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

Woarawichai C.

Department of Industrial Engineering, Faculty of Engineering, KMUTNB, Bangkok, Thailand

Kullpattaranirun T.

Industrial Management Program Faculty of Business Administration, Mahanakorn University of Technology Nong Chok, Bangkok, Thailand

Rungreunganun V.

Department of Industrial Engineering, Faculty of Engineering, KMUTNB, 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 (GAs) are applied to the multi-product and multi-period inventory lot-sizing problems 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. 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 GAs. The detailed computation results are presented.

***Keywords**: Genetic Algorithms, Inventory lot-sizing, Supplier selection, Storage space*

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.

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 provides a literature review on the current inventory lot-sizing. Section 3 we describe our model. Section 4 the genetic algorithm approach is applied to problem. In Section 5 presents a numerical example of the model. Finally, computation results and conclusions are presented in Section 6 and 7.

2 Literature review

Inventory lot-sizing has been one of the most studied problems in production and inventory management literature. Bahl et al [4] proposed four categories for classifying works in this area: (1) single-level unconstrained resources, (2) single-level constrained resources, (3) multiple-level constrained resources, and (4) multiple-level unconstrained resources. Levels refer to the different levels in a bill of material structure where dependency of requirements exists, and constrained resources refer to production capacity limitations.

The scenario discussed in this paper belongs to the second category. The multi-period inventory lot-sizing which involves with multiple products and multiple suppliers under storage space. The study lot-sizing began with Wagner and Whitin [2], provided a dynamic programming algorithm for a single product case. This problem is known as the uncapacitated single item single level lot-sizing problem.

With the advent of supply chain management, much attention is now devoted to supplier selection. Rosenthal et al [5] studied a purchasing problem where one needs to select among suppliers who offer discounts selling a ‘‘bundle’’ of multiply products. Then a mixed integer programming formulation was presented. Chaudhry et al [6] considered vendor selection under quality, delivery and capacity constraints and price-break regimes. Ganeshan [7] presented a model to determine lot sizes that involve multiple suppliers including multiple retailers, and consequent demand on a warehouse. Kasilingam and Lee [8] incorporated the fixed cost of establishing a vendor in a single-period model that includes demand uncertainties and quality considerations in the selection of vendors. Also vein, Jayaraman et al [9] proposed a supplier selection model that considers quality (in terms of proportion of defectives supplied by a supplier), production capacity (constraining the order placed on a supplier), leadtime, and storage capacity limits. This is also a single period model that attaches a fixed cost to deal with a supplier.

Included in the stream of researches integrating supplier selection and procurement lot-sizing are works by Oliver [10], Rule [11], Chappell [12], Williams and Redwood [13], Anthony and Buffa [14], Buffa and Jackson [15], Bender et al [16], Pan [17], Tempelmeier [18], and Basnet and Leung [3]. They consider a multi-period planning horizon and define variables to determine the quantity purchased in each elementary period. Buffa and Jackson [15] presented a schedule purchase for a single product over a defined planning horizon via a goal programming model considering price, quality and delivery criteria. Bender et al [16] studied a purchasing problem faced by IBM involving multiple products, multiple time periods, and quantity discounts. The authors described, but not developed, a mixed integer optimization model, to minimize the sum of purchasing, transportation and inventory costs over the planning horizon, without exceeding vendor production capacities and various policy constraints.

Basnet and Leung [3] presented a multi-period inventory lot-sizing scenario where there are multiple products and multiple suppliers. They considered a situation where the demand of multiple discrete products is known over a planning horizon. The model determines the type, amount, supplier and purchasing time of products. Their model is one of the most useful ones for supply selection in a single stage category. They proposed an uncapacitated mixed integer programming that minimizes the aggregate purchasing, ordering and holding costs subject to demand satisfaction. The authors proposed an enumerative search algorithm and a heuristic algorithm to solve the problem.

