

Genetic Algorithm-Based Coordinated Replenishment in Multi-Item Inventory Control

Keisuke Nagasawa*, Takashi Irohara

Department of Information and Communication Sciences, Sophia University, Tokyo, Japan

Yosuke Matoba, Shuling Liu

Fairway Solutions Inc., Tokyo, Japan

(Received: January 31, 2013 / Revised: July 1, 2013 / Accepted: August 28, 2013)

ABSTRACT

We herein consider a stochastic multi-item inventory management problem in which a warehouse sells multiple items with stochastic demand and periodic replenishment from a supplier. Inventory management requires the timing and amounts of orders to be determined. For inventory replenishment, trucks of finite capacity are available. Most inventory management models consider either a single item or assume that multiple items are ordered independently, and whether there is sufficient space in trucks. The order cost is commonly calculated based on the number of carriers and the usage fees of carriers. In this situation, we can reduce future shipments by supplementing items to an order, even if the item is not scheduled to be ordered. On the other hand, we can reduce the average number of items in storage by reducing the order volume and at the risk of running out of stock. The primary variables of interest in the present research are the average number of items in storage, the stock-out volume, and the number of carriers used. We formulate this problem as a multi-objective optimization problem. In a numerical experiment based on actual shipment data, we consider the item shipping characteristics and simulate the warehouse replenishing items coordinately. The results of the simulation indicate that applying a conventional ordering policy individually will not provide effective inventory management.

Keywords: Inventory Management, Coordinated Replenishment, Ordering Policy, Genetic Algorithm

* Corresponding Author, E-mail: nagasa-k@sophia.ac.jp

1. INTRODUCTION

Effective inventory management is important to the success of supply chain management. As such, inventory management has been investigated extensively. Tiacci and Saetta (2009) considered the relationship between forecasting and ordering policy along with carrier capacity. Xaio *et al.* (2011) considered the importance of inventory based on a loss rule. Berling and Marklund (2006) used a linear regression technique to obtain approximate values of the induced backorder cost in a one-warehouse multiple-retailer system.

In inventory management, item shortages can be result in either backorders or lost sales. Recently, a number of studies have investigated the lost-sales model. For the

case in which an item is out of stock, Gruen *et al.* (2002) revealed that only 15% of customers are willing to wait for the item to become available again. The other 85% of customers buy a different product, visit another store, or do not make a purchase. According to Zipkin (2008), cost deviations can reach 30% when the lost-sales system is approximated using a backorder model. Janakiraman *et al.* (2007) compared the performance of optimal replenishment policies in lost-sales and backorder models. Huh *et al.* (2009) compared the performance of base-stock policies in lost-sales and backorder models. Levi *et al.* (2008) proposed a dual-balancing policy, in which the risk of ordering and holding are balanced and proved that the expected total cost of this policy is at most twice the

expected cost of the optimal policy. Bijvank and Vis (2011) classified models based on the characteristics of the inventory system and reviewed their proposed replenishment policies. Van Donselaar and Broekmeulen (2013) showed that the assumption that the lost sales system can be simply approximated by a backordering system if the target fill rate is at least 95% may lead to serious approximation errors.

The single warehouse multi-retailer (SWMR) problem is a well-known inventory problem (Chan *et al.*, 2002; Yang *et al.*, 2012). In the general SWMR problem, retailers know the external demand for products over a finite planning horizon, called deterministic demand. Items are shipped from the suppliers to the warehouse and are distributed from the warehouse to the retailers. The goal of this SWMR problem is to find an optimal replenishment timing and quantity to minimize the total transportation cost or the inventory cost in the system. Yang *et al.* (2012) applied a genetic algorithm (GA) to a problem which involves known multi-item demand, multiple retailers, a finite planning horizon, cost functions that include discounts, and total cost optimization.

The full-truckload problem considers the delivery of products by trucks having a finite capacity. When fixed transportation costs are charged irrelevant to ordered volume, high coordination of orders and full-truckload shipments can benefit. Kiesmüller (2009) proposed a dynamic order-up-to policy resulting in full-truckloads. Kiesmüller (2010) provided approximation formulas for computing the parameters of the replenishment policy, which takes into account the capacity restrictions of the total order volume, such that given target service levels can be met.

Next, we consider a multi-item inventory, which sells and replenishes multiple items. For inventory management, we must decide when and how much should be ordered. A number of inventory management models consider either a single item or are based on the assumption that multiple items are ordered independently, even if the total order volume is small and there is room available on the truck. The order cost is commonly calculated by factoring in the number of carriers and the carrier usage fees. Therefore, since the sum of the order volume is often not equal to the sum of the carrying capacity, we can reduce the number of carriers needed in the future by supplementing items to an order. On the other hand, we could reduce the average number of items in storage by reducing the order volume, which risks running out of stock.

