



# An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem

N.H. Moin<sup>a,\*</sup>, S. Salhi<sup>b</sup>, N.A.B. Aziz<sup>a</sup>

<sup>a</sup> Institute of Mathematical Sciences, University of Malaya, 50603 Kuala Lumpur, Malaysia

<sup>b</sup> Centre for Logistics and Heuristic Optimisation, Kent Business School, The University of Kent, UK

## ARTICLE INFO

Available online 9 July 2010

### Keywords:

Inventory routing  
Genetic algorithm  
ILP formulation  
Inbound logistics

## ABSTRACT

The inventory routing problem (IRP) addressed in this study is a many-to-one distribution network consisting of an assembly plant and many distinct suppliers where each supplies a distinct product. We consider a finite horizon, multi-periods, multi-suppliers and multi-products where a fleet of capacitated homogeneous vehicles, housed at a depot, transport products from the suppliers to meet the demand specified by the assembly plant in each period. The demand for each product is deterministic and time varying. A mathematical formulation of the problem is given and CPLEX 9.1 is run for a finite amount of time to obtain lower and upper bounds. A hybrid genetic algorithm, which is based on the allocation first route second strategy and which considers both the inventory and the transportation costs, is proposed. In addition to a new set of crossover and mutation operators, we also introduce two new chromosome representations. Several medium and small sized problems are also constructed and added to the existing data sets to show the effectiveness of the proposed approach.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

The need for the integration and coordination of various components in a Supply Chain Management (SCM) has been recognized as an important factor for most companies to remain competitive. Most of the activities in the SCM are inter-related and changes in one part of the SCM are likely to affect the performance of other processes. Inventory management and transportation are two of the key logistical drivers of the SCM. Other components include production, location, marketing and purchasing (see Moin and Salhi (2007) for an overview). The coordination of these two drivers, often known as the inventory routing problems (IRP), is critical in improving the SCM. The IRP seeks to determine simultaneously an optimal inventory and distribution strategy that minimizes the total cost. The resulting inventory and transportation policies usually assign suppliers to routes and then determine the replenishment intervals and collection sizes for each supplier. The implementation of IRP is critical especially in a Vendor Managed Inventory (VMI) replenishment system where the supplier or manufacturer observes and controls the inventory levels of its customers or retailers. One of the most important benefits of VMI is that it permits a more uniform utilization of transportation resources. This leads to a higher level of efficiency and a much lower distribution cost that often constitutes the largest part of the overall cost.

IRP can be broadly categorized according to the following criteria: planning horizon, single or multi-periods and whether the demand is deterministic or stochastic. Several other variants of IRP can also be found depending on the underlying assumptions in the models. We will first present a brief review emphasizing those studies that consider finite horizon, multi-period and deterministic demand as these are closely related to the proposed research.

Most of the earlier works in IRP focus on a single period model, which is actually the classical vehicle routing problem (VRP). Federgruen and Zipkin (1984) are amongst the first to study the inventory routing problem. The problem was treated as a single day problem with a limited amount of inventory and the customers' demands were assumed to be a random variable. They have extended and modified some well-known VRP heuristics to handle the problem that was formulated as a nonlinear integer program. The problem is decomposed into a nonlinear inventory allocation problem that aims to determine both the inventory and the shortage costs, and a traveling salesman problem (TSP) for each vehicle to find the least transportation cost. Federgruen et al. (1986) expand the idea for the case of perishable products. The solution has been adopted from their earlier work (Federgruen and Zipkin, 1984) with a variation that the inventory allocation subproblems account for two product classes instead.

Chien et al. (1989) are among the first to simulate a multiple period planning model based on a single period approach. This is achieved by passing some information from one period to the next through inter-period inventory flow. In their problem, there

\* Corresponding author.

E-mail address: noor\_hasnah@um.edu.my (N.H. Moin).

is one central depot with all customers assigned to it. The capacity of the depot and the demand of the customers are fixed. An integer program is modeled using a Lagrangian dual ascent method to handle the allocation of the limited inventory available at the plant to the customers, the customer to vehicle assignments and the routing of the vehicles. A similar approach has been adopted by Fisher et al. (1982) to solve a real-life problem of inventory routing at Air Products, an industrial gases producer. The objective is to maximize the profit from product distribution over several days. The demand is given by the upper and lower bounds on the amount to be delivered to each customer for every period in the planning horizon. The effect of the short-term planning period in subsequent decisions has also been studied by Dror and Ball (1987) and Dror et al. (1985). They proposed an integer program where consequences of present decisions on later periods are accounted for using penalty and incentive factors. In this problem, the single period models are used as subproblems.

