The Seventeenth International Symposium on Artificial Life and Robotics 2012 (AROB 17th '12),
B-Con Plaza, Beppu, Oita, Japan, January 19-21, 2012

Optimal Base-Stock Policy of the Assemble to Order Systems

Shih-Cheng Horng[1], Feng-Yi Yang[2]

[1] Department of Computer Science & Information Engineering, Chaoyang University of Technology
Taichung, Taiwan, R.O.C.
[2] Department of Biomedical Imaging and Radiological Sciences, National Yang-Ming University
Taipei, Taiwan, R.O.C.

[1] schong@cyut.edu.tw
[2] fyyang@ym.edu.tw

Abstract: In this work, an ordinal optimization based evolution algorithm (OOEA) is proposed to solve for a good enough target inventory level of the assemble to order (ATO) system. First, the ATO system is formulated as a combinatorial optimization problem with integer variables that possesses a huge solution space. Next, the genetic algorithm (GA) is used to select N excellent solutions from the solution space, where the fitness is evaluated with the radial basis function (RBF) network. Finally, we proceed with the OCBA technique to search for a good enough solution. The proposed OOEA is applied to an ATO system comprising 10 items on 6 products. The good enough target inventory level obtained by the OOEA is promising in the aspects of solution quality and computational efficiency.

Keywords: ordinal optimization, genetic algorithm, radial basis function, optimal computing budget allocation, assemble to order system.

1 INTRODUCTION

Assemble to order (ATO) systems refer to a production strategy in which the customer must first order specified products before the item is manufactured [1]. The main advantage of ATO systems is that customers can quickly receive products customized to meet their requirements. Since the optimal inventory policies for the general ATO systems are not known, heuristic policies such as independent base-stock policies or batch-ordering policies are often used in practice to manage the component inventories. However, there is an exponential growth in the huge solution space as the number of items increases. The huge solution space makes the considered problem very hard for conventional heuristic methods to find near-optimal base-stock policy in a reasonable time.

To overcome the drawback of consuming much computation time for complex ATO systems, we propose an ordinal optimization based evolution algorithm (OOEA) to solve for a good enough target inventory level within a reasonable amount of time. The key idea of the OOEA is to narrow the solution space stage by stage or gradually restrict the search through iterative use of ordinal optimization (OO) theory [2]. The OO theory has been successfully applied to cope with the NP-hard optimization problems such as the wafer probe testing process [3], the cyclic service of the centralized broadband wireless networks [4], and resource allocation of grid computing system [5]. The ATO system operating with a continuous-review base-stock policy can be formulated as a combinatorial optimization problem that possesses a huge solution space and is most suitable for demonstrating the validity of the OOEA.

Therefore, the purpose of this paper is to determine a good enough target inventory level using limited computation time such that the expected total profit per period is maximized. The developed mathematical formulation and simulation procedure can be used for any distributions of arrival processes and production times. The proposed OOEA consists of diversification stage and intensification stage. A radial basis function (RBF) network [6] is firstly treated as a surrogate model to roughly evaluate the objective value of an inventory level. In the diversification stage, the genetic algorithm (GA) [7] is utilized to efficiently select N excellent solutions from the solution space where the fitness is evaluated with an RBF network. In the intensification stage, we proceed with the optimal computing budget allocation (OCBA) method [8], which allocates the computing resource and budget by iteratively and adaptively selecting the excellent solutions.

We organize our paper in the following manner. In Section 2, we present a mathematical formulation of a general ATO system. In Section 3,we illustrate the proposed OOEA. In Section 4, the proposed OOEA is applied to an ATO system that comprises 10 items on 6 products. Finally, we draw a conclusion in Section 5.