In the present study, we examine the relationship between lost-sales, the holding cost of each item, and the number of trucks used for shipping. Therefore, we define this problem as a multi-objective optimization problem. We can consider this problem to be a full-truckload problem with some constraints. However, our problem

can include the case of full-truckload problem and can solve this problem using a GA. Thus, we herein propose a new GA scheme for finding a reasonable order supplementation/reduction policy.

In the present study, we focus on obtaining an order supplementation/reduction rule based on each decided ordering policy of every item for the stochastic multi-item inventory problem involving carrier capacity and lost-sales. The remainder of the present paper is organized as follows. The mathematical formulation for the deterministic demand model is presented in Section 2. In Section 3, we propose a GA in order to obtain a better solution by supplementation/reduction rule for the stochastic demand problem when ordering policy for each item is decided. The computational results are presented in Section 4. Finally, conclusions are presented in Section 5.

2. PROBLEM FORMULATION

We herein assume the existence of shipping data comprising the shipping date and order volume of items, as shown in Table 1. We also assume that the order quantity of each item in each period or the order policy for each item is initially decided individually, based on the shipping data. In this situation, we consider a policy or rule for the loading of additional items in order to fill the remaining space in the trucks.

The multi-item inventory problem considered herein is expressed simply as follows when the demands are deterministic:

minimize

$$f_1 = \sum_{\forall t \in T} u_t y_t \quad (1)$$

$$f_2 = \sum_{\forall i \in I} \sum_{\forall t \in T} p_t^i s_t^i \quad (2)$$

$$f_3 = \sum_{\forall i \in I} \sum_{\forall t \in T} h_t^i l_t^i \quad (3)$$

subject to

$$l_{t-1}^i + x_{t-LT}^i - l_t^i + s_t^i = d_t^i, \quad \forall i \in I, \forall t \in T \quad (4)$$

$$x_t^i = m^i o_t^i \quad \forall i \in I, \forall t \in T \quad (5)$$

Table 1. Shipping data

Item no.	Date: T		
	Day : 1	...	Day : t
1	d_1^1	...	d_t^1
...
n	d_1^n	...	d_t^n

$$\sum_{\forall i \in I} v^i x_t^i \leq cy_t \quad \forall t \in T \quad (6)$$

$$y_t, s_t^i, l_t^i, x_t^i, o_t^i \in Z_+ \quad \forall i \in I, \forall t \in T$$

Decision Variables

- y_t : the number of carrier used in t^{th} day
- s_t^i : the lost sales quantity of item i on the t^{th} day
- l_t^i : inventory level of item i on the t^{th} day
- x_t^i : the ordering quantity of item i on the t^{th} day
- o_t^i : the number of ordering lot of item i on the t^{th} day

Parameters

- u_t : usage fee of the carrier on the t^{th} day
- p_t^i : lost sales penalty of item i on the t^{th} day
- h_t^i : cost of holding one unit item i on the t^{th} day
- d_t^i : demand of item i on the t^{th} day
- m^i : lot size of item i
- v^i : volume of one unit item i
- c : capacity of carrier
- LT : lead time
- I : the set of items
- T : the set of days

The first objective function (Eq. (1)) considers the transportation cost and the usage fee of the truck within the planning period. The second objective function (Eq. (2)) considers the shortage cost within the planning period. The third objective function (Eq. (3)) considers the holding cost within the planning period. Eq. (4) is the inventory balancing equation for all items. Eq. (5) is a constraint on the order quantity equation for all items within the planning period such that the order quantity of item i can only be an integral multiple of the lot size of item i . Eq. (6) is a constraint on the quantity of trucks that can be loaded with order quantity within the planning period. During each period, the total volume of ordered items cannot exceed the truck volume multiplied by the number of available trucks.

If the coefficient of each objective function can be determined, a single objective function can be constructed for the problem. However, even in this case, the problem remains NP-hard. For the problem considered here, the coefficient of each objective function cannot be determined, and so the optimization problem involves three objective functions.

3. IMPLEMENTATION OF THE GENETIC ALGORITHM

In the present study, the mathematical formulation could not be used because the demands are stochastic where the inventory manager did not know future demand when we order items. We apply the ordering policy for each item and

estimate the ordering amount roughly. Then, we apply the supplementation/reduction rule and determine the amount of each item to order and the number of carriers. For the model, the GA was applied in order to obtain a better solution than deciding the ordering quantity of each item individually, as the number of trucks, the number of shortage of items or the number of storage of items.

The proposed mathematical formulation model for deterministic demand includes multiple objective functions and a large number of constraints. Therefore, it is difficult to solve this problem using the standard mathematical programming approaches.