Complexity theory and computational experiments indicate that most lot sizing problems are hard to solve [19]. To deal with the complexity and find optimal (or near-optimal) results in reasonable computational time, in recent years, a growing number of researchers have employed heuristic approaches to solve lot sizing problems [20] [21] .

Among these heuristic approaches, evolutionary computation has received the greatest attention. The most well known evolutionary computation is genetic algorithms (GAs). Recently, GAs has been applied to lot-sizing problem [22]. Rezaei and Davoodi [23] have applied GAs for multi-period inventory lot sizing scenario while demand and all costs are considered as fuzzy numbers. Moghadam et al [24] presents inventory lot-sizing with supplier selection and the multi-echelon using a hybrid intelligent algorithm based on a fuzzy neural networks and GAs is designed. A multi objective program for a single item model [25] was assumed that all suppliers’ lots simultaneously arrive at the beginning of each replenishment period. To deal with the multi objective optimization, a GAs was applied.

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.
- Product needs a storage space and available total storage space is limited.

Base on the above assumption of model, Figure 1 shows the behavior of the model considering the scenario of multi-period inventory lot-sizing problem with supplier selection under storage space. The characteristics of the model used to determine what products i, with which suppliers j, and in which periods t to order (X_{ijt}).

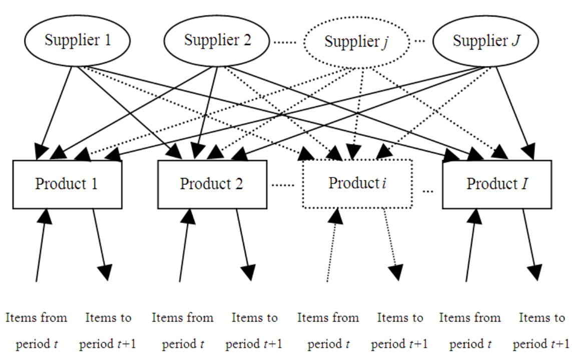

Figure 1: Behavior of the model in period t .

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 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 space

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:

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

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 Eq.(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 Eq. (2) all requirements must be filled in the period in which they occur: shortage or backordering is not allowed. Constraint in Eq. (3) there is not an order without charging an appropriate transaction cost. Constraint in Eq. (4) each products have limited capacity. Constraint in Eq. (5) is binary variable 0 or 1 and Constraint in Eq. (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 Genetic algorithms approach

The genetic algorithms (GAs) approach is developed to find optimal (or near – optimal) solution. Detailed discussion on GAs can be found in books by Holland [26], Michalewicz [27], Gen and Cheng [28] [29], Davis [30] and, Goldberg [31]. In this section, we explain GAs procedure is illustrated in Figure 2. Topics covered include (1) Chromosome structure (2) Initial population (3) Evaluation (4) Selection (5) Crossover (6) Mutation, and (7) Termination rule.

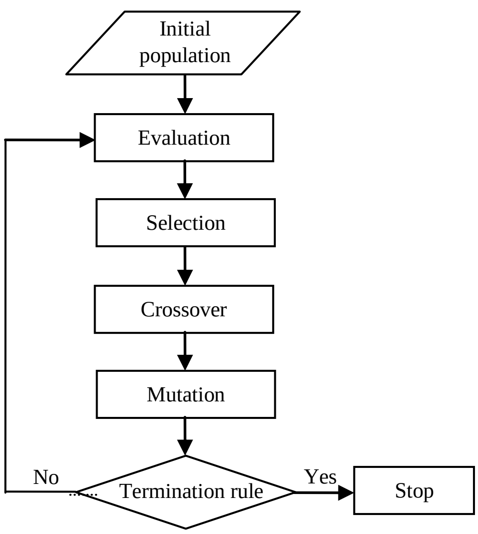

Figure 2: The genetic algorithm procedure

4.1 *Chromosome structure*

In this problem, we take each chromosome as a model solution, where *I*, *J* and *T* are the number of products, suppliers and periods, respectively, and each chromosome is a real values vector (we make it by *X*) by length of (*I* x *J* x *T*) and a binary values vector are 0 or 1 (we make it by *Y*) by length of (*J* x *T*), appropriate by each X_{ijt} and Y_{jt} (decision variables). For example, the representation of a chromosome is illustrated in Figure 3.

