Proceedings of the 2004 Winter Simulation Conference
R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

A STOCHASTIC ON-LINE MODEL FOR SHIPMENT DATE QUOTING WITH ON-TIME DELIVERY GUARANTEES

Yunpeng Pan
Leyuan Shi

Department of Industrial Engineering
University of Wisconsin-Madison
Madison, WI 53706, U.S.A.

ABSTRACT

The paper introduces a new model for shipment date quoting with potential applications in E-commerce. First, a customer sends to the vendor a request for an item advertised at a certain price on the company's web site. Upon receiving the request, the vendor immediately quotes the customer a no-later-than shipment date for the requested item, taking into account the amount of time to produce the item and any outstanding order previously placed but not yet fulfilled. If the quoted date is deemed acceptable, the customer subsequently places an order for the item; otherwise, the customer rejects the quote and looks for an alternative vendor (the deal is thus lost). The back-end of the quoting system is a single server production system. We propose heuristics that account for the intricate combinatorics of the server scheduling problem, as well as the uncertainty in customer demand and customer behavior.

1 INTRODUCTION

The rise of E-commerce has profoundly changed how business deals are conducted between buyers and sellers. From an online shopper's perspective, this new mode of shopping is empowering, allowing the shopper to compare the prices and qualities of goods and services from many different vendors. To a vendor, the Internet, along with inexpensive means of shipping, brings tremendous opportunities; the vendor can potentially sell into a market with virtually no geographical boundaries. Unfortunately, with the opportunities also comes more fierce competition. The winners of competition will inevitably be the greatest innovators that are willing to challenge the status quo of business practice and to constantly adapt themselves to the increasingly demanding needs of customers.

In this paper, we introduce a new model for shipment date quoting. First, a customer sends to the vendor a request for an item advertised at a certain price on the company's web site. Upon receiving the request, the vendor immediately quotes the customer a no-later-than shipment date (or *deadline*) for the requested item, taking into account the amount of time to produce the item and any outstanding order previously placed but not yet fulfilled. If the quoted date is deemed acceptable, then the customer subsequently places an order for the item; otherwise, the customer rejects the quote and looks for an alternative vendor (the deal is thus lost).

A distinctive feature of our model is the shipment date—the latest time by which the order will be shipped to the customer. This offers the customer an assurance of on-time delivery (we can safely assume the transit time to be constant for commonly used modes of transportation). By contrast, Internet retailers such as Amazon.com do not usually commit themselves to a deadline. It is therefore difficult for a customer to know with certainty when an ordered item will arrive. Clearly, our new order placement model eliminates this uncertainty.

The above scenario also reflects the empowerment of consumers. Unlike the occurrences in many supplier-manufacturer relationships, the customer here does not just passively accept the date dictated by a vendor, and can walk about from the deal literally with the click of a button. Therefore, existing due date quoting models (e.g., the one suggested by Hopp and Spearman 2000) may not be applicable here.

We propose a method that takes into account the interests of both the vendor and the customer. The vendor can certainly give the earliest date quote to maximize the possibility of securing a particular order. On the other hand, doing so could adversely affect the outstanding orders previously committed, and diminish the vendor's flexibility to honor future requests. We examine this intricacy of potential payoffs and costs associated with an order request.

The remainder of the paper is organized as follows. In Section 2, we introduce some necessary notation and discuss assumptions of our model. Our main results are presented in Section 3, including a heuristic for shipment date quoting. In Section 4, we compare the performance of the proposed heuristic with that of the adapted First-In-First-Out (FIFO) rule. In Section 5, we discuss some extension of the heuristic. A few concluding remarks are given in Section 6.

2 NOTATION AND ASSUMPTIONS

We view the stream of customer order requests as an arrival process. Interarrival times are independent identically distributed random variables with the cumulative distribution function $F(x)$. Denote the arrival epoch of a request by r_j, where $j \in \mathbb{Z}^+$ is the request index. In our model, the vendor's production system is a manufacturing work center in which a single machine or server produces a variety of make-to-order items. We assume that the vendor can precisely know the amount of time p_j needed to produce an item requested. This time p_j is assumed to be deterministic, which is a reasonable assumption for many manufacturing processes where the processing time variability of a job is negligible. Our model is said to be *on-line* because the information about a request is revealed only after it has arrived.

