Customer Order Fulfillment Based on a Rolling Horizon Available-to-Promise Mechanism: Solution by Fuzzy Approach and Genetic Algorithm

Chi-Bin Cheng and Meng-Tsung Wu

Department of Information Management, Tamkang University,
151 Yingzhuan Rd., Tamsui Dist., New Taipei City, Taiwan
cbcheng@mail.tku.edu.tw,
pistolpete@mail.im.tku.edu.tw

Abstract. This study attempts to solve a dynamic order promising problem, where customer requests arrive in a random fashion, and the producer processes customer orders on a batch basis. This decision process repeated for every predefined batching interval, and the current decision-making must take into account the previously committed orders. The problem is formulated as a mixed integer programming model with fuzzy constraints, which express the decision-maker's subjective judgment regarding customer's price tolerance. The proposed model embeds the advanced available-to-promise (AATP) concept to support accurate computation of profit and customer order promising. A genetic algorithm is developed to solve the problem. Experiments by computer simulations are carried out to demonstrate the proposed approach.

Keywords: Reverse auction, Bidding, Advanced available-to-promise, Fuzzy mathematical programming, Genetic algorithm.

1 Introduction

Many of the Fortune 2000 companies have adopted reverse auctions as a common technique for sourcing goods and services [7]. In a reverse auction, multiple qualified suppliers are invited to participate in a bidding process, in which the suppliers bid against one another over the Internet by proposing their ideal price, quantity, delivery date, etc., in an attempt to win the business.

It was noted by Cheng [4] that a profitable and promising bid depends on accurate estimates of the production costs associated with specific delivery dates, and the awareness of market competition by the decision-maker. For such consideration, Cheng [4] incorporated the concept of available-to-promise (ATP) inventory in a multiple-objective decision-making model to determine the bid price and delivery time. Basically, the ATP inventory is the uncommitted portion of a company's inventory and planned production which is maintained in the master schedule to support customer order promise [1]. The use of ATP enables the company to respond immediately to a buyer's request and facilitates satisfaction of the delivery promise [2].

D. Filev et al. (eds.), *Intelligent Systems'2014*,
Advances in Intelligent Systems and Computing 323, DOI: 10.1007/978-3-319-11310-4_41

The traditional ATP is merely a bookkeeping function in the master production schedule, and it can only render limited flexibility of the allocation of production resources. In recent years, advanced available-to-promise (AATP) is increasingly considered as an effective tool to achieve on-time delivery, to reduce the number of missed business opportunities, and to enhance revenue and profitability [6]. AATP is based not only on pre-calculated quantities, but also on effective order-driven optimization approaches that serve the front end customers from the perspective of the entire supply chain. AATP can directly link available resources with customer orders to improve the performance of a supply chain [2, 3].

Cheng and Cheng [5] integrated the AATP concept with a dynamic pricing (i.e. bidding) mechanism to address the competition between customer orders for limited production resources to improve the efficiency of resource utilization. The problem was formulated as a mixed integer programming with the bid price being constrained by a fuzzy membership function, which modeled the decision-maker's subjective judgment on market competition. Though the model of Cheng and Cheng [5] can determine a future production plan along with the bidding decision, it did not provide a mechanism for updating the production plan when new customer orders arrive. It is possible that the overall profit can be increased by changing the production plan to allow the production of new orders. This attempt can be achieved by reviewing the ATP on a rolling horizon basis. In other words, after every time interval the production resources are reallocated to fulfill previously promised orders and to promise newly-arrival orders. The present study extends the approach of Cheng and Cheng [5] to model the bidding decision with a rolling horizon AATP planning by mixed-integer programming. The proposed model enables the dynamic allocation of production resources when new orders arrive and hence improve the efficiency of resource utilization. The problem is solved by a fuzzy approach based on the concept of compromise solutions between the overall profit and the possibility to win the contract. The solution procedure is carried out by a genetic algorithm, and experiments by computer simulation are conducted to evaluate the performance of the proposed approach.

