Applying Genetic Algorithms for Inventory Lot-Sizing Problem with Supplier Selection under Storage Capacity Constraints

Chirawat Woarawichai[1], Kritsana Kuruvit[1], Paitoon Vashirawongpinyo[2]

[1] Department of Industrial Engineering, Faculty of Engineering, Rajamangala University of Technology Lanna Tak, Thailand

[2] Department of Industrial Engineering, Faculty of Engineering, Siam University Bangkae Bangkok, Thailand

Abstract

The objective of this research is to calculate the optimal inventory lot-sizing for each supplier and minimize the total inventory cost which includes joint purchase cost of the products, transaction cost for the suppliers, and holding cost for remaining inventory. Genetic algorithms are applied to the multi-product and multi-period inventory lot-sizing problems with supplier selection under storage capacity. Also a maximum storage capacity for the decision maker in each period is considered. The decision maker needs to determine what products to order in what quantities with which suppliers in which periods. It is assumed that demand of multiple products is known over a planning horizon. The problem is formulated as a mixed integer programming and is solved with the Genetic algorithms. The detailed computation results are presented.
Keywords: *Genetic Algorithms, Supplier selection, Storage capacity.*

1. Introduction

Lot-sizing problems are production planning problems with the objective of determining the periods where production should take place and the quantities to be produced in order to satisfy demand while minimizing production and inventory costs [1]. Since lot-sizing decisions are critical to the efficiency of production and inventory systems, it is very important to determine the right lot- sizes in order to minimize the overall cost.

Lot-sizing problems have attracted the attention of researchers. The multi-period inventory lot-sizing scenario with a single product was introduced by Wagner and Whitin [2], where a dynamic programming solution algorithm was proposed to obtain feasible solutions to the problem. Soon afterwards, Basnet and Leung [3] developed the multi-period inventory lot-sizing scenario which involves multiple products and multiple suppliers. The model used in these former research works is formed by a single-level unconstrained resources indicating the type, amount, suppliers and purchasing time of the product. This model is not able to consider the capacity limitations. One of the important modifications we consider in this paper is that of introducing storage capacity. One of the important modifications we consider in this paper is that of introducing storage capacity constraints. With the advent of supply chain management, much attention is now devoted to supplier selection. Rosenthal et al. [4] study a purchasing problem where suppliers offer discounts when a "bundle" of products is bought from them, and one needs to select suppliers for multiple products. Then a mixed integer programming formulation is presented. Jayaraman et al. [5] proposed a supplier selection model that considers quality, production capacity, lead-time, and storage capacity limits.

In this paper based on Basnet and Leung [3] genetic algorithms (GAs) are applied to the multi-product and multi-period inventory lot-sizing problem with supplier selection under storage space. Also a maximum storage space for the decision maker in each period is considered. The decision maker needs to determine what products to order in what quantities with which suppliers in which periods. The objective of this research is to calculate the optimal inventory lot-sizing for each supplier and minimize the total inventory cost.

This paper is organized as follows: Section 2 the genetic algorithm approach is applied to problem. Section 3 we describe our model. In Section 4 presents a numerical example of the model. Finally, computation results and conclusions are presented in Section 5 and 6.

2. Methodology

The genetic algorithms (GAs) approach is developed to find optimal (or near – optimal) solution. Detail discussion on GAs can be found in Holland [6], Michalewicz [7], and Gen and Cheng [8] [9]. In this section, we explain GAs procedure is illustrated in Fig. 1 to start the search GAs are initialized with a population of individuals. The individuals are encoded as chromosomes in the search space. GAs use mainly two operators namely, crossover and mutation to direct the population to the global optimum. Crossover allows exchanging information between different solutions (chromosomes) and mutation increases the variety in the population. After the selection and evaluation of the initial population, chromosomes are selected on which the crossover and mutation operators are applied. Next the new population is formed. This process is continued until a termination criterion is met [1].