Chromosome	X_{111}	X_{112}	$\ldots$	X_{ijt}	$\ldots$	X_{IJT-1}	X_{IJT}
	Y_{11}	Y_{12}	$\ldots$	Y_{jt}	$\ldots$	Y_{JT-1}	Y_{JT}

Figure 3: Chromosome structure

4.2 *Initial population*

The population initialization technique used in the GAs approach is a randomly generate solutions for the entire population. Population size depends only on the nature of problems and it must balance between time complexity and search space measure. More population size may increase the probability of finding optimal solution, but may induce a longer computer time. In this paper, we use a population size is set not less than twice the length of the vector of the chromosomes [24].

4.3 *Evaluation or fitness function*

It is evaluated by the chromosome structure which results in positive value in [32]. Fitness value defines the relative strength of a chromosome compared with the others, and the optimality of the solution to the problem. The fitness function of this model is an objective one (to minimize cost).

4.4 *Selection*

The selection of parents to produce successive generations plays an extremely important role in the GAs. The goal is to allow the fittest individuals to be selected more often to reproduce. However, all individuals in the population have a chance of being selected to reproduce the next generation. In this paper, the roulette wheel selection technique is used. [33] [34].

4.5 *Crossover operator*

Crossover operators combine information from two parents in such a way that the two children (solutions for the next population) resemblance to each parent. There are several available methods to do so [27] [35]. This paper adapts two point crossover operators to solve GAs [33].

4.6 *Mutation operator*

Mutation operators alter or mutate one chromosome by changing one or more variables in some way or by some random amount to form one offspring. For mutation, we use a linear mutation by probability (1/I x J x T) for mutating X vector and bit-wise mutation by probability (1/J x T) for Y vector [23] [34].

4.7 *Termination rule*

The GAs moves from generation to generation selecting and reproducing parents until a termination criterion is met. The most frequently used stopping criterion is a specified maximum number of generations. In this paper, there are two stop criteria. First, the process is stopped when the number of interations has reached the maximum generations. Second, the process is stopped when the maximum time exceeds a given value (set at 120 minutes) [3].

5 A 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 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}) .

	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 space (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_{131} = 12	X_{132} = 15	X_{113} = 37	-	X_{135} = 13
B	X_{231} = 20	X_{231} = 21	X_{213} = 22	X_{234} = 23	X_{235} = 24
C	X_{331} = 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.

6 Computation results

In this section the comparison of the two methods solved problem size is using a commercially available optimization package like LINGO12 and GAs code is developed in MATLAB7. 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 LINGO12 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 Eq.(7) and (8)

Percentage error of LINGO12

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

Percentage error of GAs

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

The solution time of LINGO12 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 LINGO12 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).

Table 5: Comparative results of the two methods

Problem size	Optimization approach with LINGO12			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.

Next, we study differences in the problem sizes between solutions from the optimization with LINGO12 and the GAs. The results are show in Figure 4, a plot of the problem size versus solution time. LINGO12 uses longer computation time more than GAs with seven problem sizes, but uses equal time with two problem sizes.

As show in Figure 5 a plot of the problem size versus % error when the problem size is very large, LINGO12 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). Figure 6 and Figure 7 show compares result between LINGO12 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 LINGO12 fails to find the optimum. However, the GAs provides superior solutions to those from LINGO12 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 space.