In order to solve such a complex multi-item inventory problem, we used a GA approach. The GA is an adaptive heuristic search algorithm premised on the evolutionary concept of natural selection. The basic concept of the GA is to simulate processes in a natural system that are necessary for evolution, specifically the processes that follow the principles of survival of the fittest first laid down by Charles Darwin. As such, these processes represent an intelligent exploitation of a random search within a defined search space for the purpose of resolving a problem. First pioneered by John Holland in 1975, GAs have been widely investigated, experimented, and applied in several fields of engineering, e.g., cell formation (Brown and Sumichrast, 2001), job-shop scheduling problems (Watanabe *et al.*, 2005), and supply chain network design (Altiparmak *et al.*, 2006). The GA has also been used to solve combinatorial problems that cannot be handled by exhaustive, multi-objective, or exact methods due to their prohibitive complexity.

The basic concept of the GA approach is to code the decision variables of the problem as a finite length array (referred to as a chromosome) and calculate the fitness (objective function) of each string. Based on the fitness value, the probability of survival of each chromosome is defined. The surviving chromosomes then reproduce and form the chromosomes for the next generation through the crossover and mutation processes. Figure 1 shows the procedure of the GA.

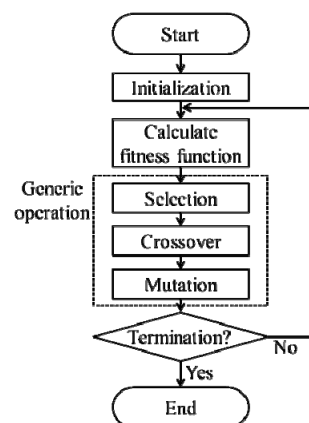


Figure 1. Basic procedure of the genetic algorithm.

3.1 Chromosome Representation

A chromosome is a set of genes related to the solution of the problem. A critical point when applying GAs to an optimization problem is to find a suitable encoding scheme that transforms feasible solutions into representations and then decode these representations.

We assume the ordering quantities of each item are roughly decided, and in this paper we apply an ordering policy for deciding the ordering quantity of each item.

In the present study, the rule for supplementing and reducing items from an order can be represented as a $4n$ -length chromosome, as shown in Table 2. There are four decision variables for each item: the supplementation limit, the reduction limit, the supplementation step, and the reduction step. The length of the chromosome is equal to the number of decision variables multiplied by the number of items.

An overview of the decoding routine is shown in Figures 2 and 3.

Next, we explain how items are supplemented to an order. An overview of this procedure is shown in Figure 2. We consider three items: A, B, and C. The values of some genes are changed from initial genes at each ordering. But we will not use the changed gene at the next ordering; instead, we use the gene, which is generated in each generation, at each ordering. So, we first copy the lot volume, supplementation limit chromosome, and supplementation step chromosome of each item in each ordering. In the initial phase, the remaining truck volume is 25. In each step, we search for the largest number appearing in the supplementation limit chromosome, and by using that chromosome, then repeat the following procedure, which is explained by example. In Step 1, item B is the largest gene in the supplementation limit chromosome, 30. Then, we recognize the supplementation step chromosome. In the supplementation step chromosome, the gene of item B is 10, and the remaining truck volume is 25. Item B becomes the candidate of supplementation. Because we could supplement quantity until the amount of the supplementation step chromosome, we supplement item B by 10.

We then reduce the supplemented volume from the supplementation limit chromosome of item B. If the remaining volume is less than the supplementation step gene, we supplement the maximum integer multiple of

Table 2. Example chromosome of the genetic representation scheme

	Supplementation limit chromosome			Reduction limit chromosome			Supplementation step chromosome			Reduction step chromosome		
Index	1	...	n	$n+1$...	$2n$	$2n+1$...	$3n$	$3n+1$...	$4n$
Item	1	...	n	1	...	n	1	...	n	1	...	n
Gene	$\triangle 17$...	$\triangle 10$	$\nabla 17$...	$\nabla 1$	$\triangle 4$...	$\triangle 3$	$\triangle 4$...	$\nabla 0$

lots that does not exceed the remaining truck volume and related supplementation step gene of the item, to the truck, as step 4 of item C. Steps are repeated and the items are varied until the remaining volume becomes zero.

