Genetic Algorithm Approach for the Inventory Routing Problem with Backlogging

S. Sofianopoulou,

Department of Industrial Management and Technology, University of Piraeus,
80 Karaoli & Dimitriou str., 18534 Piraeus, Greece
sofianop@unipi.gr

Abstract. We consider a multiperiod inventory-routing problem where a vendor serves multiple geographically dispersed customers who receive units of a single product from a depot with adequate supply. The class of problems arising from the combination of distribution and inventory management decisions is perhaps the most striking example of this concept and is known as the inventory routing problem (IRP). In this category of problems, the inventory routing problem with backlogs (IRPwB) deals with determining inventory level, backlogging and vehicle routing decisions from a single depot to a set of n customers over a specific number of time periods, using a fleet of homogenous vehicles. The aim is to minimize the average daily cost for the planning period, while ensuring that inventory level capacity constraints are not violated. We first develop an Integer Programming model to provide an accurate description of the problem and in a second phase a Genetic Algorithm (GA) with suitably designed genetic operators, is employed in order to obtain near optimal solutions.

Keywords: Supply Chain Management, Inventory Routing Problem, Genetic Algorithms.

1. Introduction

Nowadays, the business world comes to realize the immense potential that the combination-integration of various elements of the Supply Chain Management has to offer such as significant savings in inventory handling cost, increased efficiency, higher delivery performance to requested or committed due dates. More and more researchers are focusing on implementing techniques that render this integration not only viable but also profitable. The present work considers the most typical example of this concept, that is, the integration of distribution management and inventory control. Specifically, instead of the customers having to monitor their inventory and make sure that it is replenished on a regular basis, the supplier is responsible for serving customer needs by determining simultaneously the timing and sizes of the deliveries as well as efficient vehicle schedules so as to minimize total transportation and inventory carrying costs. The category of problems arising from the combination of distribution and inventory management decisions is known as “Inventory Routing Problems” (IRP) and generally is considered to be an extension of the Vehicle Routing Problems (VRP). In IRP, the inventory routing problem with backlogs (IRPwB) deals with determining inventory level, backlogging and vehicle routing decisions from a single depot to a set of n customers over a specific number of time periods, using a fleet of homogenous vehicles. A variety of industries have provided fertile ground for the development of IRP and within such industries there are situations where backorders are permitted. Researchers have generated numerous methods of solving IRP and in a less extend IRPwB. Most of the methods proposed employ either a “theoretical” approach or a more “practical” one. The theoretical one, consists of attempting to specify the lower bounds to the problem, whereas the practical one employs heuristics in order to obtain near optimal solutions.

Currently most researchers concentrate on the IRP where inventory and routing decisions are combined in a cost optimization problem where backorders are not permitted and multiple distribution levels may occur (Jianxiang Li , Feng Chu and Haoxun Chen, 2011). Depending now on the approaches developed for the solution of IRP, we will adopt the classification introduced by Bramel and Simchi-Levi (1997). According to this, most of the research on IRP is in one of three directions: single-day models using deterministic demand (Chien, Balakrishnan, and Wong, 1989; Bertazzi, Paletta, and Speranza, 2002), multi-day models using deterministic demand (Bard, Huang, Jaillet and Dror, 1998; Campbell and Savelsbergh, 2004), and infinite time horizon usually for long-term planning purposes (Anily and Bramel, 2004). Finally, stochastic demand has also been considered by several researchers (Kleywegt, Nori, and Savelsbergh, 2002). Concerning now the literature that refers to IRP models with backlogging recent works include a local search approach applied to the problem (Zachariadis, Tarantilis, and Kiranoudis, 2009), where two local search operators for jointly dealing with the inventory and routing aspects of the problem are employed together with a Tabu Search approach for the transportation cost reduction. Also in (Abdelmaguida, Dessouky and Ordóñez, 2009) the authors study an inventory-routing problem in which multiperiod inventory holding, backlogging, and vehicle routing decisions are to be taken using constructive and improvement heuristics.

In this paper a multi-retailer single product routing problem where backordres are permitted is examined and thoroughly reviewed. In the following section the problem is formally stated, while in the next section Genetic Algorithms are introduced and related operators are discussed. Then, the application of the algorithmic procedure to IRPwB is implemented, while concluding remarks and suggestions for further research are presented in the last section.