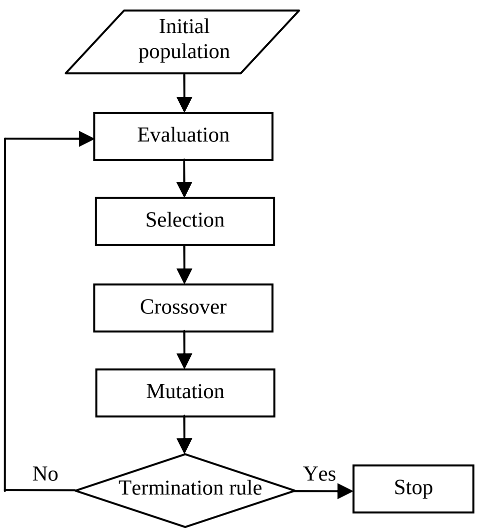

Fig. 1 The genetic algorithm procedure

3. Formulation

We also make the following assumptions and mathematical for the model:

3.1 Assumptions

- Demand of products in period is known over a planning horizon.
- All requirements must be fulfilled in the period in which they occur: shortage or backordering is not allowed.
- Transaction cost is supplier dependent, but does not depend on the variety and quantity of products involved.
- Holding cost of product per period is product-dependent.
- Initial inventory of the first period and the inventory at the end of the last period are assumed to be zero.
- Product needs a storage space and available total storage space is limited.

Base on the above assumption of model, Fig 2 shows the behavior of the model considering the scenario of multi-period inventory lot-sizing problem with supplier selection under storage space and budget constraints.

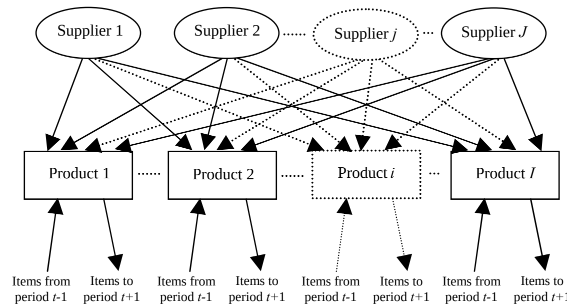

Fig. 2 Behavior of the model in period t .

3.2 Mathematical modeling

This paper is built upon Basnet and Leung [3] model. We formulate the multi-product and multi-period inventory lot-sizing problem with supplier selection under storage space and budget constraints using the following notation:

Indices:

i = $1,\ldots,I$ index of products
j = $1,\ldots,J$ index of suppliers
t = $1,\ldots,T$ index of time periods

Parameters:

D_{it} = demand of product i in period t
P_{ij} = purchase price of product i from supplier j
H_i = holding cost of product i per period
O_j = transaction cost for supplier j
w_i = storage space product i
S = total storage capacity

Decision variables:

X_{ijt} = number of product i ordered from supplier j in period t
Y_{jt} = 1 if an order is placed on supplier j in time period t , 0 otherwise

Intermediate variable:

R_{it} = Inventory of product i , carried over from period t to period $t+1$

Regarding the above notation, the mixed integer programming is formulated as follows:

$$Minimize\ (TC) = \sum_i \sum_j \sum_t P_{ij} X_{ijt} + \sum_j \sum_t O_j Y_{jt} + \sum_i \sum_t H_i \left(\sum_{k=1}^{t} \sum_j X_{ijk} - \sum_{k=1}^{t} D_{ik} \right) \quad (1)$$

Subject to:

$$R_{it} = \sum_{k=1}^{t} \sum_j X_{ijk} - \sum_{k=1}^{t} D_{ik} \geq 0 \quad \text{for all } i \text{ and } t, \quad (2)$$

$$\left(\sum_{k=t}^{T} D_{ik} \right) Y_{jt} - X_{ijt} \geq 0 \quad \text{for all } i,\ j, \text{ and } t, \quad (3)$$

$$\sum_i w_i \left(\sum_{k=1}^{t} \sum_j X_{ijk} - \sum_{k=1}^{t} D_{ik} \right) \leq S \quad \text{for all } t, \quad (4)$$

$$Y_{jt} = 0 \text{ or } 1 \quad \text{for all } j \text{ and } t, \quad (5)$$

$$X_{ijt} \geq 0 \quad \text{for all } i,\ j, \text{ and } t, \quad (6)$$

The objective function as shown in (1) consists of three parts: the total cost (TC) of 1) purchase cost of the products, 2) transaction cost for the suppliers, and 3) holding cost for remaining inventory in each period in t+ 1. Constraint in (2) all requirements must be filled in the period in which they occur: shortage or backordering is not allowed. Constraint in (3) there is not an order without charging an appropriate transaction cost. Constraint in (4) each products have limited capacity. Constraint in (5) is binary variable 0 or 1 and Constraint in (6) is non-negativity restrictions on the decision variable. According to a large optimal problem, a GAs approach is applied to solve this problem.