A multi-period model with two types of customers, namely the vendor-managed inventory (VMI) customers and the customer managed inventory (CMI) customers, is proposed in Ribeiro and Laureço (2002). The VMI customers face stochastic demands while the demands for the CMI customers are deterministic. In addition, the demand of all the CMI customers must be satisfied. On the one hand, the deliveries of products for the VMI customers are determined by the distributor who is also responsible for both the inventory holding cost and the stock-out cost at these customers. On the other hand, the CMI customers determine the amount to be delivered and the day in which the products need to be delivered; thus, the distributor is not liable to any inventory holding cost at these customers. The authors propose an iterated local search heuristic to determine the delivery routes (for both the VMI and the CMI customers) and the amount to be delivered for the VMI customers. The results show that this integrated model yields a better performance than its counterpart, the nonintegrated one.

Recently, Yu et al. (2008) propose a hybrid approach to solve a large scale IRP. An approximate model was developed for a single product multi-period IRP with deterministic demands. The model incorporates a split delivery where deliveries to customers can be made by more than one vehicle. The split delivery was first introduced by Dror and Trudeau (1989) for the VRP by relaxing a constraint of the VRP that every customer is served by only one vehicle. This allows for a better utilization of the vehicles, thus reducing the transportation cost, especially if the mean of customers demand is not too small compared to the vehicle capacity (10% or more). However, it is worth noting that such flexibility obviously adds to the complexity of the problem. The authors design a Lagrangian relaxation method to generate a good solution that is then used to construct a near optimal solution of the IRP by solving a series of assignment problems.

Bertazzi et al. (1997) propose a set of decomposition heuristics for the transportation of multi-product in multi-period with constant demand. Starting from a link by link solution of the problem (i.e. direct shipping), a local search is performed looking for improvement through consolidation. The second phase aggregates customers visited at the same frequency on the same route. The authors introduce the concept of split deliveries where the quantity of a product required at a destination may be split between different shipments, possibly with different frequencies. For simplicity, most multi-product models assume that each retailer requires only one type of product. We note that the same assumption is adopted in our model where each supplier is assumed to supply a distinct product to the assembly plant.

Bertazzi et al. (2002) consider a multi-period model with deterministic demand in which a set of products is shipped from a

supplier to several retailers. They adopted the order up to a level management inventory ( $S,s$ ), where the maximum ( $S$ ) and the minimum ( $s$ ) levels of inventory are specified by each retailer and the products have to be replenished before the minimum level is attained. The quantity of the product delivered is the amount such that the maximum level is reached at the retailer. The authors proposed a two stage heuristic algorithm whereby the first stage focuses on route construction algorithms whilst the second stage attempts to improve the existing solution by performing simple swap operators that aim to remove or insert customers at different positions of the route used by the vehicle. This solution procedure is implemented for a single product and a single vehicle case only though the authors suggested that their approach can easily be modified to cater for the multi-product case.

Lee et al. (2003) tackle a class of IRP where there are multiple suppliers and an assembly plant in an automotive part supply chain. They address the problem as a finite horizon, multi-period, multi-supplier, single assembly plant part-supply network. Each supplier supplies a distinct product and the objective of their study is to minimize the total transportation and inventory cost over the planning horizon. The problem is divided into two subproblems, namely the vehicle routing and the inventory control. The problem is formulated as a mixed integer programming model. They put forward a simulated annealing heuristic to generate and evaluate alternative sets of vehicle routes. Once the routes are determined, the model reduces to a linear programming model (LP) which is solved to obtain the optimum inventory level for each product. The authors also report that the optimal solution is mainly dominated by the transportation cost and is not sensitive to the unit inventory carrying cost. Our work is based on this interesting logistic problem for which some data sets exist (<http://www.mie.utoronto.ca/labs/ilr/IRP/>).

Several models for the case of an infinite horizon have also been developed. These models attempt to minimize the long-run average, or mean average, cost where the lower bound for this cost can be derived. Several IRP with stochastic demands have also been studied, see for example (Campbell et al., 1998; Trudeau and Dror, 1992). For a comprehensive review on IRP, we refer the reader to Moin and Salhi (2007). An interesting recent survey that focuses on the distribution of a single product over a finite planning horizon is given by Bertazzi et al. (2002).