2 ASSEMBLE TO ORDER SYSTEMS

2.1 Problem Statement

The ATO system consists of a set of h key items and m non-key items from which n products are assembled [9]. Orders for each of n different products arrive according to independent Poisson processes with constant arrival rates $\lambda_i, i=1,\ldots,n$. Products are made up of a collection of items of different types. Items are either key items or non-key items. If any of the key items are out of stock, then the product order is lost. If all key items are in stock, the order is assembled from all key items and the available non-key items. Product i requires $a_{i,j}$ items of type j, where j ranges from 1 to (h+m). Each item sold brings a profit p_j, and each item in inventory has a holding cost per unit time of h_j, $j=1,\ldots,h+m$. There are inventory capacities C_j for each item, so that $0 \le x_j \le C_j$, where x_j is the inventory level of item j, $j=1,\ldots,h+m$. The production time for each item is truncated normally distributed with mean μ_j and variance σ_j^2, $j=1,\ldots,h+m$, lower bound 0 and upper bound ∞ and is independent of the products' arrival processes.

The ATO system operates with a continuous-review base-stock policy under which each item has a target base stock x_j, $j=1,\ldots,h+m$, and each demand for an item triggers a replenishment order for that item. There are (h+m) machines, each producing a single type of item. Items are produced one at a time on dedicated machines. We wish to maximize the expected total profit per unit time by selecting the target inventory level for a given arrival rate.

2.2 Mathematical Formulation

The ATO system under the assumed arrival rates for each order can be formulated as follows:

$$\max_{\mathbf{x}\in\Omega} E[f(\mathbf{x})]$$
$$0 \le x_j \le C_j,\ x_j \in Z,\ j=1,\ldots,h+m. \quad (1)$$

where $\mathbf{x}=[x_1 \cdots x_{h+m}]$ denote the vector of inventory level, $E[f(\mathbf{x})]$ denotes the expected total profit per unit time of $\mathbf{x}$. The inequality constraint for each x_j is $0 \le x_j \le C_j$, where the upper bound ensures that we will not go over capacity. Apparently, the solution space is

$$\Omega = \{\mathbf{x}=(x_1,\cdots,x_{h+m}) \mid 0 \le x_j \le C_j,\ x_j \in Z,\ j=1,\ldots,h+m\} \quad (2)$$

where Ω consists of $(C_j+1)^{h+m}$ possible $\mathbf{x}$. Suppose that $h=8$, $m=2$ and $C_j=20$ for $j=1,\ldots,10$, the size of Ω will be 21^{10}, which is very huge. The relationship between inputs and output of ATO stochastic simulated procedures is described in Fig. 1, in which $\mathbf{x}$ and $\boldsymbol{\lambda}$ are the input variables, $f(\mathbf{x})$ is the output objective value, T_w denotes the warm-up period, and $T_{\max}$ denotes the predetermined measurement of time. Each simulation replication should start from a fully stocked system with no orders in production, have a warm-up period of T_w time units, and then capture statistics for the next $T_{\max}-T_w$ time units of operation.

In practice, it is impossible to perform an infinitely long simulation. Thus, the sample mean of the stochastic simulation for a given value of $\mathbf{x}$ is defined as

$$\bar{f}(\mathbf{x}) = \frac{1}{L}\sum_{l=1}^{L} f_l(\mathbf{x}) \quad (3)$$

where L is the total number of simulation replications, $f_l(\mathbf{x})$ denotes the objective value of the lth simulation replication. The $\bar{f}(\mathbf{x})$ is an approximation to $E[f(\mathbf{x})]$, and $\bar{f}(\mathbf{x})$ is asymptotically close to $E[f(\mathbf{x})]$ as L increases. Thus, let $L_e=10^4$ represent the sufficiently large L. In the sequel, we define the exact model of (1) as when $L=L_e$. For the sake of simplicity in expression, we let $\bar{f}_e(\mathbf{x})$ denote the objective value of a given $\mathbf{x}$ computed by exact model.

