpler policies in which the decisions are made jointly and separately; the latter is found to be inferior, as in other OAS papers (see Table 3).
3.4. The multiple-machine problem (Table 6)

3.4.1. Hierarchical production planning

A major stream of multiple-machine research on OAS builds on previous single-machine models (Wester et al., 1992; Kate, 1994, 1995) to focus on comparing different methods of production control, including order acceptance, in hierarchical production planning. These papers provide insights about the benefits of coordination and information-sharing between sales and operations for scheduling and order acceptance. In particular, they answer the question: when is a detailed scheduling method worthwhile, as opposed to a more general and less expensive aggregate approach? This theme occurs in other areas of the OAS literature as well (see Table 3).

Batch chemical manufacturing is the motivating example for Raaymakers (1999) and Raaymakers et al. (2000a,b), which compare the performance of three types of order acceptance and capacity loading policies: workload-based rules, detailed scheduling and makespan estimation using regression. The objective is to maximize capacity utilization with service level constraints, in a deterministic system. Workload rules provide feasible schedules but relatively low utilization (Raaymakers et al., 2000a), and a regression-based makespan estimation model dominates the workload rules (Raaymakers et al., 2000b). The estimation policy is better when there is high demand or high variety in product mix. Raaymakers (1999) compares these two methods with a detailed scheduling policy, which dominates when capacity requirements are high, but has the longest running time.

Also motivated by batch process industries, Iváñescu et al. (2002, 2006a,b) and Iváñescu (2004) extend the work of Raaymakers by considering uncertainty in arrivals and processing times. Iváñescu et al. (2002) compare three methods of order acceptance: a scheduling policy that constructs and evaluates a new schedule for each potential order, a workload policy that uses slack and processing uncertainty to construct aggregate workload profiles, and makespan estimation using multiple linear regression. Performance measures include utilization, service level (percentage of accepted orders completed before due dates), lateness of the order set, acceptance rate and feasibility performance. A simulation study demonstrates that when processing times are uncertain, the scheduling policy (which is the most time consuming) performs best. Of the two faster methods, the regression policy yields significantly better results than the workload policy. Iváñescu et al. (2006b) develop a hybrid policy using Simulated Annealing and regression, which dominates detailed scheduling and regression in terms of performance (percentage of accepted orders completed on time, and capacity utilization). Iváñescu et al. (2006a) find that bootstrapping is effective when there is limited data.

Ebben et al. (2005) contribute to this research stream by comparing order acceptance methods that consider precedence relationships, release dates and due dates of orders with those that only consider aggregate capacity restriction in an MTO shop. The objective is to maximize utilization with service-level requirements. The methods employed are aggregate resource loading (over all resources), resource loading per individual resource, a scheduling method based on EDD, and a Branch-and-Price (B&B) approach. Consistent with previous work, the authors find that when there is a high workload and little slack, the detailed scheduling method performs significantly better than the other procedures.

3.4.2. Other models with static arrivals

Ono and Jones (1973) investigate the effect of various policies for order acceptance, as they interact with dispatching and overtime policies, in a deterministic job shop with variable setup times and overtime costs. They apply a modification of the effective gradient method of Senju and Toyoda (1968), which accepts orders on the basis of capacity and due-date. Two heuristic rules determine whether to use overtime, and three scheduling rules include SPT, least slack and largest contribution to profit. The objective function maximizes profit, which is the revenue contributed by completed orders minus costs of production, tardiness and overtime, analogous to (1). Simulation studies suggest that performance is best when all factors are taken into account (see Table 3).

Jain et al. (1978) develop and implement a simulation model that uses linear programming and heuristic procedures to balance the order book and schedule orders for a steel manufacturer. Order acceptance is part of the order-book balancing procedure, where capacity and demand are coordinated. The objective function of the linear program is to maximize throughput (total tons of rolls produced during the planning period).

Pourbabi (1989, 1992) formulates an OAS model for a multiple-machine shop with setups, order splitting and product families. Scheduling is done by a dispatching rule that takes into account due-dates and order availability. The customer pays a higher price for a complete order that is delivered on time; partial orders may be rejected. The MILP maximizes profit, as in (1), including two different prices, and costs of production, tardiness and setups.

