A Dynamic Resource Reservation Policy for Multi-Class Available-to-Promise Problems

Chien-Yu Chen
School of Management
George Mason University, USA
cchen8@gmu.edu

Arm Pangarad
Systems Engineering and Operations Research
George Mason University, USA
apangara@gmu.edu

Abstract

To gain competitive advantage, firms rely on decision and information technologies to enable more responsive and efficient e-business operations. Among others, optimization-based available-to-promise (ATP) models are capable of effectively allocating and reallocating critical resources among multiple customer orders or channels within an execution horizon. However, due to relatively short execution horizon, optimization-based ATP models may experience lower profitability when repeatedly implemented over a longer rolling horizon. It is therefore important to take into account demand uncertainty among different demand classes. We model a dynamic resource reservation policy into a stochastic programming model. Our numerical experiments show that the model can obtain the optimal reservation level that gives reasonable protection to more profitable customer demand classes.

1. Introduction

Advanced information technology and expanded logistics infrastructure are reshaping the global business environment. These developments present new opportunities as well as new threats to companies all over the world. Buyers and sellers can now collaboratively make business decisions in real time, and products can be moved from place to place globally within a matter of days or even hours. As a result, the standard of doing business has been elevated to the highest level ever. In order to gain competitive advantage in the new business environment, firms are redefining their business models not only to improve front-end customer satisfaction but also to enhance back-end logistics efficiency. Available to Promise (ATP) is such a system that directly links customer orders with enterprise resources and simultaneously takes into consideration the tradeoffs between front-end and back-end performance.

An ATP system performs two fundamental order management functions: 1) order promising and 2) order fulfillment, under a resource-limited short-term operational environment. In a customer order cycle, a company has to make an order promise on delivery time, quantity, and even product configuration to each of its customers as soon as an order is processes. However, the actual order fulfillment may involve complicated, time-consuming production and distribution operations. With uncertainty in future customer orders and fulfillment operations, it is really challenging to make order promising and fulfillment decisions that balance front-end customer satisfaction and back-end logistics efficiency. Motivated by this difficulty, ATP systems are developed to support companies in making responsive, reliable, and profitable promises based on the company's actual fulfillment ability.

Among many advanced execution mechanisms, the order promising and order fulfillment functions are particularly important. These two functions are like nerve and muscle systems that aim to synchronize supply chain activities across firms. Any discrepancy between an order promising signal and the corresponding order fulfillment action could result in imbalance and instability for the entire supply chain system. Especially in an e-business arena, customers not only demand quick order promising but also expect reliable order fulfillment. Firms need ATP systems to enhance customer satisfaction as well as to sustain profitability. More and more practitioner articles state the needs of ATP systems (see [2], [6] and [13]). ATP has also gained a high interest within the research community (see [1], [5], [7], [8], [9], [10], [11] and [12]).

2. Problem Statement

Among many ATP research streams (see [1]), this paper focuses on optimizing the reservation level of a critical resource by using a stochastic programming model. In an MTO or ATO production environment, order promising decisions need to be made before a company realizes what kinds of customer orders it will see in the future. In general, customer orders are realized over time with a mix of a spectrum of profitability and resource requirement. By using mixed integer programming models, Chen, Zhao and Ball (see [3] and [4]) report an increasing trend of profitability when a company can postpone order promising decisions and process customer order requests periodically in a batch mode. However, this strategy may not be feasible or desirable in e-business operations, in which a shorter order promising cycle is a key to success.

Suppose that we can classify potential customer orders that require the critical resource into four demand classes according to their profitability and arriving time stage, as shown in Table 1. Note that Class I includes customer orders that are more profitable and arrive in the current stage, and that Class II consists of customer orders that are also more profitable but arrive in the future stage. Similarly, Classes III and IV include relatively less profitable customer orders that arrive in current stage and in future stage, respectively. Due to the fact that Class II customer orders arrive after Class III customer orders, we need to reserve enough resources for these Class II orders in order to improve overall profitability. Otherwise, Class III customer orders may consume too much, which would lead to lower overall profitability because of severe lost sales among Class II customer orders. While this reservation implies possible denial of a certain portion or even all of Class III customer orders, it make a perfect sense to satisfy as many Class I customer orders as possible with available resources because the uncertain Class II customer orders in the future have similar profitability as those in Class I. We call this revenue management approach a dynamic resource reservation policy for the multi-class available-to-promise problems.