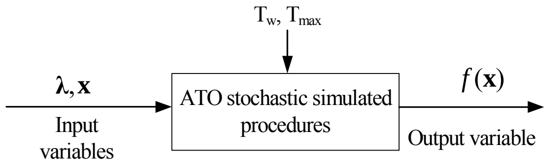

Fig. 1. Relationship between the inputs and the output of ATO stochastic simulated procedures

3 SOLUTION METHOD

3.1 RBF network

The RBF networks are three layer networks including input source nodes, hidden neurons with basis functions, and output neurons with linear activation functions, as shown in Fig. 2 [6].

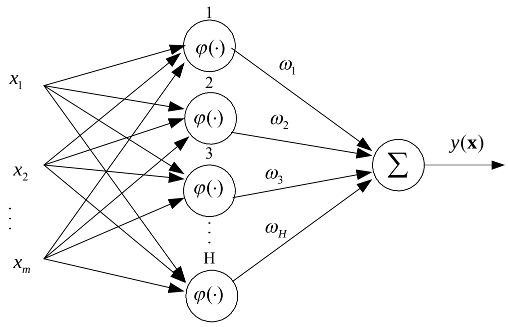

Fig. 2. Structure of an RBF network

An RBF network with a single output can be expressed as follows:

$$y(\mathbf{x}) = \sum_{h=1}^{H} \omega_h \varphi(\|\mathbf{x}-\mathbf{z}_h\|) \quad (4)$$

where $y(\mathbf{x})$ is the objective value of a given $\mathbf{x}$, H is the number of hidden nodes, $\mathbf{z}_h$ are the centers of RBF network, $\varphi(\cdot)$ is a set of RBF, ω_j are weight coefficients, and $\|\cdot\|$ is usually the Euclidean norm. In this work, the Gaussian function is used as the RBF. Once the centers have been chosen and fixed, we use the given M samples to find the $\boldsymbol{\omega}=[\omega_1,\ldots,\omega_H]$ by setting up the M equations:

$$\begin{aligned}
&\omega_1\varphi(\|\mathbf{x}^{(1)}-\mathbf{z}_1\|)+\omega_2\varphi(\|\mathbf{x}^{(1)}-\mathbf{z}_2\|)+\ldots+\omega_H\varphi(\|\mathbf{x}^{(1)}-\mathbf{z}_H\|)=y^{(1)}\\
&\omega_1\varphi(\|\mathbf{x}^{(2)}-\mathbf{z}_1\|)+\omega_2\varphi(\|\mathbf{x}^{(2)}-\mathbf{z}_2\|)+\ldots+\omega_H\varphi(\|\mathbf{x}^{(2)}-\mathbf{z}_H\|)=y^{(2)}\\
&\qquad\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots\\
&\omega_1\varphi(\|\mathbf{x}^{(M)}-\mathbf{z}_1\|)+\omega_2\varphi(\|\mathbf{x}^{(M)}-\mathbf{z}_2\|)+\ldots+\omega_H\varphi(\|\mathbf{x}^{(M)}-\mathbf{z}_H\|)=y^{(M)}
\end{aligned} \quad (5)$$

The above equations can be solved using the least squares error as long as the vectors $\mathbf{x}^{(i)} \neq \mathbf{x}^{(j)}$, for $i \neq j$. Once we have the weight coefficients and centers, we can evaluate the RBF network at a new sample by using (4) as a guide. We let $F(\mathbf{x}, \boldsymbol{\omega})$ denote the functional output of the trained RBF network of a given $\mathbf{x}$.