2 Modeling of Bidding Decision with Rolling Horizon AATP

A customer request involves three dimensions, namely quantity quoting (i.e. committing order quantity), delivery-date quoting (i.e. committing order quantity), and price quoting (i.e. price demanded by the supplier to deliver the order). Customer requests arrive in a random fashion, and order-promising and -fulfillment decisions are made for a batch of requests collected over a batching interval. In the production planning for fulfilling the newly-arrived orders, it must also take into account the previously committed orders in earlier runs of order review whose production has not been fully completed. In other words, the production resources usage planed in the earlier runs could be reassigned in the current review subject to that previously committed order quantities and delivery date remain unchanged.

The order-promising and -fulfillment mechanism proposed in this study is to determine which orders to accept and, for each accepted order, determine a bid price based on the cost associated with this order. The decision regarding the fulfilment of an accepted order includes the delivery time and quantity, which is translated into a production schedule consisting of production lots at various periods. The cost of an order is computed according to the production cost and the holding cost of its production lots. The objective of the proposed mixed-integer programming model is to maximize an overall profit, which is defined as the difference between the total revenue and the tangible/intangible costs, where the intangible costs are defined as the cost of denying an order.

2.1 Mixed-Integer Programming with Fuzzy Constraint

The MIP model in this study is an extension of the model proposed by Cheng and Cheng [5] to enable the execution of a rolling horizon planning. The model is presented as follows.

Decision variables:

$Q_i(t)$: quantity to be delivered to customer i at time t.

$q_i(s,t)$: quantity to be produced at time s to fulfil the order $Q_i(t)$.

$D_i(t)$: binary variable, equal to 1 if the order of customer i is delivered at time t; 0 otherwise.

Z_i: binary variable, equal to 1 if the order of customer i is promised; 0 otherwise.

c_i: unit cost to deliver order $Q_i(t)$.

p_i: unit bid price submitted to customer i.

Parameters:

O: set of newly-arrived requests.

$\hat{O}$: set of previously-promised orders.

t_e: ended time of the previous planning.

T: length of the planning horizon.

$\hat{Q}_i(t)$: quantity has been promised to customer i.

$\hat{q}_i(s,t)$: production lot that has been completed to fulfil $\hat{Q}_i(t)$.

$\hat{d}_i$: delivery date has been promised to customer i.

$[d_i^l, d_i^u]$: acceptable delivery time interval requested by customer i.

$[a_i^l, a_i^u]$: acceptable delivered quantity interval requested by customer i.

θ_i: unit penalty cost for denying the order of customer i.

r_i: unit production time required to produce $q_i(s,t)$.

$\pi_i(s)$: unit production cost of $q_i(s,t)$.

h_i: holding cost per unit of time of per unit of $q_i(s,t)$.

$K(t)$: available production capacity at time t (in time unit).

g_i: minimum acceptable gross profit rate of the supplier to deliver the order of customer i.

$\tilde{u}_i$: upper bound of unit bid price perceived by the supplier for the order of customer i, i.e. the price above which the supplier considers the bid to have no chance of success.

M: very large number.

Maximize $$P = \sum_{t=t_e+1}^{t_e+T} \sum_{i \in O \cup \hat{O}} (p_i - c_i) \cdot Q_i(t) - \sum_{i \in O} \theta_i \cdot (1 - Z_i) \quad (1)$$

Subject to:

Order promising and fulfillment:

$$\sum_{t_e+1 \le s < t} q_i(s,t) = Q_i(t), \ \forall i \in O, \text{ and } t_e+1 \le t \le t_e+T \quad (2)$$

$$Q_i(t) \ge a_i^l \cdot D_i(t), \ \forall i \in O, \text{ and } t_e+1 \le t \le t_e+T \quad (3)$$

$$Q_i(t) \le a_i^u \cdot D_i(t), \ \forall i \in O, \text{ and } t_e+1 \le t \le t_e+T \quad (4)$$

$$\sum_{d_i^l \le t \le d_i^u} D_i(t) = Z_i, \quad \forall i \in O \quad (5)$$

$$\sum_{t=1}^{T} D_i(t) \le 1, \ \forall i \in O \quad (6)$$

$$Q_i(t) = \hat{Q}_i(t), \ \forall i \in \hat{O} \quad (7)$$

$$Q_i(t) = 0, \ \forall i \in \hat{O}, \text{ and } t \ne \hat{d}_i \quad (8)$$

$$\sum_{1 \le s \le t_e} \hat{q}_i(s,t) + \sum_{t_e+1 \le s < t} q_i(s,t) = \hat{Q}_i(t), \ \forall i \in \hat{O}, \text{ and } t_e+1 \le t \le t_e+T \quad (9)$$

Production and capacity constraints:

$$\sum_{t_e+1 \le t \le t_e+T} \sum_{i \in O \cup \hat{O}} r_i \cdot q_i(s,t) \le K(s), \quad t_e+1 \le s \le t_e+T \quad (10)$$

$$q_i(s,t) \le D_i(t) \cdot M, \ \forall i \in O, \ s < t \text{ and } t_e+1 \le t \le t_e+T \quad (11)$$

$$q_i(s,t) = 0, \ \forall i \in O \cup \hat{O}, \ s \ge t \text{ and } t_e+1 \le t \le t_e+T \quad (12)$$

Order cost computation:

$$\{ \sum_{t_e+1 \le s < t} \pi_i(s) \cdot q_i(s,t) + \sum_{t_e+1 \le s < t} h_i \cdot q_i(s,t) \cdot (t-s) \} / Q_i(t) = c_i, \ \forall i \in O, \quad t_e+1 \le t \le t_e+T, \text{ and } Q_i(t) \ne 0 \quad (13)$$

$$\{ \sum_{1 \le s \le t_e} \pi_i(s) \cdot \hat{q}_i(s,t) + \sum_{t_e+1 \le s < t} \pi_i(s) \cdot q_i(s,t) + \sum_{1 \le s \le t_e} h_i \cdot \hat{q}_i(s,t) \cdot (t-s) + \sum_{t_e+1 \le s < t} h_i \cdot q_i(s,t) \cdot (t-s) \} / Q_i(t) = c_i, \forall i \in O, t_e + 1 \le t \le t_e + T, \quad (14)$$

Bid price interval:

$$p_i \ge (1 + g_i) \cdot c_i, \ \forall i \in O \quad (15)$$

$$p_i \le \tilde{u}_i, \ \forall i \in O \quad (16)$$

$$p_i = \hat{p}_i, \ \forall i \in \hat{O} \quad (17)$$

$$c_i \le \frac{\hat{p}_i}{1 + g_i}, \ \forall i \in \hat{O} \quad (18)$$

Integrality:
$Z_i \in \{0, 1\}, \forall i \in O$
$D_i(t) \in \{0, 1\}, \forall i \in O$, and $t_e+1 \le t \le t_e+T$

Non-negativity:
$Q_i(t) \ge 0, \forall i \in O, s < t$ and $t_e+1 \le t \le t_e+T$
$q_i(s,t) \ge 0, \forall i \in O \cup \hat{O}$, and $t_e+1 \le t \le t_e+T$
$c_i \ge 0, \forall i \in O \cup \hat{O}$
$p_i \ge 0, \forall i \in O$