4. A numerical example

In this section we solved a numerical example of the model using the LINGO. We consider a scenario with three products over a planning horizon of five periods whose requirements are as follows: demands of three products over a planning horizon of five periods are given in Table 1. There are three suppliers and their prices and transaction cost, holding cost and storage space are show in Table 2 and Table 3, respectively.

Table 1: Demands of three products over a planning horizon of five periods (D_{it}) and Budget of them (B_t) .

	Planning Horizon (Five Periods)				
Products	1	2	3	4	5
A	12	15	17	20	13
B	20	21	22	23	24
C	20	19	18	17	16

Table 2: Price of three products by each of three suppliers X, Y, Z (P_{ij}) and transaction cost of them (O_j) .

	Price		
Products	X	Y	Z
A	30	33	32
B	32	35	30
C	45	43	45
Transaction Cost	110	80	102

Table 3: Holding cost of three products A, B, C (H_i) and storage space of them (w_i) .

	Products		
	A	B	C
Holding Cost	1	2	3
Storage Space	10	40	50

The total storage capacity (S) is equal to 200.

The results of applying the proposed method are shown in Table 4. The solution of this problem (I = 3, J = 3, and T = 5) is to place the following orders.
All other $X_{ijt} = 0$:

Table 4: Order of three products over a planning horizon of five periods (X_{ijt}) .

	Planning Horizon (Five Periods)				
Products	1	2	3	4	5
A	X_{111} = 12	X_{132} = 15	X_{113} = 37	-	X_{135} = 13
B	X_{231} = 20	X_{232} = 21	X_{213} = 22	X_{234} = 23	X_{235} = 24
C	X_{321} = 20	X_{332} = 19	X_{313} = 18	X_{334} = 17	X_{335} = 16

Cost calculation for this solution:
Purchase cost for product 1 from supplier 1, 3
= (37×30) + (12+15+13) × 32 = 2,390.
Purchase cost for product 2 from supplier 1, 3
= (22×32) + (20+21+23+24) × 30 = 3,344.
Purchase cost for product 3 from supplier 1, 3
= (18×45) + (20+19+17+16) × 45 = 4,050.
Transaction cost from supplier 1, 3
= (1×110) + (4×102) = 518.
Holding cost for product 1
$R_{13} = X_{113} - D_{13} = 37 - 17 = 20.$
$= H_1 \sum R_{1t} = 1\times (0 + 0 + 20 + 0 + 0) = 20.$
Thus, the total cost for this solution
= 2,390 + 3,344 + 4,050 + 518 + 20
= 10,322.

5. Computation results

In this section the comparison of the two methods solved problem size is using a commercially available optimization package like LINGO and GAs code is developed in MATLAB. Experiments are conducted on a personal computer equipped with an Intel Core 2 duo 2.00 GHz, CPU speeds, and 1 GB of RAM. The transaction costs are generated from *int* [50; 200], a uniform integer distribution including 50 and 200. The prices are from *int* [20; 50], the holding costs from *int* [1; 5], the storage space from *int* [10; 50], and the demands are from *int* [10; 200].

The result in Table. 5 shows the GAs comparing with LINGO for the nine problem sizes. A problem size of *I; J; T* indicates number of suppliers = *I*, number of products = *J*, and number of periods = *T*. Computation time limit is set at 120 minutes. For comparison, the percentage error is calculated by (7) and (8).