Table 1. Demand Class Definition

Profitability	Time Stage	
	Current	Future
High	*Class I*	*Class II*
Low	*Class III*	*Class IV*

3. Stochastic Programming Formulation

The following notation is used in describing stochastic programming models for the resource reservation problem mentioned above.

I = set of demand scenarios for current high profitable customer orders (i.e., for Class I)

J = set of demand scenarios for current low profitable customer orders (i.e., for Class III)

K = set of demand scenarios for future high profitable customer orders (i.e., for Class II)

L = set of demand scenarios for future low profitable customer orders (i.e., for Class IV)

N = set of demand scenarios $n = (i, j, k, l)$, where $i \in I$, $j \in J$, $k \in K$ and $l \in L$

M = set of demand classes = $\{1,2,3,4\}$

a = availability of the critical resource

d_m^n = demand in Class m under scenarios n

p^n = probability of demand scenarios n

v_m = profit margin in Class m

u_m = lost sales penalty in Class m

h_m = inventory holding cost in Class m

α = penalty of short-fall inventory (with respect to the resource reservation level R) after sales in Class I

β = penalty of excess inventory (with respect to the resource reservation level R) after sales in Class I

γ = penalty of excess inventory (with respect to the resource reservation level R) after sales in Class III

The main decision variables include:

R = resource reservation level

S_m^n = sales in Class m under scenarios n

T_m^n = lost sales in Class m under scenarios n

I_m^n = inventory after sales in Class m under scenarios n

E^n = the short-fall inventory (with respect to the resource reservation level R) after sales in Class I under scenarios n

F^n = the excess inventory (with respect to the resource reservation level R) after sales in Class I under scenarios n

G^n = the excess inventory (with respect to the resource reservation level R) after sales in Class III under scenarios n

We can then formulate this resource reservation optimization problem into the following stochastic programming model.

Maximize

$$\sum_{n \in N} p^n \cdot \left[\begin{array}{r} \sum_{m \in M} \left(v_m \cdot S_m^n - u_m \cdot T_m^n - h_m \cdot I_m^n \right) \\ - \alpha \cdot E^n - \beta \cdot F^n - \gamma \cdot G^n \end{array} \right] \quad (1)$$

Subject to

$$S_m^n + T_m^n = d_m^n \quad \forall\, n \in N, m \in M \quad (2)$$

$$a - S_1^n = I_1^n \quad \forall\, n \in N \quad (3)$$

$$I_1^n - S_3^n = I_3^n \quad \forall\, n \in N \quad (4)$$

$$I_3^n - S_2^n = I_2^n \quad \forall\, n \in N \quad (5)$$

$$I_2^n - S_4^n = I_4^n \quad \forall\, n \in N \quad (6)$$

$$I_1^n = R - E^n + F^n \quad \forall\, n \in N \quad (7)$$

$$I_3^n = R - E^n + G^n \quad \forall\, n \in N \quad (8)$$

$$E^{(i,j,k,l)} = E^{(i,j',k',l')} \quad \forall\, (i,j,k,l),(i,j',k',l') \in N \quad (9)$$

$$F^{(i,j,k,l)} = F^{(i,j',k',l')} \quad \forall\, (i,j,k,l),(i,j',k',l') \in N \quad (10)$$

$$G^{(i,j,k,l)} = G^{(i,j,k',l')} \quad \forall\, (i,j,k,l),(i,j,k',l') \in N \quad (11)$$

$$R, S_m^n, T_m^n, I_m^n, E^n, F^n, G^n \geq 0 \quad (12)$$

The objective function (1) basically maximizes the expect total net profit, which takes into account profit margin, lost sales penalty, and inventory holding cost for customer orders in each demand class. The additional three terms are needed to assure accurate short-fall and excess inventory counting. The reason for including these additional objective function terms will be explained later when we discuss constraints (9)-(11).