It is also assumed that the vendor has complete knowledge about the orders in the system, including when the order request arrived, how long it takes to produce the item, and the committed shipment date. At any moment t, let $q(t)$ be the number of orders currently waiting in the system. If an item for order j is being produced, let the residual time to finish producing the item be denoted by $\tilde{p}_j$; preemption of processing is not allowed. When the vendor quotes the customer (who submitted request j) the latest shipment date d_j, the customer either accept it, or reject it and looks elsewhere. Define the amount of additional waiting time (or slack) for the requested item by $x_j \equiv d_j - r_j - p_j$. The probability of the customer accepting a quote is a function of x_j and is monotonically decreasing. Let this acceptance probability be $g(x_j)$ for the pool of customers. The rationale behind $g(x_j)$ is that, although it is impossible to know the tolerance of individual customers to various levels of waiting, the tolerance of all customers as a whole (i.e., $g(x_j)$) can be estimated fairly accurately. The acceptance probability, $g(x_j)$, can also be viewed as the cumulative density function of a random variable representing the slacks in accepted quotes. Thus, $g(x_j)$ is a monotonically nonincreasing function.

If the customer accepts the quote, an order (with the same index as the initial request) is subsequently placed to the vendor. Fulfillment of the order generates a profit of a_j (i.e., the sale price of the item less the material and labor costs). Let the completion time of the item (i.e., the time of shipment) be C_j. Then, the throughput time of order j is $T_j = C_j - r_j$. In the spirit of reducing inventory wastes and minimizing throughput times, there is a cost of w_j per unit throughput time. The total throughput time penalty for order j is therefore $w_j T_j$.

We assume that early shipments of orders are permissible. More precisely, the customer prefers the ordered item to be shipped as early as possible before the deadline d_j, or at least, is indifferent about an early shipment. An example of this would be a customer receiving a book ordered from an Internet bookstore. Note that this is different from Just-in-Time (JIT) models in which a certain penalty may be incurred on an early shipment.

For convenience of exposition, all data takes on integer values. For all practical purposes, a continuous problem can be discretized using a sufficiently small time unit.

3 MAIN RESULTS

In this section, we present the main results of the paper. Specifically, a method is proposed for maximizing the *expected net profit*, which is defined by the profits from the orders less the throughput time costs.

Consider the costs and profits associated with a new request that arrives at time $r_n = t$. By our notation, $q(t)$ is the number of orders already in the system. Let these orders be numbered by $\mathrm{J} = \{1, \ldots, n-1 = q(t)\}$. For each order $j \in \mathrm{J}$, the processing time and the committed shipmen dates are therefore p_j and d_j, respectively. Also, there could be an order in processing, in which case, the residual time is $\tilde{p}$. Additionally, we create a tentative order n for the new request. Our objective is to identify a date d_n such that all orders can be shipped on time while the expected net profit is maximized.

3.1 The Earliest Shipment Date

The first step in our method is to determine the *earliest date* $d_n^{\min}$. By definition, if $d_n = d_n^{\min}$, then all the orders can be shipped on time; further, if $d_n < d_n^{\min}$, then at least one of the orders will be tardy. It is clear that $d_n^{\min} \geq t_0 + p_n$, where $t_0 = t + \tilde{p}$ is the time when the order in processing is completed ($t_0 = t$ if the machine is idle). Furthermore, $d_n^{\min} \leq t_0 + \sum_{j=1}^{n} p_j$, since the right hand side of the inequality is the completion time of the new order should it be worked on last. It turns out that for a fixed d_n, the question of whether all orders can be completed by their respective deadlines is a deterministic sequencing problem solvable using the earliest-due-date (EDD) rule (Pinedo 1995): If the orders are processed according to the increasing sequence of committed shipment dates and are completed on time, then this particular d_n is a feasible choice. To find $d_n^{\min}$, we incorporate the EDD rule in the following dichotomy.