Similarly, an overview of order reduction is shown in Figure 3. We consider three items: A, B, and C. First, we copy the lot volume, reduction limit chromosome, and reduction step chromosome of each item. In the initial phase, there are only a few items having excess volumes, and the remaining volume is 10. In each step, we search

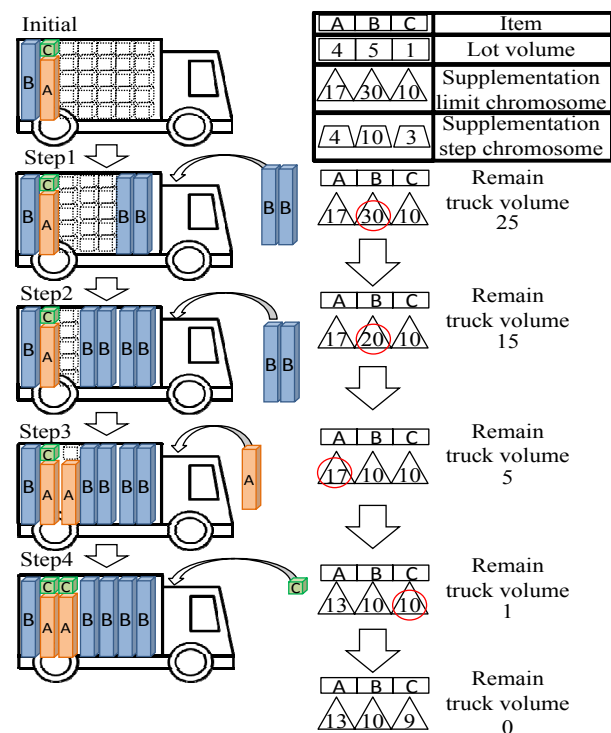


Figure 2. Basic procedure for supplementing an order.

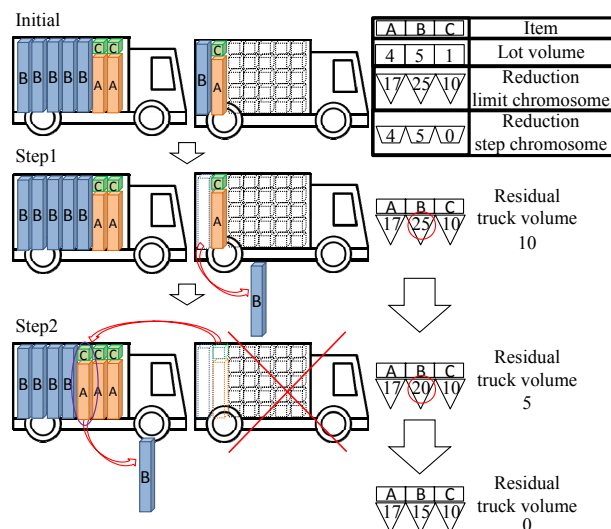


Figure 3. Basic procedure for order reduction.

the largest number in reduction volume chromosome, and repeat the following procedure. For example, in step 1, at 25, item B is the largest gene in the volume reduction chromosome. Then, we check the reduction step chromosome. In the reduction step chromosome, the gene of item B is 5, so item B becomes the candidate of reduction. Because we planned ordering item B to be more than 5, we reduce the item B 5.

If the residual volume is less than the reduction step gene, the integer multiple of lots, which does not exceed the gene of the related reduction step chromosome and the residual truck volume, are reduced. We then reduce the reduction volume from the reduction limit chromosome of item B. The steps are repeated until the remaining volume becomes zero.

3.2 Initial Population

During the initialization process, a predefined number of chromosomes are created, representing the supplementation limit, the reduction limit, the supplementation step, and the reduction step for all items.

In the initialization phase, we generate three types of random chromosomes. First, the supplementation limit and supplementation step chromosomes are set randomly, and the reduction limit and reduction step chromosomes are set to zero. These chromosomes represent the rule for supplementing items order only. Next, the supplementation limit and supplementation step chromosomes are set to zero, and the reduction limit and reduction step chromosomes are set randomly. These chromosomes represent the rule for reducing items order. Finally, the genes are set randomly, covering the entire range of possible solutions. We can set all of the chromosomes to zero, but the chromosomes represent the rule for using prescheduled ordering policy besides the remaining sufficient truck capacity.

3.3 Evaluation of the Fitness Function

A fitness function is a particular type of objective function that prescribes the optimality of a chromosome. Optimal chromosomes, or at least chromosomes with near-optimal values are allowed to breed and mix their datasets by several techniques to produce a new generation with improved characteristics.

In the present study, the fitness function is obtained by the original objective function. The fitness function of the chromosome, f (chromosome), is represented as follows:

$$f(\text{chromosome}) = \sum_{e=1}^3 \left(\frac{f_e(\text{chromosome}) - f_e^{\min}}{f_e^{\max} - f_e^{\min}} \right),$$

where f_e (chromosome) denotes each objective value

represented in Eqs. (1)-(3). These values are the results of the simulation when the chromosome rule is applied with the prescheduled ordering policy. Here, f_e^{\max} describes maximum value in the generation of the objective function e calculated by the simulation results. And, f_e^{\min} describes minimum value in the generation of the objective function e calculated by the simulation results. We calculated the fitness value at each generation for improving solution

3.4 Genetic Operations

Once initialization and fitness calculation have been performed, the genetic operations, which include selection, crossover, and mutation, are started. This is repeated until the termination conditions are satisfied. At the end of the each operation, the new generation and chromosomes are obtained from the following process. The new chromosome will be included with the chromosomes for the next iteration, which will eventually yield the optimal individual.