Percentage error of LINGO

$$= \left[\frac{\text{Upper bound - Lower bound}}{\text{Upper bound}}\right] \times 100 \qquad (7)$$

Percentage error of GAs

$$= \left[\frac{\text{Upper bound LINGO - GAs}}{\text{Upper bound LINGO}}\right] \times 100 \qquad (8)$$

The solution time of LINGO to optimal is a short time as the small problem size (with the problem sizes 3 x 3 x 5; 3 x 3 x 10; 3 x 3 x 15; and 4 x 4 x 10). For large problems sizes LINGO cannot obtain optimal solutions within limit time due to as the larger problem size (with the problem sizes 4 x 4 x 15; 5 x 5 x 20; 10 x 10 x 50; 10 x 10 x 80; and 15 x 15 x 50). The GAs can optimally solve when the problem size is small (with the problem sizes 3 x 3 x 5; 3 x 3 x 10; 3 x 3 x 15; 4 x 4 x 10; 4 x 4 x 15; 5 x 5 x 20; and 10 x 10 x 50). There are two problems which GAs cannot obtain optimal solutions (with the problem sizes 10 x 10 x 80; and 15 x 15 x 50). Next, we study differences in the problem sizes between solutions from the optimization with LINGO and the GAs. The results are show in Fig. 4, a plot of the problem size versus solution time. LINGO uses longer computation time more than GAs with seven problem sizes, but uses equal time with two problem sizes. As show in Fig. 5 a plot of the problem size versus % error when the problem size is very large, LINGO used a maximum % error from the optimal solutions is found to be 4.41% (at the problem size 10 x 10 x 80) which has more % error than GAs. The GAs can solve small % error of two problem sizes (at the problem size 10 x 10 x 80; and 15 x 15 x 50). Fig. 6 and Fig. 7 show compares result between LINGO and GAs in problem size 3 x 3 x 5. Thus, it is evident that GAs is an effective means for solving the problem. GAs solution is optimal when the problem size is small. For larger problems GAs can find feasible solution within time limit for which LINGO fails to find the optimum.

Table 5: Comparative results of the two methods

Problem size	Optimization approach with LINGO			Genetic Algorithms (GAs)		
	Total cost	Solution time (minute)	% Error	Total cost	Solution time (minute)	% Error
3 x 3 x 5	10,322	0.01	0	10,322	0.02	0
3 x 3 x 10	20,644	0.14	0	20,644	0.21	0
3 x 3 x 15	30,966	14.35	0	30,966	1.45	0
4 x 4 x 10	25,436	6.34	0	25,436	0.51	0
4 x 4 x 15	38,154[a] , 37,828[b]	120	0.85	38,154	2.47	0
5 x 5 x 20	60,218[a] , 59,527[b]	120	1.14	60,200	3.36	0.03
10 x 10 x 50	285,344[a] , 274,758[b]	120	3.70	284,940	108.50	0.14
10 x 10 x 80	456,494[a] , 436,317[b]	120	4.41	455,904	120	0.12
15 x 15 x 50	417,800[a] , 405,155[b]	120	2.66	416,473	120	0.31

[a]LINGO12 = Upper bound, [b]LINGO12 = Lower bound.

However, the GAs provides superior solutions to those from LINGO that are close to optimum in a very short time, and thus appears quite suitable for realistically sized problems.
Additionally, the computation time when using GAs is also short, making it a very practical means for solving the multiple products and multi-period inventory lot-sizing problem with supplier selection under storage capacity.

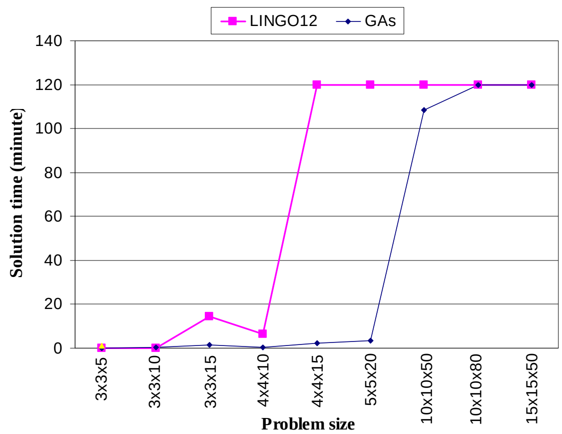

Fig. 4 Plot of the problem size vs. solution time (minute)