2. Problem statement

IRPwB is concerned with the repeated distribution of a single product, from a single facility, i.e. a central depot, to a set of *N={1,2,…,n}* dispersed retailers over a given planning horizon of length *T*. For each retailer $i \in N$, we have a deterministic demand d_i^t , a level of inventory I_i^t (with a maximum of C_i) and a backlogging level B_i^t per time period $t \in \mathrm{T}$. The amount of delivery q_i^t , to customer *i* in period *t*, is to be decided and based on the delivery amounts in period t. Each retailer *i*, incurs an inventory holding cost of h_i and a shortage cost (penalty) of b_i per period per unit. We assume that the depot has a sufficient supply of items to cover all customers' demands throughout the planning horizon. Deliveries can be carried out at any time period $t \in \mathrm{T}$, using vehicles with virtually unlimited capacity. Any combination of retailers can be visited in a single delivery route and the transportation cost c_{ij} , i.e. from retailer *i* to retailer *j*, is given. The following is an Integer programming formulation of IRPwB:

$$\min \sum\sum h_i I_i^{\ t} + \sum\sum b_i B_i^{\ t} + \sum\sum\sum c_{ij}\ w_{ij}^{\ t} \qquad (1)$$

subject to the following constraints:

$$I_i^{\ t+1} = I_i^t - B_i^t + q_i^t - d_i^t \qquad (2)$$

$$I_i^t \le C_i \qquad (3)$$

$$\sum_j w_{ij}^t \le 1 \ , \quad \forall\ i,t \qquad (4)$$

$$z_{ij}^t \le M w_{ij}^t \qquad \forall\ i,j,t \qquad (5)$$

$$\sum w_{ij}^t - \sum w_{li}^t = 0 \qquad \forall\ i,t \qquad (6)$$

$$\sum_l z_{li}^t - \sum_k z_{ik}^t = q_i^t \qquad \forall\ i,t \qquad (7)$$

$$I_i^t, B_i^t, q_i^t, z_{ij}^t \ge 0,\ w_{ij}^t = 0,1 \text{ , M: large number} \qquad (8)$$

The total cost comprises of the inventory holding cost, the transportation cost, and the shortage cost as depicted in the objective function (1). Constraint (2) is the inventory balance equation for the customers. Constraint (3) limits the total amount of inventory to C_i . Constraint (4) ensures that a vehicle will not visit the location of a specific retailer more than once. w_{ij}^t is a binary variable that is equal to 1, if the vehicle visits retailers at location *i* and *j* successively. Constraint (5) makes sure that if there isn't a vehicle travelling between two locations (i.e. retailers) then the amount delivered between them will be zero. z_{ik}^{t} is a continuous variable representing material flow. Constraint (6) is used in order to ensure route continuity and constraint (7) reflects the equity between materials flow and the quantity of products to be

3. Genetic algorithms

Since the IRPwB is NP-hard, accurate solution methods are difficult to adopt, while heuristic approaches are more promising. Genetic algorithms (GAs) are efficient heuristic procedures often surpassing the effectiveness of more "traditional" algorithms. There are several reasons for that. Firstly, GAs work with a coding of the parameters whereas "normal" optimisation and search procedures use the parameters themselves. Secondly, GAs perform their search from a population of points, not a single point. Thirdly, they use payoff (objective function) information and do not rely on derivatives or other auxiliary information. Finally, GAs make use of probabilistic transition rules, not deterministic ones.

In genetic algorithms, the term "chromosome" typically refers to a candidate solution to a problem, often encoded as a bit string. The "genes" are either single bits or short blocks of adjacent bits that encode a particular element of the candidate solution (e.g., in the context of multi-parameter function optimization the bits encoding a particular parameter might be considered to be a gene). An allele of a bit string is either 0 or 1; for larger alphabets more alleles are possible at each locus. In correspondence to the term genotype used in natural systems, in artificial systems the term structure is employed in order to refer to the total package of strings. Moreover, the term "phenotype" is analogous to a "parameter set", or a "solution alternative" that is formed by the decoding of a structure and the locus of a gene is analogous to the position of a bit in a string.

3.1 **Genetic Operators**

As mentioned earlier, GAs resemble natural systems and basically follow the same principles. In GAs, during each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions (strings) are selected through a *fitness-based* process. In GAs, as a number of genetic operators is employed (selection, crossover, mutation) to ensure that the average fitness of the population of strings will continue to increase in successive generations, good partial solutions combine to form even better composite solutions.

Genetic operators lie at the core of GAs and are of crucial importance to the ability of the algorithm to produce high quality solutions. *Selection* ensures the survival of the fittest and resembles the "natural selection" process. It determines during each successive generation, which strings will reproduce and pass on their genes to the next generation, according to fitness criteria. Popular and well-studied selection methods include roulette wheel selection and tournament selection. In roulette-wheel selection, the fitness function assigns a fitness value to possible solutions (strings).

