Lecture Notes in Management Science (2014) Vol. **6**: 49–57
6th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

A heuristic approach for an inventory routing problem with backorder decisions

Stella Sofianopoulou

Department of Industrial Management & Technology, University of Piraeus, Greece
sofianop@unipi.gr

Abstract. A multi-period inventory-routing problem is considered where a vendor serves multiple geographically dispersed customers who receive units of a single product from a depot, with adequate supply, using a capacitated vehicle. The class of problems arising from the combination of routing and inventory management decisions is known as the inventory routing problem (IRP). In this category of problems, the inventory routing problem with backorders (IRPB) deals with determining inventory levels when backorders are allowed. The aim is to minimize the total cost for the planning period, comprising of holding cost, transportation and backorder penalty cost while ensuring that inventory level capacity constraints are not violated. An Integer Programming model is first developed to provide an accurate description of the problem and then a Genetic Algorithm (GA) with suitably designed genetic operators is employed in order to obtain near optimal solutions. Computational results are presented to demonstrate the effectiveness of the proposed procedure.

Keywords: inventory routing problem; backlogging; genetic algorithms

Introduction

Vendor Managed Inventory together with the underlying Inventory Routing Problem results in significant cost savings of inventory handling, increased efficiency, better performance, and committed delivery dates. More precisely, instead of customers need to keep their stocks and make sure it is replenished on a regular basis, the supplier is responsible for responding to customer needs by simultaneously determining the timing and size of deliveries and schedules effectively vehicles to minimize the total transportation and inventory holding costs. The class of problems arising from the combination of inventory control problems and NP-hard vehicle routing problems, which determines delivery volumes to the customers that the supplier serves in each period, and vehicle routes to deliver the volumes is known as "inventory routing problems" (IRP). The inventory routing problem with backlogging (IRPB), i.e. the inventory routing problem when backorders are allowed, is an IRP variant with the aim to determine inventory levels, backlogging and vehicle routing decisions for a set of n customers on a number of time periods. A variety of industries have provided fertile ground for the development for the IRP and its variants. Researchers have generated many methods of solving IRP and to a lesser extend IRPB. Most of the proposed methods use either a "theoretical" or a more "practical" approach. The theory concentrates mainly to compute lower bounds of the problem, while the practical approaches employ heuristic procedures to obtain near optimal solutions.

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 transportation cost reduction. Also in (Abdelmaguida, Dessouky and Ordóñez, 2009) the authors study an inventory-routing problem in which multi-period inventory holding, backlogging, and vehicle routing decisions are to be taken using constructive and improvement heuristics.

The inventory routing problem with backlogging (IRPB) is introduced and solved in this work. In this paper a multi-retailer single product routing problem where backorders are permitted is examined and reviewed. In the following section the problem is formally stated, while in the next section a Genetic Algorithm is presented and related operators are discussed. Next the application of the algorithmic procedure to IRPB is implemented, while concluding remarks and suggestions for further research are presented in the last section.

Problem Statement

IRPB is formally presented in this section. The problem is concerned with the repeated distribution of a single product, from a single facility, i.e. a central depot, to a set of $N=\{1,2,\dots,n\}$ dispersed retailers over a given planning horizon of length T using a single vehicle of given capacity. For each retailer $i \in N$, we have a deterministic demand d_i^t, a level of inventory I_i^t (with a maximum capacity of C_i) and a backlogging level B_i^t with a maximum level Bl_i 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 to other retailers in period t.

Each retailer *i*, incurs an inventory holding cost of h_i and a shortage cost (penalty) of b_i per time 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 a vehicle of limited 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 IRPB presented in the paper mainly for accurately describing the problem tackled:

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

Subject to:

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

$$I_i^t \le C_i \ \forall\ i,t \qquad (3)$$

$$B_i^t \le Bl_i \qquad \forall\ i,t \qquad (4)$$

$$\sum_j w_{ij}^t \le 1 \qquad \forall\ i,t \qquad (5)$$

$$z_{ij}^t \le M w_{ij}^t \qquad \forall\ i,j,t \qquad (6)$$

$$\sum_i q_i^t \le VQ \qquad \forall\ t \qquad (7)$$

$$\sum_j w_{ij}^t - \sum_l w_{li}^t = 0 \qquad \forall\ i,t \qquad (8)$$

$$\sum_l z_{li}^t - \sum_k z_{ik}^t = q_i^t \qquad \forall\ i,t \qquad (9)$$

$$I_i^t, B_i^t, q_i^t, z_{ij}^t \ge 0,\ w_{ij}^t = 0{,}1 \ \text{M: large number} \qquad (10)$$

The total cost comprises of the inventory holding cost, the transportation cost, and the shortage cost as depicted in the objective function (1). Constraints (2) are the inventory balance equations for customers. Constraints (3) and (4) limit the total amount of inventory and backlogging to C_i and B_i respectively. Constraints (5) ensure 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. Constraints (6) make 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_{ij}^t is a continuous variable representing material flow. Constraints (7) ensure that the total quantity delivered to all customers during period t does not exceed vehicle's capacity VQ. Constraints (8) are used in order to ensure route continuity and constraints (9) reflect the equity between materials flow and the quantity of products to be delivered. Constraints (10) are domain constraints.

Genetic Algorithms

The IRP is classified as NP-hard problem since it subsumes the Vehicle Routing Problem (VRP). This fact led to the development of many heuristic approaches, although a small number of exact methods have been recently introduced (Archetti *et al.* 2007; Solyalı and Süral 2011; Adulyasak, Cordeau, and Jans 2012; Coelho and Laporte 2013). Nevertheless, there are limitations in their use as benchmarks as there exist several IRP variants. Since the IRPB is NP-hard, accurate solution methods are difficult to adopt, and 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. Second, GAs perform their search from a population of points, not a single point. Third, 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.