3.4.1 Selection

The selection operator selects chromosomes in the population for producing the next generation. Well-fitting chromosomes are likely to be selected for the next generation. There are several selection methods, such as roulette wheel selection, tournament selection, and random selection. In the present study, we use the roulette wheel selection method, in which the probability of being selected is directly proportional to the fitness of the chromosome.

3.4.2 Crossover

The crossover operator roughly mimics the biological recombination between two single-chromosome organisms. In the present study, we use two crossover methods, single-point crossover and average crossover.

The crossover operation performed in the present analysis is shown in Figure 4. First, we select two parent chromosomes, as indicated in Figure 4, by the red dashed line (---) and the blue double line (===). In the single-point crossover, genes are randomly selected and subsequences are exchanged left side and right side so that the genes between the two chromosomes create two children, as indicated by the red and blue genes in Figure 4. In average crossover, the average of each gene of two chromosomes is taken to create one child, as indicated by the purple double-dashed line (==). Children retain one section of the chromosome from each parent.

The point of crossover at which the chromosome is broken depends on the randomly selected crossover point. Often, both child 1 and child 2 are created and placed into the new population. However, crossover does not always occur. Sometimes, based on a set probability, no crossover

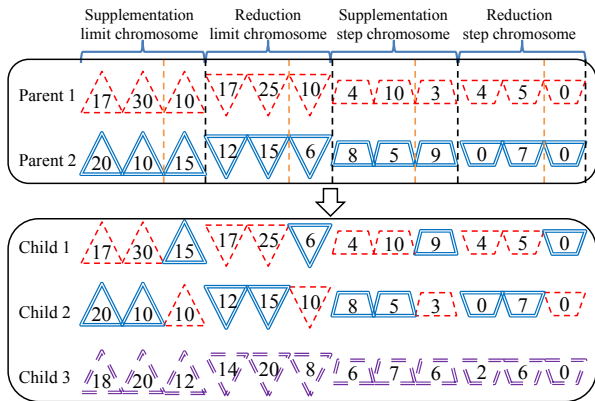


Figure 4. Basic procedure of the crossover operation.

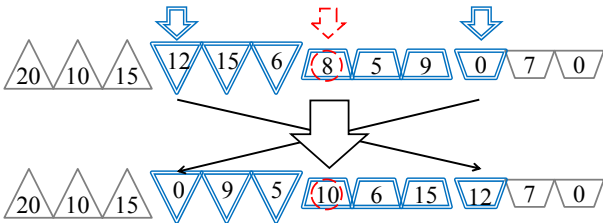


Figure 5. Basic procedure of the mutation operation.

occurs, and the parents are copied directly to the new population. The probability of crossover occurring is usually 40% to 90%.

3.4.3 Mutation

The mutation operator is used to maintain the genetic diversity from one generation of a population of chromosomes to the next. Mutation should allow the algorithm to avoid local optima by preventing the population of chromosomes from becoming too similar to each other. The mutation operator is applied to each child solution resulting from the crossover operation and is usually defined as a change in the number of genes in a chromosome. In the present study, as shown in Figure 5, the mutation operator operates by swapping two randomly selected genes, indicated by the blue double line (==), and by randomly changing one randomly selected gene, indicated by the red dashed line (- -). Mutation can occur at each chromosome with some small probability, usually between 0.0001 and 0.1.

4. EXPERIMENT AND RESULTS

Next, we apply the GA to actual shipping data. We assume that an ordering policy was applied to all of the items. We apply the proposed supplementation/reduction rule GA and calculate the fitness values from the simulation results.

4.1 Order Policy and Experiment

The ordering policy used in the present study is referred to as the (R, S) policy. There are two reasons for selecting this ordering policy. First, this policy is commonly used for inventory control. Second, the company that provided the actual shipment data used in the present study set the parameters and applied the ordering policy. Therefore, we focused on this policy.

In the present study, we use the (R, S) policy. The ordering policy is the special case of (R, s, S) policy. For the (R, s, S) policy, at the end of each time period of length R , if the inventory position decreases to or below the reorder point s , an amount equal to the difference between the order-up-to level S and the current inventory position is ordered. In the special case in which $s = S - 1$, we refer to this as the (R, S) policy.

