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

Chirawat Woarawichai[1*], Tarathorn Kullpattaranirun[2] and Vichai Rungreunganun[1]
1) Department of Industrial Engineering, Faculty of Engineering,King Mongkut's University of Technology North Bangkok, Bangkok 10800
2) Industrial Management Program Faculty of Business Administration Mahanakorn University of Technology Nong Chok, Bangkok 10530
**Corresponding Author : Chirawat-w@hotmail.com*

Abstract

Genetic algorithms are applied to the multiple products and multiple periods inventory lot-sizing problem with supplier selection under storage capacity. 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. 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 genetic algorithms. Finally, numerical example is provided to illustrate the solution procedure. The results determine what products to order in what quantities with which suppliers in which periods.

Keywords: genetic algorithms, inventory lot-sizing, 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.

The multiple periods 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 multiple periods inventory lot-sizing scenario which involves multiple products and suppliers.

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. Ganeshan [5] has presented a model for determining lot-sizes that involves multiple suppliers while considering multiple retailers, and consequent demand on a warehouse. Jayaraman et al. [6] 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 multiple products and multiple periods inventory lot-sizing problem with supplier selection under storage capacity. The objective of this research is to calculate the optimal inventory lot-sizing for each supplier and minimize the total inventory cost. The results determine what products to order in what quantities with which suppliers in which periods.

2. Methods

2.1 Genetic Algorithms Approach

The genetic algorithms (GAs) approach is developed to find optimal (or near – optimal) solution. Detail discussion on GAs can be found in Holland [7], Michalewicz [8], and Gen and Cheng[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].

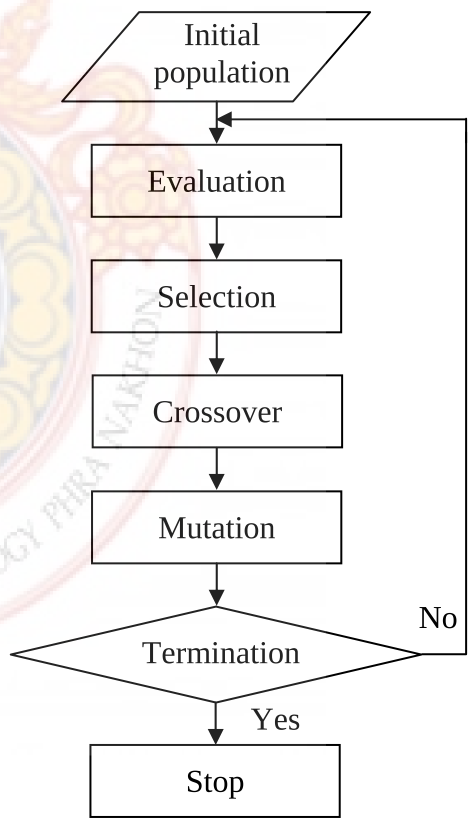

Fig.1 The genetic algorithms procedure

2.2 Mathematical Formulation

This paper is built upon Basnet and Leung's [3] work. We formulate the multi-product multi-period lot sizing with supplier selection under storage capacity problem using the following notations and assumptions:

Notations:

- Indices:

  $i = 1,\dots,I$ index of products.
  $j = 1,\dots,J$ index of suppliers.
  $t = 1,\dots,T$ 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.
  S_t = storage capacity in period t.

- 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 $t + 1$.

Assumptions:

- Demand of products in period is known over a planning horizon.
- Multi-product and multi-period.
- 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 is product-dependent.
- Product needs a storage space and available total storage capacity is limited.

Based on the above assumptions a mathematical model is developed. The behavior of the model is illustrated in Fig. 2

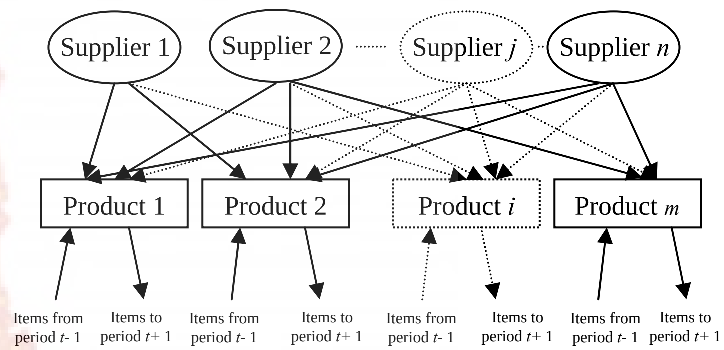

Fig. 2 Behavior of the model in period t.