3.2. The Genetic Algorithm

GA is a stochastic search algorithm based on the mechanism of natural selection and natural genetics. By the aid of the above effective objective value evaluation model, we can select N excellent solutions from Ω using GA. Assuming an initial random population produced and evaluated, genetic evolution takes place by means of three basic genetic operators: (a) parent selection; (b) crossover; (c) mutation [7]. The chromosome in GA terminology represents a solution $\mathbf{x}$, and each chromosome is encoded by a string of 0s and 1s. Parent selection is a simple procedure whereby two chromosomes are selected from the parent chromosome based on their fitness values. Solutions with high fitness values have a high probability of contributing new offspring to the next generation. The selection rule we used in our approach is a simple roulette-wheel selection. Crossover is an extremely important operator for the GA. It is responsible for the structure recombination and the convergence speed of the GA and is usually applied with relatively high probability denoted as p_c. The chromosomes of the two parents selected are combined to form new chromosomes that inherit segments of information stored in parent chromosomes. The crossover scheme we employed is the single-point crossover. Mutation is the operator responsible for the injection of new information. With a small probability, random bits of the offspring chromosomes flip from 0 to 1 and vice versa and give new characteristics that do not exist in the parent chromosome. In our approach, the mutation operator is applied with a relatively small probability denoted as p_m to every bit of the chromosome.

We randomly selected I chromosomes from Ω to be the initial populations of the GA. The GA is iterated until the maximum iteration count $k_{\max}$ is reached. After reaching the stop criterion, we rank the final generation of these I chromosomes based on their fitness values and pick the top N chromosomes, which form the N excellent solutions.

3.3. The OCBA technique

First, a small number of simulation replications denoted by n_0 is applied to calculate the means and variances of the sample mean of objective value for the N excellent solutions. Let T denote the allowable computing budget for selecting the best solution and n_i denote the number of simulation replications allocated to the i th solution from T. We increase the computing budget by Δ for each iteration, and the criteria of the OCBA technique is to optimally allocate T to n_1, n_2, …, n_N with $n_1 + n_2 + \cdots + n_N = T$, such that the probability of selecting the best solution is maximized. The value of the allowable computing budget is determined by $T = (N \times L_e)/s$, where s is a speed-up factor for corresponding N using the OCBA technique. The OCBA technique can be stated in the following [8].

Step 0. Perform a small number of simulation replications n_0 for all N excellent solutions. Set $l = 0$, $n_1^l = n_0, \ldots, n_N^l = n_0$, and set the value of Δ and T.

Step 1. If $\sum_{i=1}^{N} n_i^l \geq T$, stop and output the best solution $\mathbf{x}^*$ with the maximum number of simulation replications; else, go to Step 2.

Step 2. Increase Δ additional number of simulation replications to $\sum_{i=1}^{N} n_i^l$, and compute the new allocation of simulation replications by $n_j^{l+1} = (\sum_{i=1}^{N} n_i^l + \Delta)/(1 + \alpha_b^l + \sum_{i=1, i\neq j\neq b}^{N} \alpha_i^l)$, $n_b^{l+1} = \alpha_b^l n_j^{l+1}$, and $n_i^{l+1} = \alpha_i^l n_j^{l+1}$ for all $i \neq j \neq b$, where $\alpha_i^l = \left(\frac{\delta_i^l \times (\bar{f}_b^l - \bar{f}_j^l)}{\delta_j^l \times (\bar{f}_b^l - \bar{f}_i^l)} \right)^2$, $\alpha_b^l = \delta_b^l \sqrt{\sum_{i=1, i\neq b}^{N} (\frac{\alpha_i^l}{\delta_i^l})^2}$, $\bar{f}_i^l = \frac{1}{n_i^l} \sum_{k=1}^{n_i^l} f_k(\mathbf{x}_i)$, $\delta_i^l = \sqrt{\frac{1}{n_i^l} \sum_{k=1}^{n_i^l} \left(f_k(\mathbf{x}_i) - \bar{f}_i^l \right)^2}$, $\mathbf{x}_i$ represents the i th excellent solution, $f_k(\mathbf{x}_i)$ denotes the objective value of $\mathbf{x}_i$ at the k th simulation replication, and $b = \arg \max_i \bar{f}_i^l$.