The objective of the above model is to maximize the overall profit of the supplier, where the profit is defined as the total revenue deducted by the total costs of orders to be delivered and the penalties of all denial orders. There are four major groups of constraints. Constraints (2)-(9) are to ensure feasible deliveries, where Constraint (2) is the production plan for newly-arrived orders where $q_i(s,t)$ is the production lot to be produced at time s in order to fulfil the order $Q_i(t)$ which will be delivered at time t. Constraints (3) and (4) specify the acceptable range of delivery quantity requested by customers; and Constraints (5) and (6) define the feasible delivery time window. Constraints (7) and (8) are to guarantee the previously-promised orders will be delivered on quantity and on time. The production of a previously-promised order may have been partially carried out at the current time point; thus, Constraint (9) expresses the possible resources rearrangement for the unfinished part of such an order. Constraint (10) enforces that the total production of each period cannot exceed its available capacity, while Constraints (11) and (12) relate the existence of $q_i(s,t)$ with the delivery decision $D_i(t)$. Constraints (13) and (14) define the cost of an order, for newly-arrived orders and formerly-promised orders respectively, as the summation of its production cost and holding cost. Constraint (15) guarantees the bid prices for newly-arrived orders render a minimum profit margin of their costs, while Constraint (16) expresses the decision-maker's belief that the maximum price the customer can tolerate. It is difficult to assign a precise value to this parameter, and therefore in the current study, it is defined by a fuzzy upper-bound $\tilde{u}_i$. Finally, Constraint (17) specifies that the previously-promised prices to customers keep unchanged; and Constraint (18) guarantees the cost of a previously-promised order not increasing to harm the minimum profit margin due to production resources rearrangement.

2.2 Fuzzy Approach

The above mixed-integer programming contains a fuzzy constraint (16); and thus cannot be solved by regular optimization techniques. This study adopts the concept of Werners [8] to formulate a fuzzy approach to solve the problem.

The satisfaction of the fuzzy constraint (16) in the MIP model is evaluated through a fuzzy membership function. As shown in Figure 1(a), the fuzzy membership function of the price variable is defined as $\mu_{\text{price}}(p_i) \in [0, 1]$. Basically, this membership function models the supplier's confidence in the possibility of winning the contract with an offer price p_i. Note that the values of $p_i^{\inf}$ and $p_i^{\sup}$ in Figure 1(a) are assigned by the supplier in accordance with his experience and knowledge of the market. This membership function coincides with the fact that the possibility of winning the contract decreases when the bid price increases. By the definition of the membership function in Figure 1(a), the fuzzy constraint can be rewritten as:

$$p_i \le p_i^{\inf} + (1-\alpha)(p_i^{\sup} - p_i^{\inf}), \tag{19}$$

where α is a membership value and $\alpha \in [0, 1]$. Equation (19) means that the upper bound of the bid price has a maximum tolerance of $(p_i^{\sup} - p_i^{\inf})$, and the satisfactory of Constraint (16) decreases when the tolerance increases, or equivalently, the possibility (i.e. α) to win the contract decreases when the bid price increases from $p_i^{\inf}$ to $p_i^{\sup}$. To construct a fuzzy membership function for the objective function, the inferior (P^0) and the superior (P^1) of the objective value P are defined respectively as follows:

P^0 = maximize (1)
subject to:
(2)-(15), (17) and (18)
$p_i \le p_i^{\inf}, \forall i \in O$

and

P^1 = maximize (1)
subject to:
(2)-(15), (17) and (18)
$p_i \le p_i^{\sup}, \forall i \in O$.

As a result, the membership function of the objective P can be constructed as:

$$\mu_P(P) = \begin{cases} 1, & \text{if } P \ge P^1 \\ 1-(P^1-P)/(P^1-P^0), & \text{if } P^0 \le P < P^1 \\ 0, & \text{if } P < P^0 \end{cases} \tag{20}$$

The membership function of (20) is also graphically shown in Figure 1(b).