Procedure Find-Min-D

Step 0. Set $a := t_0 + p_n, b := t_0 + \sum_{j=1}^{n} p_j$.

Step 1. If $a = b$, then let $d_n^{\min} = a$ and *STOP*; otherwise, set

$$d_n := \lfloor (a+b)/2 \rfloor$$

and apply the EDD rule to get the completion times C_j, $j = 1, \ldots, n$.
Let $L_j = C_j - d_j$ for all j, and let

$$L_{j_0} = \max_{1 \le j \le n} \{L_j\}.$$

Step 2. If $L_{j_0} > 0$, then set $a := d_n + L_{j_0}$; otherwise, set $b := C_n$. Go to 1.

3.2 Throughput Time Costs

Any quoted date that is no earlier than $d_n^{\min}$ would be feasible. Nevertheless, there are other considerations besides feasibility. Specifically, we also need to consider the throughput time costs, which is be expressed as $z(d_n) = \sum_{j=1}^{n} w_j(C_j - r_j)$. Here the completion times C_j $(j = 1, \ldots, n)$ are dependent of the single variable d_n, which results in the dependency of $z(d_n)$ on d_n. Since $C_j - r_j = (C_j - t_0) + (t_0 - r_j)$ with $(t_0 - r_j)$ being constant for any processing sequence, it boils down to minimizing $\sum_{j=1}^{n} w_j(C_j - t_0)$. For any fixed d_n with $d_n \ge d_n^{\min}$, the problem of minimizing $\sum_{j=1}^{n} w_j(C_j - t_0)$ while completing the orders on time is a deterministic NP-hard problem (Lenstra et al. 1977). It is commonly believed, though not yet proven, that fast polynomial-time solution procedures probably do not exist for this kind of problem. However, problem instances with 20–120 orders can still be solved optimally in a matter of seconds using a method recently suggested by Pan (2003). Even larger instances can be dealt by this procedure, although optimality is no longer guaranteed.

Let $z^*(d_n)$ be the minimum value of $z(d_n)$ for a fixed d_n. It is readily shown that $z^*(d_n)$ is a nonincreasing step function defined on the interval $[d_n^{\min}, \infty)$ with right-hand continuity. Therefore, we only need to identify the discontinuity points and the function values evaluated at those points. Starting with the largest discontinuity point, the following dichotomy procedure sequentially finds all the discontinuity points and their function values.

Procedure Evaluate-Z

Step 0. Let $t_1 = t_0 + \sum_{j=1}^{n} p_j$. Set $S := \{t_1\}$. Associated with t_1 is the value $z_{t_1}^*$, which is obtained by setting $d_n := t_1$ and subsequently applying Pan's algorithm.

Step 1. If $S = \emptyset$, *STOP*; otherwise, set

$$a := \begin{cases} 1 + \max_{t \in S, z^*(t) < z^*(t_{\max})}\{t\}, & \text{if } |S| > 1, \\ d_n^{\min}, & \text{otherwise;} \end{cases}$$

$$b := \min_{t \in S, z^*(t) = z^*(t_{\max})} \{t\},$$

where $t_{\max} = \max_{t \in S}\{t\}$.

Step 2. If $a = b$, then a is a discontinuity point and $z^*(a)$ is the associated value; hence, set $S := S \setminus \{t | t \in S, z^*(t) = z^*(a)\}$ and go to 1.
Otherwise, set

$$d_n := \lfloor (a+b)/2 \rfloor$$

and apply Pan's algorithm to obtain $z^*(d_n)$ and the associated completion times C_j, $j = 1, \ldots, n$.

Step 3. If $z^*(d_n) \ne z^*(b)$, then set

$$a := d_n + 1, \, S := S \cup \{C_n\}$$

(note that $z^*(C_n)$ is now associated with C_n); otherwise, set $b := d_n$. Go to 2.