This paper is organized as follows: Section 2 provides an illustrative example of our IRP, followed by a mathematical formulation. The genetic algorithms are introduced in Section 3. An enhancement to the proposed algorithms is described in Section 4, followed by the presentation of our results in Section 5 and our conclusion is given in Section 6.

## 2. Problem description and mathematical formulation

In this study, we consider a many-to-one distribution network similar to the one proposed by Lee et al. (2003). It is a part-supply network of inbound logistic where the network consists of a depot, an assembly plant and  $N$  suppliers where each provides a distinct product to the assembly plant. The problem addressed in this paper is based on a finite horizon, multi-period, multi-supplier, single warehouse, where a fleet of capacitated vehicles, housed at a depot, transports products from the suppliers to meet the demand specified by the assembly plant for each period. The vehicles return to the depot at the end of the trip. In this model, the products are assumed to be ready for collection at the supplier when the vehicle arrives. Also, as any shortage of parts can lead to excessively high costs, backordering/backlogging is not allowed at the assembly plant. However, if the demand for more than one

period is collected, then the inventory is carried forward subject to product-specific holding cost incurred at the assembly plant. The objective is thus to minimize the total transportation and inventory carrying costs over the planning horizon.

We consider the demand at the assembly plant to be product specific, deterministic (known in advance) but variable over time. We also note that it is possible for the demand for a particular product to be zero in a given period. In addition, we consider the inventory cost at the assembly plant but not at the supplier. Product specific holding cost per unit of product per unit time is assumed to be known for each product and that there is no capacity limit at the plant. Finally, the holding cost is assumed to be constant throughout the planning horizon and it is not period specific.

The number of capacitated and identical vehicles in this study is assumed to be unlimited. This allows some flexibility in the solutions as the actual number of vehicles can be determined later. The transportation cost per trip consists of a fixed charge incurred on each trip and a variable cost proportional to the traveled distance. We note that a supplier may be visited more than once in a given period as we allow split pick-ups in this study.

### 2.1. Illustrative example

Fig. 1 illustrates the case of 5 suppliers, 2 periods. In this example we consider the vehicle capacity to be 10. Note that though supplier number 4 requires 3 units in each period, a feasible and efficient schedule (depending on other costs) is to collect directly 6 units in period 1 and not to visit that supplier in period 2, thus saving on the transportation cost.

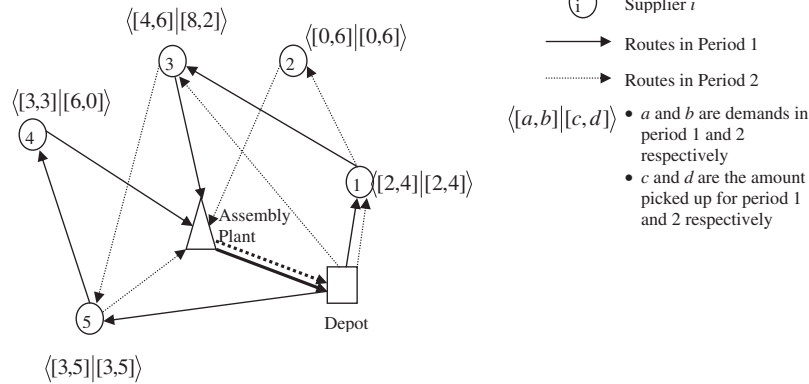


Fig. 1. An example of the pick-up routes for the 5 retailers and the 2-period problem.

### 2.2. Formulation

A mathematical formulation based on Yu et al. (2008) is given here. We first introduce the following notations:

#### Indices

$S = \{1, 2, \dots, N\}$  a set of suppliers where supplier  $i$  ( $i \in S$ ) supplies product  $i$  only.  
 $D = \{0\}$  depot  
 $P = \{N+1\}$  assembly plant  
 $\tau = \{1, 2, \dots, T\}$  period index

#### Parameters

$C$  vehicles capacity  
 $F$  fixed vehicle cost per trip (assumed to be the same for all periods)

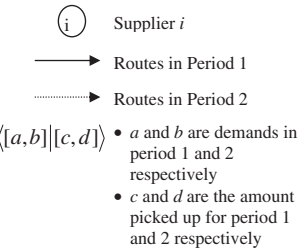
$V$  travel cost per unit distance  
 $M$  size of the vehicle fleet and it is assumed to be  $\infty$  (unlimited)  
 $d_{it}$  demand for product from supplier  $i$  (at the assembly plant) in period  $t$   
 $c_{ij}$  travel distance between supplier  $i$  and  $j$  where  $c_{ij} = c_{ji}$  and the triangle inequality,  $c_{ik} + c_{kj} \geq c_{ij}$ , holds for any  $i, j$  and  $k$  with  $i \neq j$ ,  $k \neq i$  and  $k \neq j$   
 $h_i$  inventory carrying cost at the assembly plant for product from supplier  $i$  per unit product per unit time  
 $I_{i0}$  initial inventory level of product from supplier  $i$  (at the assembly plant) at the beginning of period 1