The optimum decision of the original problem is now considered as to maximize the conjunct satisfaction of objective value and the fuzzy constraint, where the conjunct satisfaction is obtained through a minimum operator. This concept is referred to as a max-min approach. Thus, the original MIP is reformulated as:

Maximize α
Subject to:
(2)-(15), (17) and (18)
$\mu_i(p_i) \ge \alpha, \forall i \in O$ (21)
$\mu_P(P) \ge \alpha$ (22)

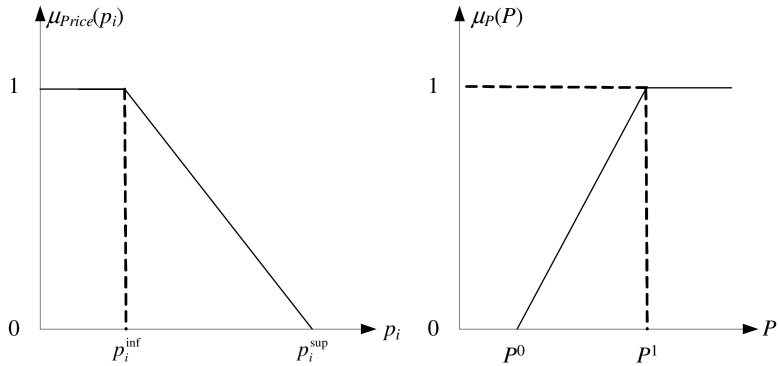

Fig. 1. (a) Membership function of the bid price (b) Membership function of the objective

3 Solution Procedure

The present study modifies the algorithm of Cheng and Cheng [5] to enable the execution of dynamic planning of order promising and fulfillment under a rolling horizon environment. There are three modules in this solution procedure, namely main algorithm, genetic algorithm, and adjustment algorithm. A conceptual diagram of the relations among the three modules are depicted in Figure 2. The main algorithm solves the max-min problem numerically by gradually increasing the value of α and activating the genetic algorithm to find a solution of order promising and fulfillment under a newly updated α value; the optimum α is then determined by applying the max-min operator on all resulting solutions under various values of α. Since the genetic algorithm does not guarantee to yield feasible solutions of order promising and fulfillment, solutions are sent to the adjustment algorithm before they can be further used in the main algorithm for solving the max-min problem.

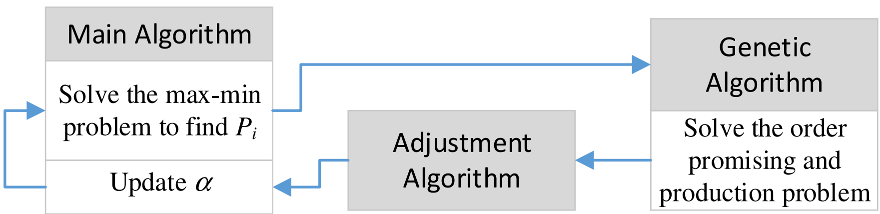

Fig. 2. Flow diagram of the solution procedure

3.1 Main Algorithm

The detailed steps of the main algorithm are described as follows.

Step 0. $zi = 1$, $\forall i \in O \cup \hat{O}$, $\alpha = 0$

Step 1. For all i∈ O, let $p_i = p_i^{\inf} + (1-\alpha)(p_i^{\sup} - p_i^{\inf})$.

For all $i \in \hat{O}$, let $p_i = \hat{p}_i$.

Step 2. Define fitness function

$$f(\alpha) = \sum_{t=t_e+1}^{t_e+T} \sum_{i \in O \cup \hat{O}} (p_i - c_i) \cdot Q_i(t) - \sum_{t_e+1 \le s < t} \lambda_s \cdot \phi(s),$$

where λ's is a unit penalty, and $\phi(s)$ is a penalty term defined as

$$\phi(s) = \begin{cases} \sum_{t=t_e+1}^{t_e+T} \sum_{i \in O \cup \hat{O}} r_i \cdot q_i(s,t) - K(s), & \text{if } \sum_{t=t_e+1}^{t_e+T} \sum_{i \in O \cup \hat{O}} r_i \cdot q_i(s,t) > K(s), \\ 0, & \text{otherwise} \end{cases}$$

Step 3. Go to Genetic Algorithm

Step 4. Go to Adjustment Algorithm

Step 5. $\alpha = \alpha + \Delta\alpha$

If $\alpha > 1$, then go to Step 5; otherwise go to Step 1.