Kolisch (1998) uses a resource-constrained project scheduling approach for a set of problems composed of different tasks, with set due-dates and revenues. He formulates an MILP for a multi-period knapsack problem, with the objective of maximizing the value of accepted orders subject to precedence, resource and other standard constraints. For large problems, a heuristic based on linear programming performs fairly well compared to a B&B benchmark.

Roundy et al. (1999, 2005) use an MILP, LP relaxation and heuristics (GA, SA, tabu search, randomized local search and a “single machine heuristic” based on their MILP using network flows) to solve OAS with lot sizing in which incoming orders, if accepted, are inserted into the current schedule. The objective is to fill as many orders as possible while minimizing holding and setup costs; overtime is not allowed. The computational study shows that the GA, SA and single-machine heuristics perform best in terms of running time and solutions.

Mestry et al. (2009) add overtime and fixed due-dates (no lateness allowed) to the multiple-machine problem that maximizes profit of accepted orders (revenue minus manufacturing costs; cf. (1)). Arrivals are static in each period, but orders arriving in the next period may require rescheduling. An MILP is developed, which can only be solved for small problems. A B&P algorithm with Lagrangian bounds and an approximate branching scheme performs well for larger problems.

3.4.3. Other models with dynamic arrivals

Snoek (2000) uses a neuro-genetic network with reinforcement learning to combine order acceptance and scheduling decisions in a deterministic job shop with dynamic arrivals. The idea is to accept or reject an order on the basis of its potential contribution to discounted future rewards. In a simulation study, this procedure outperforms two slack-based heuristics, with the performance measure defined as the percentage of accepted orders.

Nandl and Rogers (2004) use simulation to develop an order-acceptance rule for a system with two product classes (regular and urgent), dynamic arrivals, and a composite performance measure that includes the ratio of actual to maximum possible revenue, the ratio of rejection losses to maximum possible revenue, and ratio of tardiness losses to maximum possible revenue. Orders are scheduled according to least slack per remaining operation (S/OPN). The acceptance mechanism uses a pair of look-ahead simulations, at the time of each order arrival, to compare the total contribution to profit with the current order accepted or rejected. The resulting optimal control policy is tested under different environ-
mental conditions of demand and order characteristics. The authors argue that this method of order acceptance has the potential to dominate other approaches because it uses full information about the status of the shop floor (see Table 3).

Rogers and Nandi (2007) also use simulation to study order acceptance, scheduling and order release in a fixed-capacity make-to-order system with two classes of orders, maximizing net profit (revenue minus tardiness as in (1)). Order acceptance is based on rules that consider the effect of the order on total shop load and load on the busiest machine. Scheduling is done by FCFS, EDD and/or S/OPN. Results of simulation studies show that selective order acceptance and immediate release is better than accepting all orders and holding them before release.

Moreira and Alves (2005, 2006, 2009) use simulation to compare policies for order acceptance, due-date setting, order release and scheduling in a job shop. Three order acceptance rules consider workload and due-date, with total acceptance as a benchmark. EDD and FCFS are compared for scheduling. Nine different performance measures minimize various aspects of time-related penalties and workload performance. Results show that performance improves when decisions are made simultaneously (see Table 3).

3.5. The order rejection problem (Table 7)

A series of related papers in the computer science literature treat OAS from the perspective of order rejection, rather than order acceptance. That is, the objective function includes minimization of total order rejection penalties, as well as other costs (such as makespan); this is analogous to (2). These papers provide complexity results and develop and analyze algorithms for both online and offline problems.

3.5.1. Multiple-machine models

Bartal et al. (2000) formulate the order rejection problem with identical parallel processors, no preemption, orders that are characterized by processing time and rejection penalties, and an objective that minimizes makespan and the sum of order rejection penalties. The solution approach invokes the trade-off between these two costs for order rejection, and uses list scheduling (SPT). Results include competitive algorithms and bounds for the online version, and an FPTAS and an approximation algorithm for the offline version.