After selecting the strings, the next step is to generate a new population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation. For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each child, and the process continues until a new population of solutions of appropriate size is generated. These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

Analytically, during the process of *crossover*, two individuals are selected from the population and a crossover site (a position) along the bit strings is randomly chosen. Then, substrings from corresponding positions within the individuals are exchanged. One or both of the new individuals are inserted into the population at the next generation. For example if S_1=000000 and S_2=111111 are the parent strings and the crossover point is 2 then S_1'=110000 and S_2'=001111. *Mutation* helps the GA procedure to maintain genetic diversity from one generation of a population of chromosomes to the next. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified

3.2 Structure of the Algorithm

The steps for implementing a GA are the following:

1. Randomly create an initial population (generation 0)
2. Iteratively perform the following sub-steps on the population until the termination criterion (i.e. a satisfactory fitness level or a maximum number of generations), is satisfied:
   a. For each string in the population calculate its fitness.
   b. Select one or two individual(s) from the population with a probability based on fitness to participate in the genetic operations in (c).
   c. Create new individual for the population by applying the following genetic operations with specified probabilities:
      - ✓ *Reproduction:* Copy the selected individual to the new population.
      - ✓ *Crossover:* Create new offspring(s) for the new population by recombining randomly chosen parts from two individuals.
      - ✓ *Mutation:* Create one new offspring for the new population by randomly mutating a randomly chosen part of one selected individual.

3. After the termination criterion is satisfied, the single best individual (string) in the population produced during the run (the best-so-far individual) is harvested and designated as the result of the run. If the run is successful, the result may be a solution to the problem.

4. A Genetic Algorithm Adapted to the Inventory Routing Problem

In order to use GAs for solving a problem, the encoding of candidate solutions is necessary . Refering to IRPwB, a representation of a solution is a two-dimensional matrix (Abdelmaguid and Dessouky, 2006), with each cell containing the quantity of product delivered to customer *i* during time period *t,* each row corresponding to a specific customer I and each column corresponding to a time period t. It is worth noting that delivery amounts are set to be integers.

An initial population is required. The genesis of this population is conducted using a two-step heuristic algorithm. In phase one, the heuristic procedure examines which customers are in need of a delivery. Generally, retailers are served if their inventory level in period *t* cannot cope with the demand of that period. Firstly the heuristic decides which customers are in immediate need of a delivery, simply by using a function that calculates the inventory level of customer *i, in* period *t*. In phase two, a network of the customers who are to be serviced is generated and the transportation cost, for supplying each retailer in the network is minimised with the use of a modified Dijkstra algorithm. The amount of product to deliver to each retailer in the network is $q_i^t = d_i^t - I_i^{t-1} + B_i^{t-1}$. This procedure is repeated for each of the time periods. The heuristic also makes sure that none of the constrains of the IRP is violated by the proposed solution.

Genetic Operators

The *selection* of individuals to produce successive generations plays an extremely important role in GAs. The basic methods we have chosen to apply are selection based on roulette wheel and elitism. Our selection operation at first locates the elite chromosome that is the solution with the least total cost. This solution is automatically passed on to the new population in order to safeguard that it will not be "eliminated" during the selection process. Next, the fitness of each individual is computed. As we are dealing with a minimization problem in that our goal is to minimize the sum of the inventory, backorder and routing costs for a number of periods, it is vital to map the underlying natural objective function to a fitness function. This transformation is performed using the method proposed by Goldberg (1989), according to which:

$$f(x) = C_{\max} - g(x) \quad \text{when } g(x) < C_{max}$$

$$f(x) = 0 \quad \text{otherwise}$$

where $g(x)$ is the cost minimization function, $f(x)$ is the fitness function and C_{max} the largest g value in the current population. The new population is then created using the fitness values computed earlier and a roulette wheel mechanism.

Given the nature of the problem and the encoding selected, *crossover* is the most intriguing operator. Crossover in our case can be performed either horizontally or vertically. While horizontal crossover, that is random exchange of delivery schedules (rows) of certain customers between the "parent" solutions, poses no problem, vertical crossover may cause a violation of the constraint concerning the capacity of each retailer. Vertical crossover i.e. exchange of rows between "parent" solutions corresponds to the possibility of having excessive amounts of product which in turn results to poor fitness -as inventory holding cost are too great. Consequently, the latter method is not given any possibility to occur.

Following the crossover operation, *mutation* is applied to each offspring generated with a probability equal to the mutation rate. Specifically the amount of product to be delivered is mutated (decreased or increased) randomly by a small amount; and consequently new solutions are created. As with the previous operators, violations of constraints by the mutated solutions are not allowed.