Step 6. Construct $\mu_P(P(\alpha))$ by Eq. (20).

Step 7. Find $\alpha^* = \max_{\alpha} \min\{\mu_P(P(\alpha)), \alpha\}$

Stop.

In the beginning of the main algorithm, all n customer orders are accepted (i.e. $z_i = 1$, $i= 1,\ldots n$), and the satisfaction of Constraint (13) is set to 0 (i.e. $p_i \le p_i^{\sup}$, $\forall i$). The value of α is then gradually updated with a small increment $\Delta\alpha$, and the fitness function is defined in accordance with the current α, where the bid price pi is set to its current upper bound $p_i^{\inf} + (1-\alpha)(p_i^{\sup} - p_i^{\inf})$. This fitness function is passed to genetic algorithm for fitness evaluation of each chromosome. The solutions obtained by the genetic algorithm under different α values are returned to the main algorithm and are used to construct the membership function $\mu_P(P(\alpha))$. In the final step of the main algorithm, the optimum of the fuzzy mathematical programming problem is obtained by finding the maximum compromise between the overall profit and the bid price constraints.

It is noted that the fitness function defined in Step 1 of the main algorithm has an auxiliary penalty term, $\sum_{t_e+1 \le s < t} \lambda_s \cdot \phi(s)$. This auxiliary penalty term is to enforce the satisfaction of the capacity constraint (10).

3.2 Genetic Algorithm

Solutions of the problem are encoded as chromosome in a table format as shown in Figure 3, in which, each row represents the production lots to fulfill a customer order (n customers in total), the columns represent time periods excepting the last column denoted by t_i is the delivery time of order i, and the cells (excepting the column of t_i) denote $q_i(s, t_i)$. The steps of the genetic algorithm are as follows.

		Period					
		t_e+1	t_e+2	t_e+3	…	t_e+T	t_i
Order i	1	20	0	10	…	0	4
	2	0	20	40	…	0	6
	:	:	:	:	:	:	:
	n	0	70	0	…	0	5

Fig. 3. Solution encoding

Step 0. Initial gene pool generation:

For each chromosome do

Step 0.1 For all $i\in O$, randomly generate $t_i \in [d_i^l, d_i^u]$.

For all $i\in \hat{O}$, let $t_i = \hat{d}_i$.

Step 0.2 Let $q_i(s, t_i)$=0, if $s \geq t_i$, $\forall i\in O\cup \hat{O}$

Step 0.3 For all $i\in O$, randomly generate $qi(s, ti)$ for all $s<t_i$, where $q_i(s, t_i)$ satisfies

$$a_i^l \leq \sum_{s=t_e+1}^{t_i} q_i(s,t_i) \leq a_i^u .$$

For all $i\in \hat{O}$, randomly generate $q_i(s, t_i)$ for all $s<t_i$, where $q_i(s, t_i)$ satisfies

$$\sum_{s=t_e+1}^{t_i} q_i(s,t_i) = Q_i(t_i) - \sum_{s+1}^{t_e} q_i(s,t_i) .$$

Step 1. For each chromosome do

Fitness evaluation, where the fitness

$$F(\alpha) = \begin{cases} f(\alpha), & \text{if } p_i \geq (1+g_i)\cdot c_i, \forall i \\ 0, & \text{otherwise} \end{cases}$$

Step 2. Conduct reproduction.

Step 3. Conduct crossover.

Step 4. For each newly generated chromosome do

Step 4.1 Let $q_i(s, t_i)$=0, if $s \geq t_i$, $\forall i\in O\cup \hat{O}$

Step 4.2 For all $i\in O$,

if $\sum_{t_e+1\leq s<t_i} q_i(s,t_i) < a_i^l$, then randomly increase some $q_i(s, t_i)$, where $s<t_i$, until $\sum_{t_e+1\leq s<t_i} q_i(s,t_i) \geq a_i^l$; else if $\sum_{t_e+1\leq s<t_i} q_i(s,t_i) > a_i^u$, then randomly decrease some $q_i(s, t_i)$, where $s<t_i$, until $\sum_{t_e+1\leq s<t_i} q_i(s,t_i) \leq a_i^u$.