He and Min (2000) extend these results, developing a deterministic algorithm that is optimal for two or three uniform machines that process at different speeds. Seiden (2001) extends the original problem by allowing preemption, and considers the case where the scheduling part of the algorithm does not "know" the rejection costs. The author develops a two-part algorithm, which adapts the rejection procedure of Bartal et al. (2000) and a preemptive online scheduling algorithm of Chen et al. (1995). The conjecture is made that in order to do better than the lower bound, an algorithm would have to integrate rejection and scheduling (see Table 3).

Hoogeveen et al. (2003) add preemption to the model with identical, related and unrelated parallel machines, and also consider an open shop with preemption and rejection. The problem with an arbitrary number of unrelated machines has a polynomial-time approximation algorithm, and an FPTAS is provided for the others.

Cao et al. (2006) extend this research by considering compressible processing times with three objectives: minimizing makespan with discretely compressible processing times and total compression cost as a constraint; minimizing total weighted completion time constrained by total penalty cost; and minimizing the sum of total weighted completion times plus total compression cost with discretely compressible processing times. Pseudo-polynomial time DP algorithms and FPTASs are developed for the first two problems, and a greedy heuristic with a worst-case performance ratio is presented for the third.

Dósa and He (2006a) study preemptive and non-preemptive versions for online scheduling on two uniform machines with rejection, to minimize the sum of makespan and rejection penalties. An optimal algorithm is presented for the preemptive version, and improved upper and lower bounds for the non-preemptive one. Dósa and He (2006b) add to this the possibility of purchasing new machines, with the purchase cost included in the objective function. An online algorithm is presented for the case where order cost is always less than machine cost.

Lu et al. (2008) define the problem of unbounded parallel batch machines with rejection and release dates, with the objective of minimizing the sum of makespan of accepted orders and total rejection penalties. They develop a pseudo-polynomial-time DP algorithm (which can be solved in polynomial time if there is a common rejection penalty), a 2-approximation algorithm and an FPTAS. Zhang et al. (2009b) present a pseudo-polynomial time
DP algorithm and an FPTAS for identical parallel machines and a constraint on total penalties.

3.5.2. Single-machine models

Engels et al. (2003) study the single-machine version of the order rejection problem where the objective function is the minimization of rejection costs and weighted completion time, that is, (2) with $K_i = w_i(C_i)$. For the offline version, they develop algorithms for the basic problem (pseudopolynomial-time DP) and variants (release dates, precedence, parallel machines). Special cases (identical weights, identical processing times) can be solved in polynomial time, and small-constant-factor approximation algorithms are provided for NP-hard variants, by reducing each of these to the corresponding model without rejection. This paper provides insights about how to transform scheduling problems with rejection to analogous problems without rejection.

Epstein et al. (2002) extend this work by developing an algorithm for the online version with unit processing times. Competitive analysis provides upper and lower bounds for the performance of the algorithm, which uses a greedy approach for the rejection decision. Sengupta (2003) presents a pseudo-polynomial algorithm, an FPTAS, and an ε-optimization approximation for the single-machine case with the objective of minimizing rejection penalties and maximum lateness or tardiness.

Zhang et al. (2009a) develop two pseudo-polynomial-time DP algorithms, a 2-approximation algorithm and an FPTAS for the single-machine order-rejection problem with release dates. Lu et al. (2009) also include release dates and introduce batch processing, developing exact algorithms for two special cases (identical and constant number of release dates), a 2-approximation algorithm and an FPTAS for the general problem. Cheng and Sun (2009) develop optimal DP algorithms and FPTASs, with polynomial-time algorithms for the problem with deteriorating processing times, focusing on three single-machine objectives: minimizing rejection penalties plus either maximum lateness/tardiness, total completion time or total weighted completion time.

4. Future research

Opportunities for future research on OAS include extensions of what has already been done, open questions presented by the authors of these papers, and the integration of common aspects of research across methodologies. For example, extensions of the deterministic single-machine problem already under way include the consideration of sequence-dependent set-up times (Oğuz et al., 2010). Other possible extensions include changing a given problem from single to multiple machines; including probabilistic demand, arrivals or processing times; and analyzing cases where customer demand is dependent on service or reputation (such as decreased demand relative to tardiness history).