In the present study, we apply the (R, S) policy with supplementation/reduction rule optimized in GA. In particular, the values of S of the i -th item in the t -th period are denoted by S_t^i and are calculated as follows:

$$S_t^i = (LT + R) \times \bar{d}_t^i + k \times ss_t^i \times \sqrt{LT + R} \quad \forall i \in I, \forall t \in T \quad (7)$$

$$\bar{d}_t^i = \frac{\sum_{l=t-SP}^{t-1} d_l^i}{SP} \quad \forall i \in I, \forall t \in T \quad (8)$$

$$ss_t^i = \sqrt{\frac{\sum_{l=t-SP}^{t-1} (d_l^i - \bar{d}_t^i)^2}{SP}} \quad \forall i \in I, \forall t \in T \quad (9)$$

$$S_t^i - I_t^i - \sum_{r=t-LT}^{t-1} x_r^i \leq x_t^i + M(1 - w_t^i) \quad \forall i \in I, \forall t \in T \quad (10)$$

$$x_t^i \leq Mw_t^i \quad \forall i \in I, \forall t \in T \quad (11)$$

$$w_t^i \in \{0, 1\}$$

Decision variables

$$w_t^i \begin{cases} 1 & : \text{if } x_t^i > 0 \\ 0 & : \text{otherwise} \end{cases}$$

Parameter

S_t^i : order-up-to level of item i on the t^{th} day

k : safety coefficient

\bar{d}_t^i : shipment average of item i on the t^{th} day

R : periodic review term

SP : sampling period

Eq. (7) is the formula for replenishment level S_t^i , which is calculated based on the shipment average and safety stock. Eqs. (8) and (9) detail the notation in Eq. (7). Eqs. (10) and (11) are the constraints on ordering, the

ordered amount, inventory level, and received amount. In each ordering, the rough ordering quantities of each item are calculated by the (R, S) policy, and, for improving the ordering, the supplementation/reduction rule is applied.

Precise management requires the ordering policy capable of shifting for reasons such as demand forecasting or seasonality. However, since we would like to propose a rule for order supplementation/reduction that is related to items and ordering policy decision, we did not shift ordering policies in this example.

4.2 Data Set

In numerical and simulation experiments, we used the actual data for the shipments of a distribution company for a one-year period. This distribution center orders items from several suppliers. Specifically, the distribution center ordered 200 items from one supplier. These items are categorized broadly into seven varieties, for example, beer, Japanese sake, and mineral water. In these categories, there are no obvious characteristic trends. Therefore, we applied the proposed method to the supplier and the 200 related items.

We simulated the (R, S) policies from the shipping data, which were used to calculate the shipping statistics. We set the parameters as described below. The safety coefficient, k , was set to 1.96. The sampling period, SP , was set to one month, and the lead time, LT , i.e., the time from when the items were ordered to when the items were delivered, was set to two days.

The GA parameters were set as follows. The probability of crossover occurring is usually 80%. The probability for mutation is 5%. The maximum number of generations is 1,000. The number of chromosomes in each generation is 100. We used roulette selection for selection in each generation.

4.3 Evaluation Criteria

We compared the fitness values of every chromosome in each generation. A fitness function considers the objective values, which are represented in Eqs. (1)–(3).

We used the fitness values explained at Section 3.3 in GA for improving solutions. On the other hand, for checking the convergence of the GA, we use the other fitness value as follows:

$$f(\text{chromosome}) = \sum_{e=1}^3 \left(\frac{f_e(\text{chromosome})}{f_e(\mathbf{0})} \right)$$

Here, $f_e(\mathbf{0})$ describes the simulation results for a non-modified, no-supplemented-items, no-items-removed order rule using a replenish policy for each item. In the present study, a lower fitness value represents a good solution.

Each of the objective values was calculated in the simulation.

In the simulation, the first month of shipping data was used to calculate the initial values of S_i^j . The S_i^j values for the (R, S) policies were recalculated periodically.

4.4 Experimental Results

The results of the proposed model are shown in Figures 6–8. Figure 6 shows the Pareto solution for inventory vs. shortage. Figure 7 shows the Pareto solution for inventory vs. truck. Figure 8 shows the Pareto solution for shortage vs. truck. We colored the scatter chart by the number of trucks used, as in Figure 6. The results for non-modified orders, in which all genes are zero, are plotted by a purple triangle in each figure.

The results of non-modified ordering for inventories, shortage, and the number of trucks used were 197,124, 5,087, and 154, respectively. Then, in the proposed method the best results, i.e., 196,207, 5,043, and 153, were obtained in the 50th generation. The fitness was improved until 2.63 (249,815, 3,539, 104) and fitness values were not improved drastically after 1,000 generations, with 100 chromosomes in each generation.