#### Variables

$a_{it}$  total amount to be picked-up at supplier  $i$  in period  $t$   
 $I_{it}$  inventory level of product from supplier  $i$  at the assembly plant at the end of period  $t$   
 $q_{ijt}$  quantity transported through the directed arc  $(ij)$  in period  $t$   
 $x_{ijt}$  number of times that the directed arc  $(ij)$  is visited by vehicles in period  $t$

The mathematical formulation for our inventory routing problem is given as follows:

$$(P) \quad Z = \min \sum_{i \in S} h_i \left( \sum_{t \in \tau} I_{it} \right) \quad (A)$$



$$+ V \left( \underbrace{\sum_{j \in S \cup D} \sum_{i \in S \cup D} c_{ij} \left( \sum_{t \in \tau} x_{ijt} \right) + \sum_{i \in S} c_{i,N+1} \left( \sum_{t \in \tau} x_{i,N+1,t} \right)}_{(B)} \right) + \underbrace{(F + C_{N+1,0}) \sum_{t \in \tau} \sum_{i \in S} x_{0it}}_{(C)} \quad (1)$$

#### Subject to

$$I_{it} = I_{i,t-1} + a_{it} - d_{it}, \quad \forall i \in S, \forall t \in \tau \quad (2)$$

$$\sum_{\substack{i \in S \cup D \\ i \neq j}} q_{ijt} + a_{jt} = \sum_{\substack{i \in S \cup P \\ i \neq j}} q_{jit}, \quad \forall j \in S, \forall t \in \tau \quad (3)$$

$$\sum_{i \in S} q_{i,N+1,t} = \sum_{i \in S} a_{it}, \quad \forall t \in \tau \quad (4)$$

$$\sum_{\substack{i \in S \cup D \\ i \neq j}} x_{ijt} = \sum_{\substack{i \in S \cup P \\ i \neq j}} x_{jit}, \quad \forall j \in S, \forall t \in \tau \quad (5)$$

$$\sum_{j \in S} x_{ijt} = \sum_{j \in S} x_{jkt}, \quad i \in D, k \in P, \forall t \in \tau \quad (6)$$

$$q_{ijt} \leq Cx_{ijt}, \quad \forall i \in S, \forall j \in S \cup P, i \neq j, \forall t \in \tau \quad (7)$$

$$I_{it} \geq 0, \quad \forall i \in S, \forall t \in \tau \quad (8)$$

$$a_{it} \geq 0, \quad \forall i \in S, \forall t \in \tau \quad (9)$$

$$x_{ijt} \in \{0, 1\}, \quad \forall i, j \in S, \forall t \in \tau \quad (10)$$

$$x_{0jt} \geq 0 \text{ and integer}, \quad \forall j \in S, \forall t \in \tau \quad (11)$$

$$x_{i,N+1,t} \geq 0 \text{ and integer}, \quad \forall i \in S, \forall t \in \tau \quad (12)$$

$$x_{ijt} = 0, \quad i \in D, j \in P, \forall t \in \tau \quad (13)$$

$$x_{ijt} = 0, \quad i \in S, j \in D, \forall t \in \tau \quad (14)$$

$$x_{ijt} = 0, \quad i \in P, j \in S, \forall t \in \tau \quad (15)$$

$$q_{ijt} \geq 0 \quad \forall i \in S, \forall j \in S \cup P, \forall t \in \tau \quad (16)$$