Step 4.3 For all i$\in \hat{O}$,

if $\sum_{t_e+1\le s<t_i} q_i(s,t_i) < Q_i(t_i) - \sum_{1\le s\le t_e} q_i(s,t_i)$, then randomly increase some $q_i(s, t_i)$, where $t_e+1\le s<t_i$, until $\sum_{t_e+1\le s<t_i} q_i(s,t_i) = Q_i(t_i) - \sum_{1\le s\le t_e} q_i(s,t_i)$; else if $\sum_{t_e+1\le s<t_i} q_i(s,t_i) > Q_i(t_i) - \sum_{1\le s\le t_e} q_i(s,t_i)$, then randomly decrease some $q_i(s, t_i)$, where $t_e+1\le s<t_i$, until $\sum_{t_e+1\le s<t_i} q_i(s,t_i) = Q_i(t_i) - \sum_{1\le s\le t_e} q_i(s,t_i)$.

Step 5. Conduct mutation

Step 6. For the chromosome picked to mutate do

Step 6.1 If the changed bit belong to $\{t_i, \forall i\}$, then

Case $i\in O$:

If $t_i < d_i^l$, then let $t_i = d_i^l$, else if $t_i > d_i^u$, then let $t_i = d_i^u$.

Repeat Steps 4.1 and 4.2.

Go to Step 7.

Case $i\in \hat{O}$: let $t_i = \hat{d}_i$.

Step 6.2 Case $i\in O$: repeat Step 4.2.

Case $i\in \hat{O}$: repeat Step 4.3.

Step 7. If the stop criterion is satisfied, then return to Main Algorithm; otherwise go to Step 1.

Step 0 generates an initial gene pool, where for new orders, the corresponding genes must satisfy the delivery time constraint and the quantity constraint requested by customers; while for previously promised orders, the delivery time cannot be changed but the incomplete portions of the order can be rescheduled under the condition that they must meet the total quantity already promised. The fitness of each chromosome is evaluated in Step 1. The fitness function passed from the main algorithm is directly employed if the chromosome satisfies the minimum gross profit constraint (15); otherwise the fitness is set to 0 to indicate this is an unacceptable chromosome. The reproduction and the crossover operators are then applied to the pool of chromosomes in Steps 2 and 3 respectively. The resulting new generation of chromosomes is adjusted in Step 4 in order to maintain their feasibility. Step 5 conducts the mutation operation and the new chromosome is adjusted in Step 6 if necessary to guarantee its feasibility.

3.3 Adjustment Algorithm

Though the auxiliary penalty term in the fitness function enforces the satisfaction of capacity constraint (10), the solutions obtained from the genetic algorithm may still fail to satisfy this constraint. Thus, it is necessary to adjust the solution to obtain feasible solutions to the original problem. The solutions from the genetic algorithm are therefore sent to the adjustment algorithm for production rearrangement/removing, or removing the entire order if necessary.

The adjustment algorithm consists of four functions, namely access, production rearrangement, production removing, and order removing. The access function is to receive the solutions from the genetic algorithm and return the adjusted solutions to the main algorithm. The adjustment of an infeasible solution starts with the rearrangement of excessive productions to periods with surplus capacities. If such rearrangement is incompatible then part of the productions will be removed in the production removing function; and if the removing function still cannot find a feasible solution then one of the orders will be removed in the order removing function. When an order is removed, the vacant capacity would allow new rearrangement of productions if there are still excessive productions at some periods.