5. Illustrative example

The proposed method is applied to a simple example which concerns a single depot and six customers. The distribution plan spans over 3 periods, while demand, holding costs and distances between customers and depot are shown in Tables 1-3 respectively. Shortage costs are assumed to be twice the holding costs. Starting inventory at T=0 is also presented in Table 3. Maximum inventory capacity for each customer is 29, 40, 19, 42, 94 and 32 respectively.

Results computed for the distribution plan are presented in Table 4, whilst the corresponding routes travelled in each time period are as following: [0 5 6 2 3 0] [0 5 3 1 2 4 0] [0 5 6 4 3 1 0] correspondingly with a total cost of 158 .

6. Concluding Remarks

The purpose of this work is to present a GA approach for solving the Inventory Routing Problem with backlogs. The paper consists of three parts. Firstly, IRPwB is introduced, followed by a formal problem statement. A description of the problem, using a model of Mixed-Integer Programming is given at the end of this part. Secondly, the basic theoretical foundations for Genetic Algorithms are presented and a solution for the Inventory Routing Problem is demonstrated through an illustrative example which depicts the structure of the data required and the solution produced. In order to test the algorithm, further experiments should be carried out using a variety of test problems which hasn't been the case in this study, where only a small test problem was used for illustrative purposes. Further experimentation will make it possible for the researchers to determine the best combination of algorithmic features.

TABLES

Table 1: Distances between individual customers and Depot

	1	2	3	4	5	6	D
1	0	5	1	30	10	4	5
2	5	0	4	15	20	18	32
3	1	4	0	9	10	25	6
4	30	15	9	0	15	3	10
5	10	20	10	15	0	9	1
6	4	18	25	3	9	0	2
D	5	32	6	10	1	2	0

Table 2: Holding costs for each customer per period

T=	1	2	3
1	3	2	1
2	2	3	4
3	4	1	2
4	3	2	5
5	4	3	2
6	1	3	4

Table 3: Demand per period (d_i^t) and beginning inventory (I_i^0) for each customer i

	(d_i^1)	(d_i^2)	(d_i^3)	(I_i^0)
1	10	11	12	12
2	15	15	8	14
3	12	10	11	2
4	13	10	9	14
5	14	13	7	9
6	8	14	15	1

Table 4: Distribution plan for each customer per period

T=	1	2	3
1	0	9	12
2	1	23	0
3	10	10	11
4	0	9	9
5	5	13	7

References

1. Abdelmaguid, T.F. and Dessouky M.M. 2006, "A genetic algorithm to the integrated inventory-distribution problem", *International Journal of Production Research* 44 : 4445-4464.
2. Abdelmaguid , T.F. , Dessouky, M.M., and Ordóñez F. 2009, "Heuristic approaches for the inventory-routing problem with backlogging", *Computers & Industrial Engineering* 56 :1519-1534.
3. Anily S. and Bramel J. 2004. "A Probabilistic Analysis of a Fixed Partition Policy for the Inventory-Routing Problem", *Naval Research Logistics* 51: 925–948.
4. Bard, J., L. Huang, P. Jaillet and M. Dror.A. 1998. "Decomposition Approach to the Inventory Routing Problem with Satellite Facilities", *Transportation Science* 32: 189-203.
5. Bertazzi, L, Paletta, G and Speranza, M. 2002. "Deterministic order-up-to level policies in an inventory routing problem", *Transportation Science* 36 : 119–132.
6. Bramel J and Simchi-Levi D. 1997. *The Logic of Logistics*. Springer: New York.
7. Campbell A., Savelsbergh M.W.P. 2004. "A decomposition approach for the inventory-routing problem", *Transportation Science* 38 : 488-502.
8. Chien, W., Balakrishnan, A. and Wong, R. 1989 "An integrated inventory allocation and vehicle routing problem", *Transportation Science* 23 : 67-76
9. Goldberg D. 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional.

10. Jianxiang Li , Feng Chu, and Haoxun Chen 2011, “ A solution approach to the inventory routing problem in a three-level distribution system”, *European Journal of Operational Research* 210: 736-744.
11. Kleywegt J., Nori V. S., and Savelsbergh M. W. P. 2002, “The Stochastic Inventory Routing Problem with Direct Deliveries”, *Transportation Science* 36 : 94–118.
12. Zachariadis, E., Tarantilis C., and Kiranoudis C.T. 2009, “An integrated local search method for inventory and routing decisions”, *Expert Systems with Applications* 36 : 10239-10248.