$$q_{0it} = 0 \quad \forall i \in S, \forall t \in \tau \quad (17)$$

The objective function (1) comprises both the inventory costs (A) and the transportation costs (variable travel costs (B) and vehicle fixed cost (C)). We note that the fixed transportation cost consists of the fixed cost incurred per trip and the constant cost of vehicles returning to the depot from the assembly plant. The number of trips in period  $t$  is  $\sum_{i \in S} x_{0it}$ . (2) is the inventory balance equation for each product at the assembly plant whilst (3) is the product flow conservation equations, assuring the flow balance at each supplier and eliminating all subtours. (4) assures the accumulative picked-up quantities at the assembly plant and (5) and (6) ensure that the number of vehicles leaving a supplier, assembly plant or the depot is equal to the number of its arrival vehicles. We note that constraint (6) is introduced because each vehicle has to visit the plant before returning to the depot. (7) guarantees that the vehicle capacity is respected and gives the logical relationship between  $q_{ijt}$  and  $x_{ijt}$  which allows for split pick-ups. (8) ensures that the demand at the assembly plant is completely fulfilled without backorder. The remaining constraints are the nonnegativity constraints imposed on the variables. We note that (13)–(15) ensure that there is no direct link from the depot to the plant, from supplier to the depot and from plant to the suppliers, respectively. We also note that this formulation does not impose the maximum fleet size. However, if an upper bound on the fleet size is known a priori for a given period  $t$ , say  $M$ , then the following constraint  $\sum_{i \in S} x_{0it} \leq M$  can be added.

**Lemma.** *It is possible for a feasible intermediate solution for (P) to contain a route that visits a supplier  $j, j \in S$  more than once in a given period  $t, t \in \tau$ . However, the optimal solution would not have such a subtour.*

**Proof.** Such a route can easily be rectified by removing the arc that will give the minimum total distance and joining the predecessor and the successor of  $j$  to form the final route. The quantity to be picked up on the removed visit can be aggregated accordingly. However, since  $c_{ij}, i, j \in S, i \neq j$  is assumed to satisfy the triangle inequality, an optimal solution to the model will not contain such a route.  $\square$

*Note:* As pointed out in Yu et al. (2008), constraints (3) and (7) together are neither a sufficient condition for defining a feasible

solution of the split pick-up VRP, nor a necessary condition for its subtour elimination. However, the above model provides a lower bound for the IRP with split pick-up since any solution of the IRP with split pick-up is also a solution of the model.

This formulation is used and CPLEX 9.1 is executed within a limited cpu time of 3600 s (1 h). This is used to generate lower and upper bounds that are then used to assess the performance of our GAs which are presented next.

### 3. Genetic algorithms for the IRP

In the last decade, we have seen a growing interest in biologically motivated approaches such as neural networks, evolutionary strategies and genetic algorithms (GAs) being applied to many complex optimization problems. The processes occurring in the natural systems have inspired the development of these algorithms. Evolutionary strategies and genetic algorithms are constructed based on the observation of evolutionary processes in biological systems. Evolutionary processes such as adaptation, selection, reproduction, mutation and competition are closely studied and translated into the form of computer simulations. Although these algorithms are a crude simplification of the natural processes, they have been successfully applied to many complex problems that were once intractable (see for example Goldberg, 1986; Cotta and van Hemert, 2008). GA is a stochastic search technique that explores the problem domain by maintaining a population of individuals, which represents a set of potential solutions in the search space. GA attempts to combine the good features found in each individual using a structured yet randomised information exchange in order to construct individuals who are better suited to their environment than the individuals that they were created from. Through the evolution of better and better individuals, it is anticipated that the desired solution will be found.

#### 3.1. Binary matrix representation

In the first representation, we define a chromosome by a binary matrix of size  $(N \times T)$  where  $N$  is the number of suppliers while  $T$  defines the number of periods. Let

$$Ch^l(i,j) = \begin{cases} 1 & \text{if Chromosome } l \text{ has supplier } i \text{ visited in period } j \\ 0 & \text{otherwise} \end{cases}$$

We note that the amount to be collected depends on whether there will be a collection in the subsequent period or not. Since backordering is not allowed, the total collection from supplier  $i$  in period  $j$  is the sum of all the demands in period  $j, j+1, \dots, k-1$  where the next collection will be made in period  $k$ . As the initial inventory,  $s_{i0}$  for  $i=1, 2, \dots, N$  is assumed to be zero, the values in the first column consist of all ones. However, the algorithm can be adjusted accordingly if the initial inventory for part  $i$  is given or known in advance. If all the initial inventories exceed the first period demands for all the suppliers, then the first column in the chromosome can be generated randomly.

Fig. 2 illustrates an example of the chromosome representation for a problem having 5 suppliers and 5 periods with the demand matrix given in Fig. 2(a). For instance in period 2, only suppliers 1 and 2 are visited as shown in Fig. 2(b). Note that in column 1, all suppliers are visited as expected (due to the assumption that no backlogging is allowed and the initial starting inventory is zero for all suppliers), i.e.  $ch_{i1} = 1, \forall i=1, \dots, 5$ . It can be seen that supplier 1 is visited in period 1 and period 4 where a collection of 10 and 8 units is performed, respectively (see Fig. 2(c)), with a resulting inventory as shown in Fig. 2(d).