The running time of Procedure Evaluate-Z is determined by the number of runs of Pan's algorithm as well as the time required per run. It can be shown that the running time of the procedure is on the order of $O(C \cdot \log(t_1 - t_0 + 1))$, where C is the largest computation time incurred by Pan's algorithm among the runs.

By applying Procedure Evaluate-Z, we now have full knowledge of the function $z^*(d_n)$. Suppose that there will not be any future requests after the current one. Then, an *optimal* shipment date quote (not necessary unique), which we denote by d_n^*, can be determined by maximizing the expected net profit $\pi(d_n)$ defined as follows:

$$\pi(d_n) = a_n \cdot g(d_n - r_n - p_n) - z^*(d_n) + \sum_{j=1}^{n-1} a_j \qquad (1)$$

for $d_n \in [d_n^{\min}, \infty)$, with the last term of (1) being a constant. Let the discontinuity points of $z^*(\cdot)$ be denoted by $\alpha_1 < \ldots < \alpha_m$, where m is the number of discontinuity points. Due to the monotonicity of $g(\cdot)$ and the fact that $z^*(\cdot)$ is a nonincreasing step function, an optimal quote, d_n^*, can be found among the points in $D = \{\alpha_1, \ldots, \alpha_m\}$. Hence, we may let d_n^* be the largest point in D such that

$$\pi(d_n^*) = \max_{d_n \in D} \pi(d_n). \qquad (2)$$

Using Procedure Evaluate-Z, we outline a dispatching-type heuristic below for shipment date quoting.

Procedure Heuristic-1

Step 0. Initialize $Q := \emptyset$, where Q is the set of orders with committed shipment dates. Also initialize the time horizon t. Set the cumulative cost $\rho := 0$.

Step 1. If t exceeds the end of the observation window, *STOP.*

Step 2. If $|Q| > 0$, execute the orders in Q in the sequence determined by Pan's algorithm (adjust ρ accordingly), until a new request arrives.

Step 3. Determine d_n using Procedure Evaluate-Z and the formula in (2). Present the customer with a quote of $d_n + \delta_n$, where $\delta_n = h_n(d_n - r_n - p_n)$ is a slack function (δ_n is discussed in detail in Section 4.1). If the quote is accepted, add a new order to Q and add the profit from this order to ρ. Go to 1.

4 SIMULATION RESULTS

A wide variety of dispatching rules has been previously discussed in the context of queueing theory (see van Mieghem 1995 for some references). However, the main challenge is that these rules, which select the next order to process among all queued orders, cannot be easily adapted to simultaneously handle both the optimization and feasibility issues in our model. Clearly, in choosing the next order, we need to consider whether the decision would adversely affect the on-time delivery of all the orders, as well as the costs and benefits associated with the decision. It is not clear whether straightforward adaptation of dispatching rules would perform satisfactorily.

The focus of this section is to compare the proposed heuristics with a simple adaptation of the FIFO rule, which be described as follows. The orders are processed in the First-In-First-Out manner. Upon receiving a request, we examine the amount of unfinished work (i.e., total order processing time) and quote the shipment date as the current date offset by the unfinished work and the processing time of the request, should it become an order.

4.1 Experimental Design

We model the arrival process of incoming requests as a Poisson process with rate λ. This Poisson process is approximated using the uniformly distributed arrival epochs over a chosen time interval. The processing time, though assumed to be known upon the arrival of a request, follows a uniform distribution on [1, 100]. The per unit time inventory holding cost is also uniformly distributed on [1, 10]. For a stream of n requests with the expected total processing time $55n$, let their arrival epochs be randomly dispersed over interval $[0, 55n/\eta]$, where η is the traffic intensity. The rate of the approximated process is then computed as $\lambda \approx \eta/55$. Each arrival stream comprises $n = 1000$ requests.