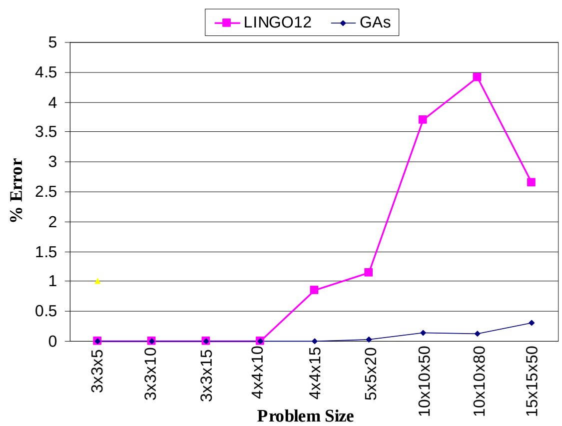

Fig. 5 Plot of the problem size vs. % error

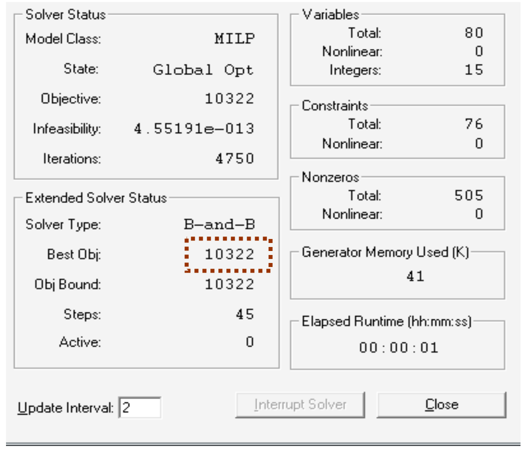

Fig. 6 The best objective of LINGO

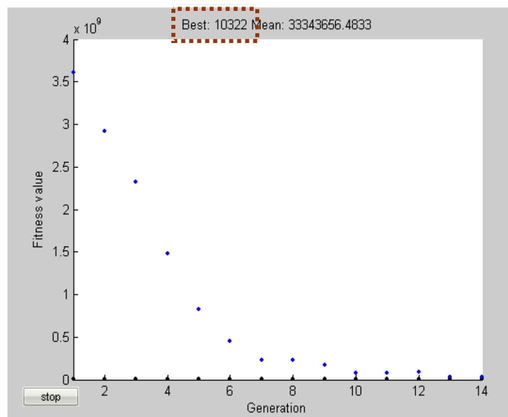

Fig. 7 The fitness value of GAs

6. Conclusions

In this paper, we present genetic algorithms (GAs) applied to the multi-product and multi-period inventory lot-sizing problem with supplier selection under storage capacity. Also a maximum storage space for the decision maker in each period is considered. The decision maker needs to determine what products to order in what quantities with which suppliers in which periods. The mathematical model is give and the use of the model is illustrated though a numerical example. The problem is formulated as a mixed integer programming and is solved with LINGO and the GAs. As compared to the solution of optimization package like LINGO, the GAs solutions are superior.

References

[1] G.H. Goren, S. Tunali, and R. Jans, “A review of applications of genetic algorithms in lot sizing”, Journal of Intelligent Manufacturing. Springer Netherlands. 2008.

[2] H. M. Wagner, and T. M. Whitin, “Dynamic version of the economic lot-size model”, Management Science. 5, pp, 89–96, 1958.

[3] C. Basnet and J. M. Y. Leung, “Inventory lot-sizing with supplier selection”, Computers and Operations Research. 32, pp, 1–14, 2005.

[4] E.C. Rosenthal, J.L. Zydiak, and S.S. Cauchy, “Vendor selection with bundling”, Decision Sciences. 26, 1995, 35–48.

[5] V. Jayaraman, R. Srivastava, and W.C. Benton, Supplier selection and order quantity allocation: a comprehensive model”, The Journal of Supply Chain Management. 35, 1999, 50–58.

[6] J. H. Holland, Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor. 1975.

[7] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. *AI Series*. Springer-Verlag, New York. 1944.

[8] M. Gen, and R. Cheng, Genetic Algorithms and Engineering Design. Wiley, New York. 1977.

[9] M. Gen, and R. Cheng, Genetic Algorithms and Engineering Optimization. Wiley, New York. 2000.