### 3.2. Crossover operator

We employ a two dimensional uniform crossover (modified to suit the matrix representation) where a binary mask of size  $(N \times T)$  is generated randomly for each pair of parents. The position of the ones in the binary mask determines the values in the first parent that are transferred to the first offspring and the elements in the position zeros are obtained from the second parent. A complimentary mask is used to deduce the second offspring. The modified uniform crossover is presented in Fig. 3. Note that in both children, period 1 is not affected as all suppliers need to be visited.

For instance, as a value of zero is assigned to crossover mask for supplier 1 in period 2, this Child 1 will inherit the gene from

Parent 2, which is 1. We note that the numbers in *italic* are inherited from Parent 2.

### 3.3. Mutation operator

Mutation is a genetic operator, analogous to the biological mutation, which is used to maintain genetic diversity from one generation of a population of chromosomes to the next. The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. The purpose of mutation in GAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus

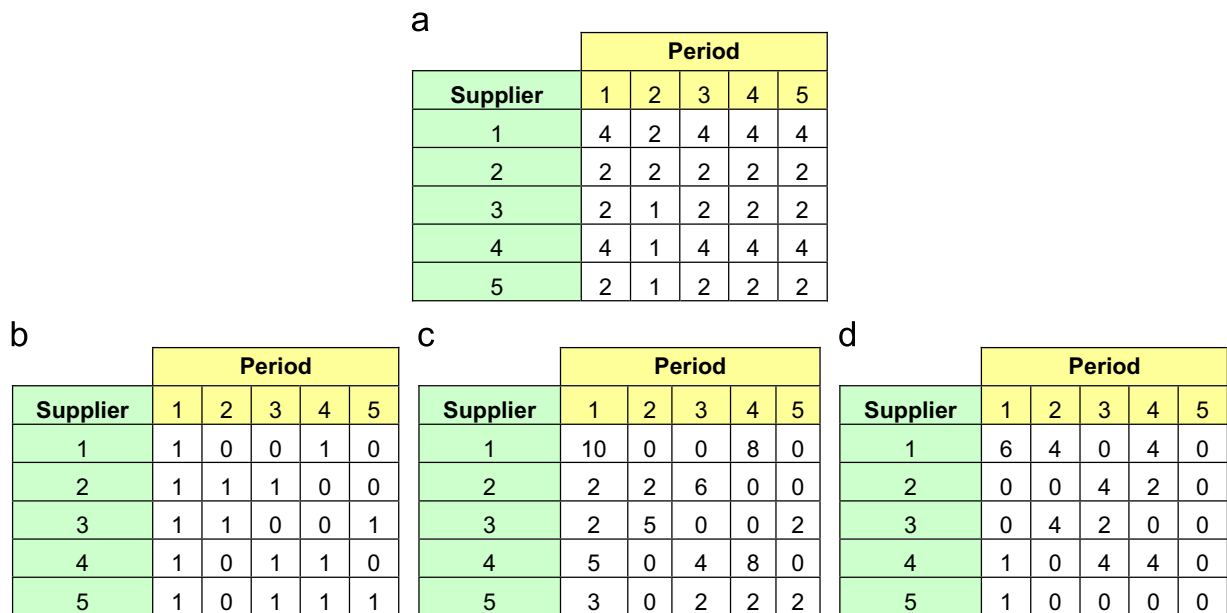


Fig. 2. (a) Demand matrix, (b) a binary chromosome representation, (c) the resulting collection matrix and (d) the resulting inventory matrix.