4 Experiments and Analysis

Shorter batching intervals mean better customer response. On the other hand, a longer batching interval contains more demand patterns and hence renders a greater opportunity to maximize the profit. If shorter batching intervals can still generate comparable profits to longer batching intervals, it would be the ideal solution to the manufacturer. Thus, the experiments in the present study focus on the effects of batching interval size and capacity availability degree on various performance measures, including total profit, denial order cost, and holding cost. The batching interval size is set to five levels, 1, 2, 3, 6, and 9 periods, and the capacity availability degree is set to three levels, 100%, 80%, and 50%, where the capacity availability degree is defined as $\sum_{s=t_e+1}^{t_e+T} K(s) / \sum_{i \in O \cup \hat{O}} \frac{1}{2}(a_i^l + a_i^u)$. The combinations of different batching interval size and capacity availability degree result in 15 scenarios.

The arrival of customer requests is assumed to follow a Poisson distribution with a mean of 5 requests per period and is randomly generated according to this distribution. The parameters associated with each request are also randomly generated within pre-specified ranges. The remainder parameters are fixed as follows: the gross profit rate g_i=20%, r_i=1, π_i=2, and h_i=0.5. The planning horizon T is set to 18 periods, and the total time length under consideration is 36 periods for all scenarios. To obtain reliable results, 10 problem instances are randomly generated for each scenario. The computer program is run on a PC with Intel® Core(TM)2 Duo CPU E8400 @3.00GHz and 1.96GB RAM. Computational results are presented in Figure 4.

When capacity availability degrees are 100% and 50%, batching interval of 3 period results in the greatest profit, while there is no significant difference among the profits by different batching intervals for the case 80% capacity availability. This result is related to the balance of the revenue and the denial penalty and the holding cost. From this result, it is suggested that batching interval of 3 period is a suitable choice in our example.

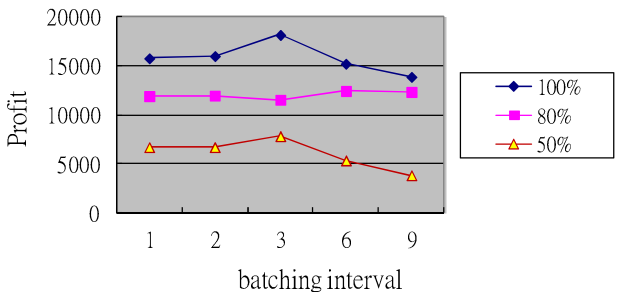

Fig. 4. Computational result

5 Concluding Remark

This study integrates the production planning and the bidding decision for customer order fulfillment decision. A mixed-integer programming model based on the concept of advanced available-to-promise inventory and fuzzy constraints on bid price was formulated. An algorithm that combines the max-min optimum approach and the genetic algorithm is developed to solve the problem. Experiments by computer simulations are carried out to demonstrate the proposed approach.

References

1. American Production and Inventory Control Society: APICS Dictionary. APICS Educational Society for Resource Manage (2004)
2. Chen, C.-Y., Zhao, Z.-Y., Ball, M.O.: Quantity and due date quoting available to promise. Information Systems Frontiers 3, 477–488 (2001)
3. Chen, C.-Y., Zhao, Z.-Y., Ball, M.O.: A model for batch advanced available-to-promise. Production and Operations Management 11, 424–440 (2002)
4. Cheng, C.-B.: Solving a sealed-bid reverse auction problem by multiple-criterion decision-making methods. Computers and Mathematics with Applications 56, 3261–3274 (2008)
5. Cheng, C.-B., Cheng, C.-J.: Available-to-promise based bidding decision by fuzzy mathematical programming and genetic algorithm. Computers & Industrial Engineering 61, 993–1002 (2011)
6. Kilger, C., Schneeweiss, L.: Demand fulfilment and ATP. In: Stadtler, H., Kilger, C. (eds.) Supply Chain Management and Advanced Planning, pp. 79–95. Springer, Berlin (2000)
7. Tully, S.: Going, going, gone! The B2B tool that really is changing the world. Fortune 141, 132–145 (2000)
8. Werner, B.: An interactive fuzzy programming system. Fuzzy Sets and Systems 23, 131–147 (1987)