Step 3. Perform additional $\max(0, n_i^{l+1} - n_i^l)$ simulation replications for the i th excellent solution, $i = 1, \ldots, N$. Set $l = l + 1$ and go to Step 1.

3.4 The OOEA

Now, the proposed OOEA can be stated as follows.

Step 1: Randomly select M $\mathbf{x}$'s from Ω. Compute the corresponding $\bar{f}_e(\mathbf{x})$ for each $\mathbf{x}$. Train an RBF network and calculate its $\boldsymbol{\omega}$ using the obtained M input-output pairs, ($\mathbf{x}$, $\bar{f}_e(\mathbf{x})$)'s. Let $F(\mathbf{x}, \boldsymbol{\omega})$ denote the functional output of the trained RBF network.

Step 2: Randomly select I $\mathbf{x}$'s from Ω as the initial population. Apply the GA to these individuals assisted by RBF network, $F(\mathbf{x}, \boldsymbol{\omega})$. After the GA converges, we rank all the final I $\mathbf{x}$'s based on their approximate fatnesses and select the top N $\mathbf{x}$'s to be the excellent solutions.

Step 3: Use the OCBA technique to select the best $\mathbf{x}$ from the N excellent solutions, and this $\mathbf{x}$ is the good enough solution that we seek.

4 TEST RESULTS

To gain more attention from broad readers, we set up a similar example provided in [10] as the application example. The ATO system has six types of product orders (n=6) and ten items including eight key items (h=8) and two non-key items (m=2). Different types of product orders come into the system as Poisson arrival processes with different rates, $\lambda_i, i = 1, \ldots, 6$, and each of them requires a set of key items and a set of non-key items. Each item sold brings a profit, p_j, and each item in inventory has a holding cost per period, h_j, $j = 1, \ldots, 10$. There are inventory capacities for each item, C_j,

$j = 1,\ldots,10$, such that $0 \le x_j \le C_j$, and the production time for each item is normally distributed with mean μ_j and variance σ_j^2, $j = 1,\ldots,10$, truncated at 0. All parameters used are included in Tables 1 and 2. The measurement of time is assumed to start T_w=20 up until $T_{\max}$=70.

Table 1. Parameters related to ten items

Item	1	2	3	4	5	6	7	8	9	10
p_j	1	2	3	4	5	6	7	8	9	10
h_j	2	2	2	2	2	2	2	2	2	2
μ_j	0.15	0.40	0.25	0.15	0.25	0.08	0.13	0.40	0.08	0.13
σ_j	0.0225	0.06	0.0375	0.0225	0.0375	0.012	0.0195	0.06	0.012	0.0195
C_j	20	20	20	20	20	20	20	20	20	20

Table 2. Parameters related to six products

Product	λ_i	Key items								Non-key items	
		$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	$a_{i,5}$	$a_{i,6}$	$a_{i,7}$	$a_{i,8}$	$a_{i,9}$	$a_{i,10}$
1	4.2	1	0	0	1	0	1	1	0	0	1
2	3.6	1	0	0	0	1	1	1	0	0	1
3	3	0	1	0	1	0	1	0	1	1	1
4	2.4	0	0	1	1	0	1	0	1	1	0
5	1.8	0	0	1	0	1	1	1	0	0	1
6	1.2	0	1	1	1	0	1	0	1	0	0

To construct the surrogate model, we use an RBF network consisting of 10 neurons in input layer, 20 neurons in hidden layer, and 1 neuron in output layer. The 10 neurons in the input layer are for $\mathbf{x}$, and the single output neuron is for $E[f(\mathbf{x})]$. The spread of Gaussian function is set to $\sigma = 1$. The system's condition is the six arrival rates of product orders, $\lambda_i, i = 1,\ldots,6$, which are given in Table 2. For a given system condition, we train the RBF network by randomly selecting M=9604 $\mathbf{x}$'s from the discrete solution space Ω first, then evaluate the corresponding $\bar{f}_e(\mathbf{x})$. We use the above 9604 pairs of $(\mathbf{x}, \bar{f}_e(\mathbf{x}))$'s as the input and output pairs to train the BRF network by calculating its weight coefficients. Once the RBF network is trained, the approximate objective value of $E[f(\mathbf{x})]$ for a given $\mathbf{x}$ can be obtained from the output of RBF network.