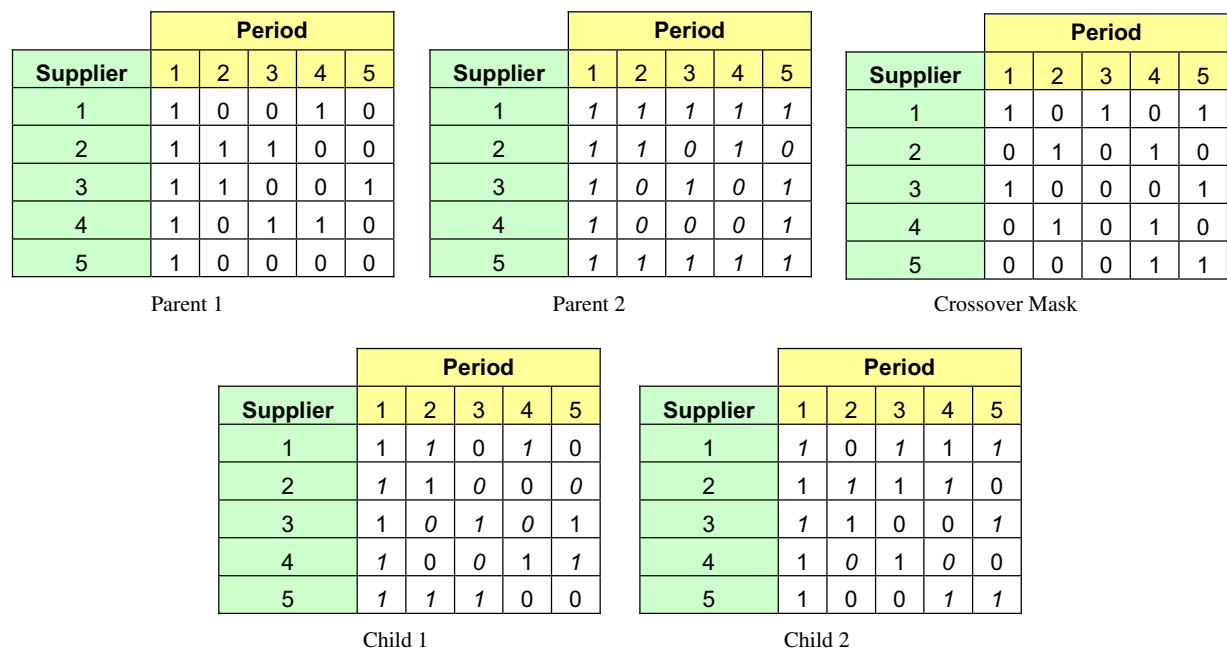


Fig. 3. The modified uniform crossover operator.



slowing or even stopping the evolution. This reasoning also explains the fact that most GA systems tend to avoid taking the fittest of the population only when generating the next chromosome but using rather a random selection (or pseudo-random with a weighting toward those that are fitter).

In the classical mutation operator, which is adopted in this study, a random variable is generated for each bit in a sequence. If this random number is found to be less than the specified mutation rate, then the value of that particular bit will be mutated (in this case the value of one will be changed to a zero and *vice versa*).

### 3.4. Real valued representation

The chromosomes in the second representation encode the collection or pick-up matrices (the amount to be collected) in the form of a 2-dimensional  $N \times T$  matrix. The initial population can be constructed in several ways. In this implementation, the initial population is constructed using the binary representation described in Section 3.1. This step is often known as the preprocessing stage. Here, the binary representation is used to identify the periods when the supplier is to be visited and the amount to be collected from supplier  $i$  in period  $j$  is generated randomly in the interval  $[\sum_{l=j}^{k-1} d_{il}, \sum_{l=j}^k d_{il}]$  where  $d_{il}$  is the demand for product  $i$  (corresponding to supplier  $i$ ) in period  $l$  and the next collection for product  $i$  is in period  $k$ . This is used to allow flexibility in satisfying some or part of the demands if the transportation cost is reduced. Note that there will always be collection in the first period from all the suppliers as the initial inventory,  $s_{i0}$  for  $i=1, 2, \dots, N$  is assumed to be zero. An example of a real representation is given in Fig. 4(a) and the resulting inventory matrix is shown in Fig. 4(b). Note that the binary representation given in Fig. 2(b) is used at the preprocessing stage.

### 3.5. Crossover operator

A new crossover operator based on exchanging the delivery schedules for a selected set of periods (chosen randomly) between the two parents, while ensuring that the resultant children do not violate either the demand or the vehicle's capacity constraints, is proposed. This is similar to the one used by Abdelmaguid and Dessouky (AbdelMaguid and Dessouky (2006) for a different problem except that in their work infeasible solutions are allowed and penalized in their objective function. In our model, all constraints are treated as hard constraints, thus a repair mechanism could have been designed to restore feasibility. However, for this particular combinatorial problem, the repair mechanism is found to be too time consuming and costly.

Firstly, a mask of size  $(N \times 1)$  is randomly generated and it is applied in the same way as the uniform crossover mask defined in the previous subsection. Fig. 5 illustrates the crossover operator. For instance, the first element in the crossover mask has a value of

1, which indicates that the first row in Child 1 is copied directly from the first row of Parent 1.