Possible extensions of stochastic single-machine models might be to multiple machines or networks; more stochastic parameters; demand depending on lead times; pricing heuristics; general processing time distributions; and more complicated objectives. Extensions to multiple-machine models could involve bigger or more complex systems; interactions among decisions (order acceptance, pricing, scheduling, capacity loading); including outsourcing, subassemblies, price/due date tradeoffs; priority systems; allowing breakdowns or other types of unreliability.

In addition to the extensions and incremental variations described above, there are opportunities for further investigation of general issues and problems that cut across different disciplines and methodologies. One possible avenue of future research lies in the further investigation of stochastic problems. Because of the innate difficulty of combining scheduling and admission control in a stochastic setting, less than one third of the papers discussed in this survey present stochastic models. It would be interesting to see what insights the optimal policies developed in papers such as Duenyas and Hopp (1995), Duenyas (1995) and Carr and Duenyas (2000) might yield for other variations of stochastic OAS, including multiple-machine models. Another way of expanding the stochastic research on OAS would be to transform various deterministic problems to stochastic ones, as has been done by the work of Raaymakers (1999), Raaymakers et al. (2000a,b), Ivanescu (2004) and Ivanescu et al. (2002, 2006a,b), or Gupta et al. (1992) and Stadje (1995).

The complexity of deterministic OAS has been fairly extensively studied, particularly by the order rejection papers. This relatively theoretical group of papers constitutes a cohesive stream of research that (for the most part) stands separate from the rest of the OAS literature, which combines theoretical and computational analysis. It would be interesting to use the insights into the problem structure provided by the theoretical analysis to develop heuristics, and test them computationally against the algorithms presented in these order rejection papers.

The advantages of greedy/myopic approaches, the apparent dominance of joint decision making and the insights about the relative value of information in OAS are persistent themes that recur throughout this literature (see Table 3). Investigations into the theoretical underpinnings of these themes (such as Aldaee et al. (2001) for greedy/myopic single-machine problems) would further integrate the research on OAS, and likely lead to additional areas of research.

Appendix A. List of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Stands for</th>
</tr>
</thead>
<tbody>
<tr>
<td>B&amp;B</td>
<td>Branch and Bound</td>
</tr>
<tr>
<td>B&amp;P</td>
<td>Branch and Price</td>
</tr>
<tr>
<td>CON</td>
<td>CONstant due date rule</td>
</tr>
<tr>
<td>DP</td>
<td>Dynamic program</td>
</tr>
<tr>
<td>EDD</td>
<td>Earliest due-date order</td>
</tr>
<tr>
<td>EO</td>
<td>Extremal Optimization</td>
</tr>
<tr>
<td>E/T</td>
<td>Early/Tardy problem</td>
</tr>
<tr>
<td>FAS</td>
<td>Final Assembly Schedule</td>
</tr>
<tr>
<td>FCFS</td>
<td>First Come First Served order</td>
</tr>
<tr>
<td>FPTAS</td>
<td>Fully Polynomial-Time Approximation Scheme</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic algorithm</td>
</tr>
<tr>
<td>IP</td>
<td>Integer program</td>
</tr>
<tr>
<td>LP</td>
<td>Linear program</td>
</tr>
<tr>
<td>MDP</td>
<td>Markov decision process</td>
</tr>
<tr>
<td>MILP</td>
<td>Mixed integer linear program</td>
</tr>
<tr>
<td>MIP</td>
<td>Mixed integer program</td>
</tr>
<tr>
<td>MPS</td>
<td>Master Production Schedule</td>
</tr>
<tr>
<td>MTO</td>
<td>Make to Order</td>
</tr>
<tr>
<td>MTS</td>
<td>Make to Stock</td>
</tr>
<tr>
<td>NPV</td>
<td>Net Present Value</td>
</tr>
<tr>
<td>S/OPN</td>
<td>Slack per remaining OPeratioN</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>SLK</td>
<td>SLack due-date rule</td>
</tr>
<tr>
<td>OAS</td>
<td>Order acceptance and scheduling</td>
</tr>
<tr>
<td>SPT</td>
<td>Shortest Processing Time order</td>
</tr>
<tr>
<td>TMP</td>
<td>Throughput maximization problem</td>
</tr>
<tr>
<td>WSPT</td>
<td>Weighted Shortest Processing Time order</td>
</tr>
</tbody>
</table>

References