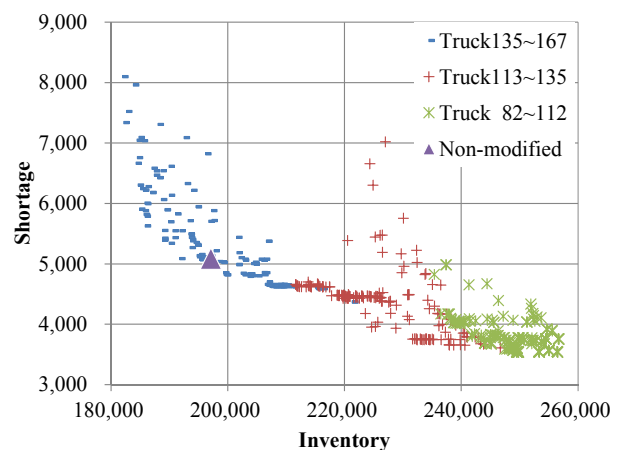


Figure 6. Pareto solution of inventory vs. shortage.

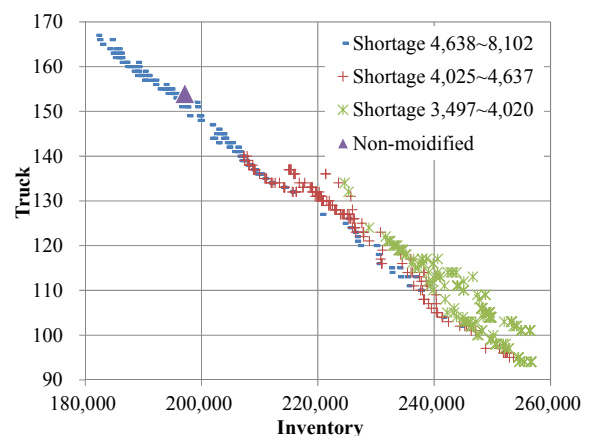


Figure 7. Pareto solution of inventory vs. truck.

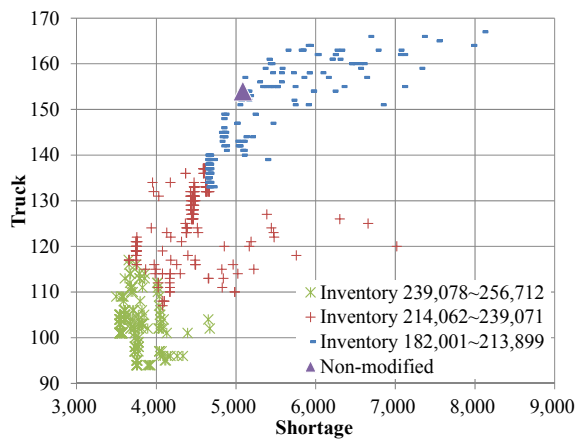


Figure 8. Pareto solution of truck vs. shortage.

From Figure 6, the number of inventories and the number of shortage of items are in inverse proportion. This result is well known in the inventory control. And, when the number of trucks of the solution is small, the number of inventory becomes large in many solutions. In those solutions, it adds items in order to reduce the number of trucks. On the other hand, when the number of trucks increases, the variation of the number of inventory and shortage becomes scattering because the number of the rules applied increases.

From Figure 7, the number of inventory and the number of trucks are in inverse proportion. When the number of trucks of the solution is small, the number of inventory becomes large. In those solutions, it adds items in order to reduce the number of trucks.

From Figure 8, the number of shortage and the number of truck were proportional. When we increase the number of trucks, the proposal rule is applied more often. Therefore, when the set rule is a reduction rule, orders were placed in a smaller quantity frequently, and when the set rule is a supplementation rule, orders were placed in a large quantity occasionally. And, these solutions remain to the Pareto solution easily.

Figure 9 shows the convergence of the fitness value, explained in Section 4.3. We calculated the fitness value

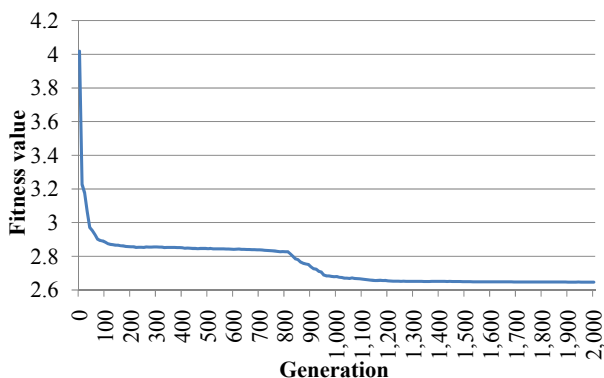


Figure 9. Convergence of the fitness value.

and plotted the average of the top 50 fitness values. We examined the fitness values until the 2,000th generation. However, the average of the top 50 fitness values did not improve dramatically after the 1,000th generation.

Based on these results, under the conditions considered herein, by reducing the number of trucks used for shipping, the shortage of items also decreased. There are several reasons for this. The number of trucks is reduced by front-loaded ordering, which increases the amount of inventory in storage. However, as a result, the additional inventory acts as a reserve stock and reduces the number of items in shortage. Although a balance must be found for the inventory, the number of trucks used in shipping was sharply reduced.