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

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

Subject to:

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

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

$$\sum_{k=1}^{t} \sum_j X_{ijk} - \sum_{k=1}^{t} D_{ik} \leq S_t - \sum_i D_{it} \quad \text{for all } t, \tag{4}$$

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

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

The objective function consists of three parts: The total cost (TC) = 1) purchase cost of the products + 2) transaction cost for the suppliers + 3) holding cost for remaining inventory in each period.

Constraint 1 all requirements must be filled in the period in which they occur: shortage or backordering is not allowed. Constraint 2 there is not an order without charging an appropriate transaction cost. Constraint 3 indicates that the sum of the inventory level must be smaller than or equal to the bound inventory. Constraint 4 is binary 0, 1 and Constraint 5 is non-negativity restrictions on the decision variable.

2.3 Numerical Example

In this section we solved a numerical example of the model using real parameter genetic algorithms. 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:

Table 1 Demands of three products over a planning horizon of five periods. In each cell, indicates D_{it}

	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. In each cell, indicates P_{ij}

	Price		
Products	**X**	**Y**	**Z**
A	30	33	32
B	32	35	30
C	45	43	45

Table 3 Transaction cost of three suppliers X, Y, Z. In each cell, indicates O_j

Transaction Cost		
X	**Y**	**Z**
110	80	102

Table 4 Holding cost of three products A,B,C. In each cell, indicates H_i

Holding Cost		
A	**B**	**C**
1	2	3

Table 5 Storage capacity of three products over a planning horizon of five periods. In each cell, indicates S_t

Storage Capacity				
Planning Horizon (Five periods)				
1	**2**	**3**	**4**	**5**
80	80	70	60	100

The solution of this problem (i = 3, j = 3 and t = 5) is to place the following orders. All other X_{ijt} = 0:

Table 6 Order of three products over a planning horizon of five periods. In each cell, indicates X_{ijt}

	Order				
	Planning Horizon (Five periods)				
Products	**1**	**2**	**3**	**4**	**5**
A	X_{131} = 12	X_{122} = 15	X_{133} = 17	X_{134} = 20	X_{135} = 13
B	X_{231} = 41	-	X_{233} = 22	X_{234} = 23	X_{235} = 24
C	X_{331} = 20	X_{322} = 19	X_{333} = 18	X_{334} = 17	X_{335} = 16

- Cost calculation for this solution:

Purchase cost for product 1 from supplier 2
= 15 × 33 = 495
Purchase cost for product 1 from supplier 3
= (12+17+20+13) × 32 = 1,984
Purchase cost for product 2 from supplier 3
= (41+22+23+24) × 30 = 3,300
Purchase cost for product 3 from supplier 2
= 19 × 43 = 817
Purchase cost for product 3 from supplier 3
= (20+18+17+16) × 45 = 3,195

Number of orders from supplier 2 = 1 (in periods 2). Transaction cost = $1\times 80 = 80$

Number of orders from supplier 3 = 4 (in periods 1, 3, 4 and 5). Transaction cost = $4\times 102 = 408$

There are no orders from supplier 1.

Table 7 Carried-over inventory of three products over a planning horizon of five periods. In each cell, indicates R_{it}

	Carried-Over Inventory				
	Planning Horizon (Five periods)				
Products	**1**	**2**	**3**	**4**	**5**
A	0	0	0	0	0
B	21	0	0	0	0
C	0	0	0	0	0

The first entry represents R_{11}
$= X_{131} - D_{11} = 12 - 12 = 0$; etc.
Holding cost for product 1
$= H_1 \sum R_{1t} = 1 \times (0 + 0 + 0 + 0 + 0) = 0.$
Holding cost for product 2
$= H_2 \sum R_{2t} = 2 \times (21 + 0 + 0 + 0 + 0) = 42.$
Holding cost for product 3
$= H_3 \sum R_{3t} = 3 \times (0 + 0 + 0 + 0 + 0) = 0.$
Thus, the total cost for this solution:
purchase cost + transaction cost+ holding cost
= 495 + 1,984 + 3,300 + 817 + 3,195 + 408 + 80 + 42 = 10,321.