Also associated with a request i is a potential payment of a_i, which is uniformly distributed on $[1, a_{\max}]$. The acceptance probability of the customer pool is assumed to be $g(x) = (1 - p)^x$, which essentially implies that the amount of waiting time a customer can stand follows a geometric distribution with rate p. Thus, the experimental data is controled by three parameters, η, $a_{\max}$, and p. Each of these parameters takes on two levels of values, as indicated in Table 1. For each combination of the parameter values, 100 replications of the arrival stream are generated.

Table 1: Parameter Values

Level	p	$a_{\max}$	η
1	0.0001	1000	.5
2	0.0002	2000	.8

The slack function $\delta_n(x)$ (where $x = d_n - r_n - p_n$) in Heuristic-1 is set through experimentation, and we have determined that $\delta_n(x) = 3a_n(x + 55\eta)/(a_{\max})$ works well. The simulation program is written in C++ programming language.

4.2 Comparison of Adapted FIFO vs. Heuristic-1

The results of Adapted FIFO vs. Heuristic-1 are shown in Tables 2 and 3. It is worth noting that under some operating conditions, the average net profit is negative for both FIFO and Heuristic-1. This is attributed to the joint effect of less revenue from potential customers shied away by overcrowded systems, as well as high inventory holding costs. Even on such a loosing proposition, Heuristic-1 results in narrower loss than Adapted FIFO. In the other situations where the operation is profitable, Heuristic-1 is able to outperform Adapted FIFO by as much as 3.7 times.

Table 2: FIFO vs. Heuristic-1: Net Profit

p	$a_{\max}$	η	FIFO	Heuristic-1
0.0001	1000	0.5	44430	79844
		0.8	-414418	-183845
	2000	0.5	539373	572609
		0.8	79549	302425
0.0002	1000	0.5	43952	78948
		0.8	-416377	-182857
	2000	0.5	539578	573191
		0.8	83008	303975

Table 3: FIFO vs. Heuristic-1: % Net Profit

p	$a_{\max}$	η	FIFO	Heuristic-1
0.0001	1000	0.5	56%	100%
		0.8	-225%	-100%
	2000	0.5	94%	100%
		0.8	26%	100%
0.0002	1000	0.5	56%	100%
		0.8	-228%	-100%
	2000	0.5	94%	100%
		0.8	27%	100%

The superior performance of Heuristic-1 is largely expected, because it considers both operating costs and revenues from orders locked in. The computation time for both methods are negligible. This makes Heuristic-1 suitable to be used in real-time online quoting.

5 FURTHER EXTENSION

As stated before, the condition in (2) precisely characterizes an optimal quote d_n^*, provided that after the current request, which corresponds to order n, there will not be any future requests. However, we generally need also consider requests to come. The influence of future requests on the quoting decision now is stochastic in nature, since we do not know when these requests will arrive or what values their other parameters take on. Under such circumstances, the conjunction of combinatorial (more precisely, sequencing) and stochastic factors makes it extremely difficult to develop an exact formula for d_n^*. Therefore, we can employ simulation and line search to approximate d_n^*.

Let d_n' be the date that satisfies condition (2) (it was denoted by d_n^* above, since in that special case, it is indeed optimal). We restrict our attention to an interval $[a, b]$ such that $a \geq d_n^{\min}$ and $d_n' \in [a, b]$. In particular, consider $a = d_n^{\min}$ and $b = a + EX$, where EX is the expected interarrival time between requests.

A set of K simulation sample paths is generated, representing possible scenarios of subsequent request arrivals after the current one. We require that the sample paths all comprise N requests. The same set of sample paths is used to evaluate the performance of each $d_n \in [a, b]$. With a given d_n, the quote is either rejected or accepted by the customer. In either case, we evaluate the total expected profit by applying Heuristic-1 to a sample path. Let γ denote a sample path, and let $y_{accept}(d_n, \gamma)$ and $y_{reject}(d_n, \gamma)$ be the expected net profits in their respective cases. The expected net profit associated with d_n and sample path γ is then computed as $y(d_n, \gamma) = g(d_n - r_n - p_n)y_{accept}(d_n, \gamma) + (1 - g(d_n - r_n - p_n))y_{reject}(d_n, \gamma)$. We get an overall performance measure, $h(d_n)$, for d_n by averaging $y(d_n, \gamma)$ over all the sample paths; i.e., $h(d_n) = \max_{1 \leq \gamma \leq K} y(d_n, \gamma)$. Now, the task boils down to maximizing $h(d_n)$ on interval $[a, b]$. Since the objective function has only one variable, a line search method can be used. Here we employ the golden section method and denote by $\tilde{d}_n$, the best point found. Using $\tilde{d}_n$ as an approximation of d_n^*, we come up with the following heuristic.