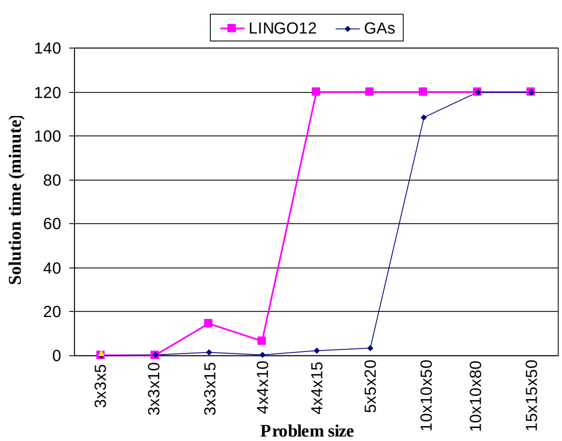

Figure 4: Plot of the problem size vs. solution time (minute)

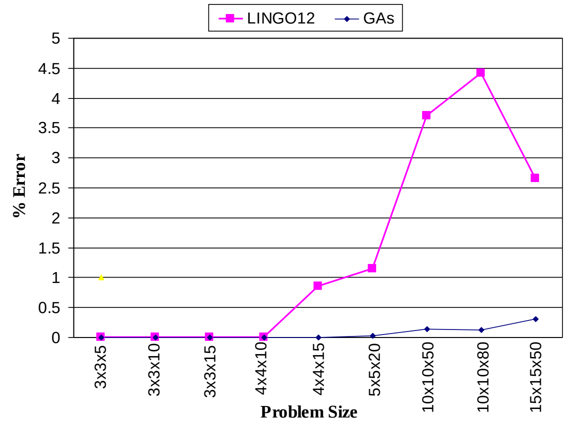

Figure 5: Plot of the problem size vs. % error

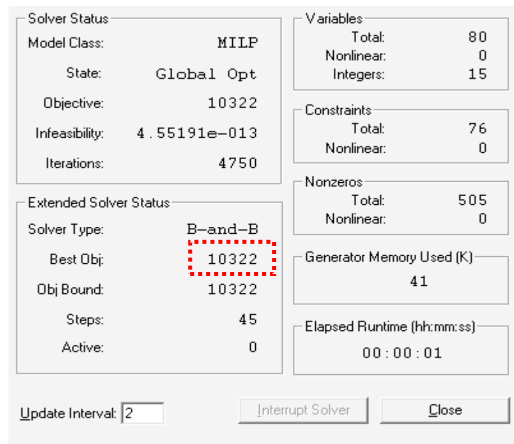

Figure 6: The best objective of LINGO 12

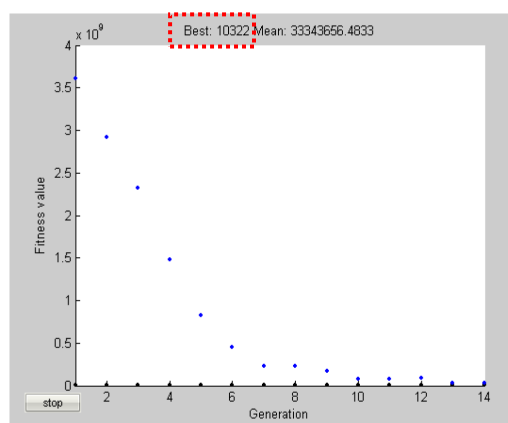

Figure 7: The fitness value of GAs

7 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 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 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 LINGO12 and the GAs. As compared to the solution of optimization package like LINGO12, the GAs solutions are superior.

Acknowledgments

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.

8 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.

[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 and Operations Research*. 32, 1–14.

[4] Bahl H.C., Ritzman L.P., and Gupta J.N.D., 1987. Determining lot sizes and resource requirements: a review. *Operations Research*. 35, 329–345.

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

[6] Chaudhry S.S., Frost F.G., and Zydiak J.L., 1993. Vendor selection with price breaks. *European Journal of Operational Research*. 70, 52–66.