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.

Structure of the algorithm

- The steps for implementing a GA are the following:
- Randomly create an initial population (generation 0)
- 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:
- For each string in the population calculate its fitness.
- Select one or two individual(s) from the population with a probability based on fitness to participate in the genetic operations in (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.
- 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.

A Genetic Algorithm Adapted to IRPB

In order to use GAs for solving a problem, the encoding of candidate solutions is necessary. Referring to IRPB, 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.

Generally, retailers are served if their inventory level in period *t* cannot cope with the demand of that period. When retailers are served at a certain time period and quantity delivered is not adequate to cover demand, backlogging is assumed. In practice, backlogging decisions are generally justified in two cases. The first is when there is a transportation cost saving that is higher than the incurred shortage cost by a customer. The second case is when there is a holding cost saving that is higher than the incurred shortage cost by a customer. Then care is taken that backorders are satisfied by the quantity shipped in next period.

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. 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 under examination. The heuristic also makes sure that none of the constraints of the IRPB model 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:

$$f(x) = C_{\max} - g(x) \quad \text{when } g(x)<C_{max} \tag{11}$$

$$f(x) = 0 \quad \text{otherwise} \tag{12}$$

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.

Algorithm Testing

The heuristic method has been coded and compiled in MATLAB (version R2008b) while computational testing was performed on a 2.17 GHz Intel Core 2 Duo processor with 4 GB of RAM. For the evaluation of the proposed approach we have used the instances suggested by Abdelmaguid *et al.* (2009) for the Inventory Routing Problem. In this latter work instances for IRP were tested both for the IRP with backorders and without. Those IRP with backorders instances were adapted to our IRPB.

Abdelmaguid *et al.* (2009) consider three scenarios in their problem instances, where the second scenario is designed using such parameters so to provide conditions in which backorder decisions are economical to be taken. We have therefore chosen to run instances with 5 customers, 5 time periods and one vehicle with limited capacity of 150, since only one vehicle is considered in our work. It should be also noted that no fixed usage cost per vehicle travelling per period was considered in our work.

Following the name convention adopted by Abdelmaguid *et al.* (2009) the instances selected to run are denoted by x-cc-t-v-i where x stands for the scenario (we have selected only second scenario instances), cc stands for the number of customers considered (five customers instances were selected), t for the number of time periods of the planning horizon (5 periods), v for the number of vehicles utilized and finally i stands for the instance number. It should be mentioned at this point that these test cases used provide fertile ground for a preliminary testing of the algorithmic procedure described and further tests should be performed on larger problems.

In Table 1 the results for those instances are reported when backorders are allowed. Table 1 presents the cost components of solutions obtained in each case. The optimum (obtained through a MPI solver) is listed in the second column while TR, H and B are the transportation, holding and backorder costs respectively for each solution obtained by our heuristic procedure while the total cost, calculated as the sum of all cost components, is listed in the last column.

Table 1. Results for the IRP with backorders

Problem	*Optimum*	*TR*	*H*	*B*	*Total Cost*
2-05-5-1-1	599,80	510	7,78	223,07	740,85
2-05-5-1-2	418,00	481	8,72	53,22	542,94
2-05-5-1-3	350,00	347	18,66	53,04	418,7
2-05-5-1-4	425,29	396	6,95	110,97	513,92
2-05-5-1-5	376,01	449	6,62	46,06	501,68

Concluding Remarks

The purpose of this work is to address the Inventory Routing Problem when backorder decisions are allowed (IRPB). We present a simple GA approach for solving the problem. The paper consists of three parts. Firstly, IRPB is introduced, followed by a formal problem statement where a description of the problem, using a model of Mixed-Integer Programming is given. Second, some basic theoretical foundations for Genetic Algorithms are presented. The solution for the IRPB is demonstrated in the third part through some basic testing using instances of the problem presented in the related literature. Further experimentation will make it possible for the researchers to determine the best combination of algorithmic features required to improve the algorithmic procedure or to incorporate additional features to the problem examined.

References

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.

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.

Adulyasak, Y., Cordeau, J.-F., and Jans, R., 2012. Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems. Technical Report G-2012-14, GERAD, Montreal.

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.

Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transportation Science, 41 (3), 382-391.

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.

Bertazzi, L, Paletta, G. and Speranza, M. (2002). Deterministic order-up-to level policies in an inventory routing problem, Transportation Science, 36: 119–132.

Bramel, J. and Simchi-Levi, D. (1997). The Logic of Logistics. Springer: New York.

Campbell A and Savelsbergh MWP (2004). A decomposition approach for the inventory-routing problem, Transportation Science, 38: 488-502.

Chien, W., Balakrishnan, A. and Wong, R. (1989). An integrated inventory allocation and vehicle routing problem, Transportation Science, 23: 67-76

Coelho, C. L., Cordeau, J-F and Laporte G. (2012). The inventory-routing problem with transshipment, Computers & amp; Operations Research, 39 (1): 2537-2548.

Coelho, L.C. and Laporte, G., 2013. The exact solution of several classes of inventory-routing problems. Computers & Operations Research, 40 (2), 558-565.

Goldberg D. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley Professional.

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.

Kleywegt, J., Nori, V. S. and Savelsbergh, M. W. P. (2002). The Stochastic Inventory Routing Problem with Direct Deliveries, Transportation Science, 36: 94–118.

Solyalı O and Süral H (2011). A branch-and-cut algorithm using a strong formulation and an a priori tour based heuristic for an inventory-routing problem. Transportation Science 45, 335-345.

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.