Procedure Heuristic-2

Steps 0-2. The same as those of Heuristic-1.

Step 3. The same as that of Heuristic-1, except that $\tilde{d}_n$ is quoted, instead.

6 CONCLUDING REMARKS

We introduced a shipment date quoting method with potential applications in E-commerce. Our approach utilizes both combinatorial optimization and simulation optimization techniques. We established the effectiveness the proposed method through simulation studies. In the future, we plan to develop solutions of the *off-line* version of the shipment date quoting problem, which would be useful in measuring the solution quality for the on-line problem. Moreover, it is possible to simultaneously consider shipment date quoting for the current request and other requests that might arrive soon. To this end, the effect of deliberately idling the machine should be looked into, perhaps with the new tool developed in Pan and Shi (2004).

REFERENCES

Hopp, W. J., and L. Spearman. 2000. *Factory physics*. Boston, MA: Irwin, McGraw-Hill.

Lenstra, J. K., A. H. G. Rinnooy Kan, and P. Brucker. 1977. Complexity of machine scheduling problems. *Annals of Discrete Mathematics* 1:343–362.

Pan, Y. 2003. An improved branch and bound algorithm for single machine scheduling with deadlines to minimize total weighted completion time. *Operations Research Letters* 31 (6): 492–496.

Pan, Y., and L. Shi. 2004. Dual constrained single machine sequencing to minimize total weighted completion time. Submitted for publication.

Pinedo, M. 1995. *Scheduling: Theory, algorithms and systems*. Englewood Cliffs, NJ: Prentice-Hall.

van Mieghem, J. A. 1995. Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule. *The Annals of Applied Probability* 5 (3): 809–833.

AUTHOR BIOGRAPHIES

YUNPENG PAN is a research associate with the Department of Industrial Engineering at University of Wisconsin-Madison. He received a Ph.D. in Industrial Engineering (2003) and an M.S. in Computer Sciences (2001) from University of Wisconsin-Madison, an M.S. in Operations Research from University of Delaware (1998), and also a B.S. in Computational Mathematics from Nanjing University, China (1995). His current research interest is hybrid combinatorial and mathematical programming-based approaches to practical shop scheduling problems that arise from the extended enterprise supply chain network. His work has appeared or is to appear in Operations Research Letters, IEEE Trans. on Automation Science and Engineering, European Journal of Operational Research, and Journal of Systems Science and Systems Engineering. Dr. Pan is a member of INFORMS and IIE.

LEYUAN SHI is an Associate Professor of the Department of Industrial Engineering at University of Wisconsin-Madison. She received her Ph.D. in Applied Mathematics from Harvard University in 1992, her M.S. in Engineering from Harvard University in 1990, her M.S. in Applied Mathematics from Tsinghua University in 1985, and her B.S. in Mathematics from Nanjing Normal University in 1982. Dr. Shi has been involved in undergraduate and graduate teaching, as well as research and professional service. Dr. Shi's research is devoted to the theory and applications of large-scale optimization algorithms, discrete event simulation and modeling and analysis of discrete dynamic systems. She has published many papers in these areas. Her work has appeared in Discrete Event Dynamic Systems, Operations Research, Management Science, IEEE Trans., and, IIE Trans. She is currently a member of the editorial board for Journal of Manufacturing & Service Operations Management, and is an Associate Editor of Journal of Discrete Event Dynamic Systems. Dr. Shi is a member of IEEE and INFORMS.