We let unfulfilled demand become lost sales in constraint (2). Constraints (3)-(6) keep track of inventory (i.e., remaining availability) over time. Note that the order of sales depends on the time stage of each demand class. In each time stage, we also allow the more profitable demand class to consume any remaining availability first. In short, the sequence of sales is Classes I, III, II and IV in our model. In constraint (7), we compare the remaining availability after sales in Class I (I_1^n) with reservation level (R) under each demand scenario n . In order to maximize profit, it makes sense for Class I to bring the remaining availability below the reservation level whenever necessary due to its higher profitability. Any shortfall amount is recorded in variable E^n and any excess amount is stored in variable F^n . On the other hand, Class III should not decrease remaining availability any further, if the inventory is below the reservation level already. That is the reason that constraint (8) only accepts the same shortfall amount E^n after sales in Class III. Similarly, we store any excess amount in variable G^n . It is not hard to see that we actually adopt standard goal programming techniques here. In order to assure that these shortfall and excess decision variables behave as expected, we need to assign small penalties α , β and γ in the objective function. Otherwise, the excess variables may contain some amount that we would like to reverse for more profitable customer orders in Class II. In other words, the resulting reservation level in the optimized solution may not truly reflect the total amount needed to protect more profitable customer orders in Class II without these penalties. Constraints (9) and (10) enforce the consistency of decisions that have been made in current time stage across different demand scenarios. Although we deal with shortfall and excess variables directly, these constraints imply same Class I and Class III sales for the common current demand scenarios regardless corresponding future demand scenarios. Finally, constraint (12) states that all decision variables hold non-negative values.

4. Numerical Experiment

We conducted a small numerical experiment and found that the model was able to obtain the optimal reservation level for a hypothetical problem within a very short time. Table 2 shows three demand scenarios (i.e., low, medium, and high) along with their associated probabilities for each demand class. Assuming independent demands across all four demand classes, we ended up with 81 combinations of demand scenarios in this example problem. The probability of a particular (combination of) demand scenario could then be calculated by multiplying four corresponding probabilities together. For example, a low-low-low-low (or 25-25-25-25) demand scenario has a probability of $(0.3)^4 = 0.0081$.

Table 2. Demand Scenarios

Demand	Class I	Class II	Class III	Class IV
Low	25 (0.3)	25 (0.3)	25 (0.3)	25 (0.3)
Medium	50 (0.4)	50 (0.4)	50 (0.4)	50 (0.4)
High	75 (0.3)	75 (0.3)	75 (0.3)	75 (0.3)

Furthermore, Table 3 presents the profit margin, lost sales penalty and inventory holding cost for each demand class. The discounts on profit margins and on lost sales penalty attempted to reflect the risk associated with larger uncertainty in future time stage. Inventory holding costs are charged based on ending inventory in each time stage. We therefore neglected the inventory holding costs for Classes I and II.

Table 3. Profit and Cost Parameters

Parameter	Class I	Class II	Class III	Class IV
Profit	100	80	60	50
Lost Sales	110	90	70	60
Holding	0	0	10	50

Given an initial availability of 200 units, the model obtained the optimal reservation level as 75 units. The experimental results showed that, with the optimal reservation level, all customer orders are protected (i.e., no lost sales) in Classes I and II under all demand scenarios. The reservation level works so that all more profitable customer orders in Class II were properly protected. Most of the lost sales happened in Class IV with one exception. When we have high demands (i.e., 75 and 75 units) in Classes I and III, the optimal reservation level forces a lost sales of 25 units in Class III. The sales of 75 units in Class I bring the remaining availability down to 125 units. Because Class III are not allowed to consume any unit below the reservation level of 75 units, we only have 50 units available for customer orders in Class III under this high-high scenarios. Therefore, we need to deny 25 units in Class III in order to provide reasonable protection to more profitable customer orders in Class II.

Overall, the stochastic model was able to obtain the optimal reservation level for this small sample problem within a couple of seconds. However, the computation time is expected to increase exponentially with the number of demand scenarios due to the complex of problem structure. For example, if we consider nine demand scenarios in each demand class, the number of combinations of demand scenarios would quickly grow to $9^4 = 6561$. We actually tried one sample problem of this size, but the personal computer ran out of its 256MB memory. It seems that the ideal solution for resolving the memory problem is to pursue the commonly used L-shape decomposition technique for solving stochastic linear programming problems.

Another computation issue is regarding how to properly assign the values for penalty parameters α, β and γ. Because of the interaction between expected net profit terms and these goal-programming penalty terms in the objective function, some combinations of these penalty parameters may not lead to the actual optimal reservation level employed in a model. This phenomenon results from unexpected behavior among corresponding penalty variables E^n, F^n and G^n. However, it is very easy to check whether the corresponding penalty variables are well-behaved or not. Recall that the purpose of these penalty variables is to calculate the differences between left-hand-side and right-hand-side values in constraints (7) and (8) and to compute the short-fall or excess values. Therefore, between short-fall amount and excess amount, at least one of them should equal zero. Only under this situation, the optimal reservation level is reliable. If we find both positive values from a pair of short-fall and excess amount of a certain constraint, the optimal reservation level is not reliable. Some adjustment on penalty parameters is necessary to correct the problem. The specific penalty parameters used in the numerical experiment were 10, 5 and 5 for α, β and γ, respectively.