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

[8] Kasilingam R.G., and Lee C.P., 1996. Selection of vendors a mixed integer programming approach. *Computers and Industrial Engineering*. 31, 347–350.

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

[10] Oliver B., 1966. British non-ferrous metals research association. *Research Report* No. A-1583.

[11] Rule P., 1970. Purchasing and melting. An economic mix. *International metals conference*, Amsterdam.

[12] Chappell A.E., 1974. Linear programming cuts costs in production of animal feeds. *Operational Research Quarterly*. 25, 19–26.

[13] Williams H.P., and Redwood A.C., 1974. A structured linear programming model in the food industry. *Operational Research Quarterly*. 25, 517–27.

[14] Anthony T.F., and Buffa F.P., 1977. Strategic purchase scheduling. *Journal of Purchasing and Materials Management*. 13, 27–31.

[15] Buffa F.P., and Jackson W.M., 1983. A goal programming model for purchase planning. *Journal of Purchasing and Materials Management*. 19, 27–34.

[16] Bender P.S., Brown R.W., 1985. Isaac MH, and Shapiro J.F., Improving purchasing productivity at IBM with a normative decision support. system. *Interfaces*. 15, 106–115.

[17] Pan A.C., 1989. Allocation of quantities among suppliers. *Journal of Purchasing and Materials Management*. 25, 36–39.

[18] Tempelmeier H.A., 2002. Simple heuristic for dynamic order sizing and supplier selection with time-varying data. *Production and Operations Management*. 11, 499–515.

[19] Jans R., and Degraeve Z., 2007. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*. 177 (3), 1855–1875.

[20] Pitakaso R., Almeder C., Doerner K. F., and Hartl R. F., 2007. A MAX-MIN ant system for unconstrained multi level lot-sizing problems. *Computers and Operations Research*. 34(9), 2533–2552.

[21] Tempelmeier H., and Helber. S., 1994. Heuristic for dynamic multiitem multi-level capacitated lot sizing for general product structures.

European Journal of Operational Research. 75(2), 296–311.

[22] Aytug H., Khouja M., and Vergara F. E., 2003. Use of genetic algorithms to solve production and operations management problems: A review. *International Journal of Production Research*. 41(17), 3955– 4009.

[23] Rezaei J., and Davoodi M., 2006. Genetic algorithm for inventory lot-sizing with supplier selection under fuzzy demand and costs. *Advances in Applied Artificial Intelligence*.Springer-Verlag Berlin Heidelberg. 4031, 1100 – 1110.

[24] Moghadam S.R.M., Afsar A., and Sohrabi B., 2008. Inventory lot-sizing with supplier selection using hybrid intelligent algorithm. *Applied Soft Computing*. 8, 1523–1529.

[25] Liao Z., and Kuhn A., 2004. Operational integration of supplier selection and procurement lot sizing in supply chain. Proceedings of global project and manufacturing management symposium, *Siegen Germany*.

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

[27] Michalewicz Z., 1994. Genetic Algorithms + Data Structures = Evolution Programs. *AI Series*. Springer-Verlag, New York.

[28] Gen M., and Cheng R., 1997. Genetic Algorithms and Engineering Design. Wiley, New York.

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

[30] Davis L., 1991. The handbook of genetic algorithms, Van Nostrand Reinhold, New York.

[31] Goldberg D. E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.

[32] Chan S.H., Chung W., and Wadhwa S., 2005. A hybrid genetic algorithm for production and distribution. *Omega*. 33, 345–355.

[33] Sarker R., and Newton C., 2002. A genetic algorithm for solving economic lot size scheduling problem. *Computers and Industrial Engineering*. 42, 189-198.

[34] Rezaei J., and Davoodi M., 2008. A deterministic, multi-item inventory model with supplier selection and imperfect quality. *Applied Mathematical Modelling*. 32, 2106–2116.

[35] Deb K., 2001. Multi-Objective Optimization using Evolutionary Algorithms Wiley, Chichester, UK.