We have simulated the OOEA for two cases of N with N=100 and 50. The following parameters are used in GA: I=1000, $p_c = 0.8$, $p_m = 0.03$, and $k_{\max} = 30$. The following parameters are used in OCBA: n_0=20, Δ=20, and $L_e = 10^4$. Since s=25 and 20 corresponding to N=100 and 50, respectively, the parameters T used in OCBA are different in the two cases: T= 40000 and 25000 for N=100 and 50, respectively. The good enough target inventory level $\mathbf{x}^*$, the corresponding $\bar{f}_e(\mathbf{x}^*)$, and the consumed CPU times for the two cases are presented in Table 3. Apparently, as N increases, the corresponding $\bar{f}_e(\mathbf{x}^*)$ increases, however the consumed CPU time also increases. Above all, the CPU times consumed in all cases are within two minutes, which are very fast.

Table 3. The good enough target inventory level $\mathbf{x}^*$, the corresponding $\bar{f}_e(\mathbf{x}^*)$, and the consumed CPU times for the two cases of N

N	s	T	Good enough target inventory level $\mathbf{x}^*$	$\bar{f}_e(\mathbf{x}^*)$	CPU times (minute)
100	25	40000	[3 4 4 13 7 4 4 2 5 8]	237.83	1.91
50	20	25000	[3 7 5 14 9 4 4 2 8 9]	236.06	1.66

5 CONCLUSION

In this work, we have proposed an OOEA to solve for a good enough target inventory level of an ATO system using reasonable computation time. By the aid of the RBF network, the objective value of an inventory level can be roughly evaluated without consuming much computation time. Via stochastic simulation optimization, the arrival processes and production times of ATO system can be from any distributions, and the dimension of the problem can be high.

ACKNOWLEDGMENT

This research work is supported in part by the National Science Council in Taiwan, R.O.C., under grant NSC 100-2221-E-324-006.

REFERENCES

[1] Xiao Y, Chen J, Lee CY (2010), Optimal decisions for assemble-to-order systems with uncertain assembly capacity. Int J Prod Econ 123(1):155-165

[2] Ho YC, Zhao QC, Jia QS (2007), Ordinal Optimization: Soft Optimization for Hard Problems. Springer-Verlag, New York

[3] Horng SC, Lin SS (2009), An ordinal optimization theory based algorithm for a class of simulation optimization problems and application. Expert Syst Appl 36(5):9340-9349

[4] Horng SC, Lin SS (2011), Optimal cyclic service of the centralized broadband wireless networks with k-limited discipline. Simul Model Pract Th 19(1):382-392

[5] Horng SC (2011), Ordinal optimization based approach to the optimal resource allocation of grid computing system. Math Comput Model 54(1-2):519-530

[6] Kumar R, Ganguli R, Omkar SN (2010), Rotorcraft parameter estimation using radial basis function neural network. Appl Math Comput 216(2):584-597

[7] Goldberg DE, Sastry K (2010), Genetic Algorithms: The Design of Innovation. 2nd ed., Springer-Verlag, New York

[8] Chen CH, Lee LH (2010), Stochastic Simulation Optimization: An Optimal Computing Budget Allocation. World Scientific, New Jersey

[9] ElHafsi M (2009), Optimal integrated production and inventory control of an assemble-to-order system with multiple non-unitary demand classes. Eur J Oper Res 194(1):127-142

[10] Hong LJ, Nelson BL (2006), Discrete optimization via simulation using COMPASS. Oper Res 54(1):115-129