5. Conclusions and Recommendations

Available-to-Promise decisions have become more and more important especially under assemble-to-order business environment. Optimization-based ATP models are able to allocate and reallocate critical resources among competing customer orders. In order to maximize expected net profit over multiple time stages, we propose a dynamic resource reservation policy and develop a stochastic linear programming model. The model is able to find the optimal reservation level for a 4-demand-class problem over a small number of demand scenarios. The optimized reservation level attempts to save enough critical resource for more profitable future customer orders (in Class II) by denying some less profitable current customer orders (in Class III).

We recommend using L-shape decomposition technique for problems with a large number of demand scenarios. The decomposition method is commonly used for solving stochastic linear program. The method decomposes the original model into a master model and sub-models. Each sub-model would optimize sales and lost-sales for a single demand scenario with a given reservation level from the master model. The master model then gathers dual price information from sub-models and re-optimizes the reservation level accordingly. We iteratively solve the master model and sub-models until the process converges to the optimal solution. This method could solve an overwhelmingly large stochastic programming model by decomposing it into more manageable master model and sub-models.

Finally, we find there is a good match between stochastic ATP problems and stochastic programming models. It is a very promising future research direction to use stochastic programming models to model and solve ATP problems.

References

[1] Ball, M.O., Chen, C.-Y. & Zhao Z.-Y. "Available to Promise," in Simchi-Levi, D., Wu, S.D. & Shen, Z.-J. eds, *Supply Chain Analysis in the eBusiness Era*, Chapter 11, Kluwer Academic Publishers, Boston, 2003, 447-483.

[2] Bayles, D.L. *E-Commerce Logistics and Fulfillment: Delivering the Goods*. Prentice-Hall PTR, Upper Saddle River, New Jersey, 2001.

[3] Chen, C.-Y., Zhao, Z.-Y. & Ball, M.O. "Quantity and Due Date Quoting Available to Promise," *Information Systems Frontiers*, 2001, 3(4), 477-488.

[4] Chen, C.-Y., Zhao, Z.-Y. & Ball, M.O. "A Model for Batch Advanced Available-to-Promise," *Production and Operations Management*, 2002, 11(4), 424-440.

[5] Ervolina, T. & Dietrich, B. "Moving toward dynamic available to promise," in Gass, S.I. and Jones, A.T. editors, *Proceeding of the Supply Chain Management Practice and Research: Status and Future Directions*, 2001, 15.1-19. R. H. School of Business, University of Maryland, Manufacturing Engineering Laboratory, National Institute of Standards, and National Science Foundation.

[6] Hill Jr., S. "Trade Exchanges & Order Promising: Can Internet Trade Exchanges Deliver Order Promising Capability?" 2000, http://www.manufacturingsystems.com/.

[7] Jeong, B, Sim, S.-B., Jeong, H.-S. & Kim, S.-W. "An available-to-promise system for TFT LCD manufacturing in supply chain," *Computer & Industrial Engineering*, 2002, 43(1-2), 191-212.

[8] Kilger, C. & Schneeweiss, L. "Demand fulfillment and ATP," in Stadtler, H. and Kilger, C., editors, *Supply Chain Management and Advanced Planning: Concepts, Models, Software and Case Studies*, Chapter 8, Springer-Verlag, Berlin, Germany, 2000, 135-148.

[9] Lee, H.L. & Billington, C. "The Evolution of Supply-Chain-Management Models and Practice at Hewlett-Packard," *Interfaces*, 1995, 25(5), 43-63

[10] Moses, S., Grant, H., Gruenwald, L. and Pulat S. "Real-time Due Date Promising by Build-to-Order Environments, Working Paper, 2002.

[11] Taylor, S.G. & Plenert, G.J. Finite "Capacity Promising," *Production and Inventory Management Journal*, 1999, 40(3), 50-56.

[12] Robinson, A.G. & Dilts, D.M. "OR & ERP: A Match for the New Millennium?" *OR/MS Today*, 1999, 26(3), 30-35.

[13] Zweben, M. "Delivering on every promise: Real-time sales order promising of supply chain resources," *APICS – The Performance Advantage*, 1996, 6(3), http://www.apics.org/magazine.