5. CONCLUSIONS

In the present paper, a multi-item inventory problem with capacity of replenishing carriers was considered. The objective is to determine which items should we supplement to or remove from orders, in order to minimize the number of items in storage, the number of out-of-stock items, and the number of carriers used.

Since the objective functions of the mathematical model are multi-objective functions, we proposed GAs to be used with a prescheduled ordering policy to provide a better solution. Actual shipment data are used to verify the performance of the proposed GA.

In a numerical experiment, based on actual shipment data, we simulated the replenishment of items in a coordinated manner. Because some rules in our numerical experiments outperformed the results of applying ordering policies individually, the simulation study indicated that applying the conventional ordering policies individually might not provide effective inventory management. Moreover, we proposed several supplementation/reduction rules for inventory management.

Further study is required for checking the accuracy of proposed GA. We should formalize the ordering policy with supplementation/reduction rules and compare optimized solution, when we weighted each objective function and use mathematical programming solver. And, we should assess the performance of the GA with other meta-heuristics (simulated annealing, Tabu search, and ant colony optimization). Furthermore, the relationship between the item characteristics and the supplementation/reduction ordering rule should be also considered.

REFERENCES

- Altıparmak, F., Gen, M., Lin, L., and Paksoy, T. (2006), A genetic algorithm approach for multi-objective optimization of supply chain networks, *Computers*

- & *Industrial Engineering*, **51**(1), 196-215.
- Berling, P. and Marklund, J. (2006), Heuristic coordination of decentralized inventory systems using induced backorder costs, *Production and Operations Management*, **15**(2), 294-310.
- Bijvank, M. and Vis, I. F. A. (2011), Lost-sales inventory theory: a review, *European Journal of Operational Research*, **215**(1), 1-13.
- Brown, E. C. and Sumichrast, R. T. (2001), CF-GGA: a grouping genetic algorithm for the cell formation problem, *International Journal of Production Research*, **39**(16), 3651-3669.
- Chan, L. M. A., Muriel, A., Shen, Z. J. M., Simchi-Levi, D., and Teo, C. P. (2002), Effective zero-inventory-ordering policies for the single-warehouse multi retailer problem with piecewise linear cost structures, *Management Science*, **48**(11), 1446-1460.
- Gruen, T. W., Corsten, D. S., and Bharadwaj, S. (2002), Retain out-of-stocks: a worldwide examination of extent, causes and consumer responses, Grocery Manufacturers of American, Washington, DC.
- Huh, W. T., Janakiraman, G., Muckstadt, J. A., and Rusmevichientong, P. (2009), Asymptotic optimality of order-up-to policies in lost sales inventory systems, *Management Science*, **55**(3), 404-420.
- Janakiraman, G., Seshadri, S., and Shanthikumar, J. G. (2007), A comparison of the optimal costs of two canonical inventory systems, *Operations Research*, **55**(5), 866-875.
- Kiesmüller, G. P. (2009), A multi-item periodic replenishment policy with full truckloads, *International Journal of Production Economics*, **118**(1), 275-281.
- Kiesmüller, G. P. (2010), Multi-item inventory control with full truckloads: a comparison of aggregate and individual order triggering, *European Journal of Operational Research*, **200**(1), 54-62.
- Levi, R., Janakiraman, G., and Nagarajan, M. (2008), A 2-approximation algorithm for stochastic inventory control models with lost sales, *Mathematics of Operations Research*, **33**(2), 351-374.
- Liao, T. W. and Chang, P. C. (2010), Impacts of forecast, inventory policy, and lead time on supply chain inventory: a numerical study, *International Journal of Production Economics*, **128**(2), 527-537.
- Tiacci, L. and Sietta, S. (2009), An approach to evaluate the impact of interaction between demand forecasting method and stock control policy on the inventory system performances, *International Journal of Production Economics*, **118**(1), 63-71.
- Van Donselaar, K. H. and Broekmeulen, R. A. C. M. (2013), Determination of safety stocks in a lost sales inventory system with periodic review, positive lead-time, lot-sizing and a target fill rate, *International Journal of Production Economics*, **143**(2), 440-448.
- Watanabe, M., Ida, K., and Gen, M. (2005), A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem, *Computers & Industrial Engineering*, **48**(4), 743-752.
- Xiao, Y., Zhang, R., and Kaku, I. (2011), A new approach of inventory classification based on loss profit, *Expert Systems with Applications*, **38**(8), 9382-9391.
- Yang, W., Chan, F. T. S., and Kumar, V. (2012), Optimizing replenishment policies using genetic algorithm for single-warehouse multi-retailer system, *Expert Systems with Applications*, **39**(3), 3081-3086.
- Zipkin, P. (2008), Old and new methods for lost-sales inventory systems, *Operations Research*, **56**(5), 1256-1263.