3. Results and discussion

Computational results

In this section the comparison of the two methods solved problem size is using a commercially available optimization package like LINGO12 and Genetic Algorithms (GAs) code is developed in MATLAB7. Experiments were executed on a personal computer equipped with a Pentium 4 processor working at a speed of 2.80 GHz. The transaction costs were generated from *int* [50; 200], a uniform integer distribution including 50 and 200. The prices were from *int*[20; 50], the holding costs from *int* [1; 5], the storage capacity from *int* [50; 4000], and the demands were from *int* [1; 200].The results are shown in Table 8, Table 9 and Table 10 ,where a problem size of l; m; n indicates number of suppliers = l, number of products = m, and number of periods = n.

The solution times of LINGO12 very slowly as the problem size increases. When LINGO12 is utilized, the optimal solution could be obtained only when the problem size is small. However, the GAs provides solutions that are close to optimum in a very short time, and thus appears quite suitable for realistically sized problems.

Table 8 Percentage error of LINGO12 [(Upper bound) – (Lower bound))/(Upper bound)] * 100

Problem size	% Error
3, 3, 5	0
3, 3, 10	0
3, 3, 15	0
4, 4, 10	0
4, 4, 15	0.98
5, 5, 20	2.08
10, 10, 50	3.27
15, 15, 100	2.63
20, 20, 100	1.19

Table 9 Percentage error of GAs [(Upper bound Lingo) – (GAs))/(Upper bound Lingo)] * 100

Problem size	% Error
3, 3, 5	0
3, 3, 10	0
3, 3, 15	0
4, 4, 10	0
4, 4, 15	-0.03
5, 5, 20	-5.41
10, 10, 50	6.61
15, 15, 100	-0.61
20, 20, 100	-1.52

Table 10 Comparative results

Problem size	Optimization approach with LINGO12		Genetic algorithms	
	Total cost	Solution time (minute)	Total cost	Solution time (minute)
3, 3, 5	10,321	0.02	10,321	0.25
3, 3, 10	20,635	1.2	20,635	0.60
3, 3, 15	30,949	95.5	30,949	1.25
4, 4, 10	25,374	17.5	25,374	1.30
4, 4, 15	38,068[a] , 37,693[b]	*	38,079	120
5, 5, 20	66,832[a] , 65,437[b]	*	70,450	120
10, 10, 50	312,488[a] , 302,243[b]	*	291,833	120
15, 15, 100	946,125[a] , 921,150[b]	*	951,920	120
20, 20, 100	1,256,210[a] , 1,232,110[b]	*	1,275,360	120

[a]LINGO = Upper bound, [b]LINGO = Lower bound. *Solutions could not be found within 2 hour.

4. Conclusions

In this paper, we present genetic algorithms (GAs) applied to the multiple products and multiple periods inventory lot-sizing problem with supplier selection under storage capacity. The problem is formulated as a mixed integer programming (MIP) and is solved with GAs which performed very well on the tested problems.

The results determine what products to order in what quantities with which suppliers in which periods.

5. Acknowledgements

The authors gratefully acknowledge the King Mongkut's University of Technology North Bangkok (KMUTNB). Out thanks go to Dr.Krisada Asawarungsaengkul help in programming and running some of the computational experiments.

6. References

[1] Goren, G.H., Tunali, S. and Jans, R. 2008. A review of applications of genetic algorithms in lot sizing. Journal of Intelligent Manufacturing. Springer Netherlands. in press, doi:10.1007/s10845-008-0205-2.

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

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

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

[5] Ganeshan, R. 1999. Managing supply chain inventories: a multiple retailer, one warehouse, multiple supplier model. International Journal of Production Economics. 59:341–54.

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

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

[8] Michalewicz, Z. 1996. Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York.

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