### 3.6. Mutation operator

We observe that this type of representation tends to produce a higher inventory holding cost. To overcome this weakness we designed a mutation operator that is suitable for this task. The algorithm attempts to transfer some amount of the product picked up in the previous period to the current selected period. If the selected period happens to be the first period, then the amount will be transferred from the selected period to the succeeding period. This will reduce inventory holding cost. The mutation process is described in Fig. 6.

The genetic algorithm for the IRP can generally be divided into four steps which includes (i) the generation of the initial population, (ii) the evaluation of the objective function that comprises the decoding of the chromosome and applying the double sweep algorithm (Lee et al., 2003) to cluster and route the suppliers, (iii) the application of the genetic operators and (iv) finally the formation of a new population. This process iterates until a suitable stopping criterion is met.

The overall algorithm of this hybrid GA, short for HGA, is summarized in Fig. 7.

We note that Step 3a to Step 3c are based on the double sweep proposed by Lee et al. (2003).

## 4. An enhanced implementation

It is observed that the last vehicle in each period normally utilizes less than half of the vehicle's capacity. This increases the number of vehicles used unnecessarily and indirectly increases the transportation cost. To guide the search we propose a coordination of transportation such that the vehicle's utilization is maximized. This is achieved by examining the total amount to be collected by the last vehicle in each period. However, this will consequently result in having to construct the route again and this is very costly in a GA platform. We propose that the excess collection in each period (with respect to the vehicle capacity) is evaluated instead. If the excess is less than  $K\%$ , then the amount of excess will be transferred to the preceding period. We limit the number of periods to be transferred to 2 to ensure that the increase in the inventory holding cost will not be too high. This modification is performed after the decoding stage in Step 2 of Fig. 7. Fig. 8 refers to the modified hybrid GA, which we refer to for short as MHGA.

We note that the maximum amount to be transferred for supplier  $k$  in period  $j$  is not more than  $d_{kj}$ . This will ensure that the resultant inventory holding cost will not increase drastically. If the remaining  $a_{kj}=0$ , then the chromosome will be modified accordingly.

Supplier	Period				
	1	2	3	4	5
1	11	0	0	7	0
2	3	3	4	0	0
3	4	4	0	0	1
4	7	0	3	7	0
5	3	0	3	2	1

Supplier	Period				
	1	2	3	4	5
1	7	5	1	4	0
2	1	2	4	2	0
3	2	5	3	1	0
4	3	2	1	4	0
5	1	0	1	1	0

Fig. 4. (a) Chromosome representation and (b) the resulting inventory matrix.

	Period				
Supplier	1	2	3	4	5
1	11	0	0	7	0
2	2	2	6	0	0
3	2	6	0	0	1
4	7	0	10	0	0
5	4	0	1	2	2

Parent 1

	Period				
Supplier	1	2	3	4	5
1	4	9	0	5	0
2	5	0	1	4	0
3	8	0	0	0	1
4	7	0	10	0	0
5	2	3	0	2	2

Parent 2

Mask = [ 1 0 0 1 1 ]'

The Mask Vector

	Period				
Supplier	1	2	3	4	5
1	11	0	0	7	0
2	5	0	1	4	0
3	8	0	0	0	1
4	7	0	10	0	0
5	2	3	0	2	2

Child 1

	Period				
Supplier	1	2	3	4	5
1	4	9	0	5	0
2	2	2	6	0	0
3	2	6	0	0	1
4	7	0	10	0	0
5	4	0	1	2	2

Child 2

Fig. 5. The crossover operator.

<p><b>Step 1:</b> Select randomly the gene that will undergo the mutation process. Let this gene be <math>gene(i, j)</math> where <math>i</math> and <math>j</math> denote supplier and period respectively.</p>	
<p><b>Step 2:</b> If <math>j \neq 1</math>, go to Step 3. Otherwise, set <math>q = 1</math> where, <math>q</math> is the number of periods before the next collection. If <math>gene(i, j + q) = 0</math> set <math>q = q + 1</math>, and repeat the process until <math>gene(i, j + q) \neq 0</math>. Next, let <math>r = \sum_{k=1}^{j+q} a_{ik} - \sum_{k=1}^{j+q} d_{ik}</math> where <math>a_{ij}</math> and <math>d_{ij}</math> is the collected amount and the demand in period <math>j</math> respectively. Generate the amount to be transferred (say <math>V</math>) randomly in the interval <math>(0, r)</math>. Let <math>gene(i, j) = gene(i, j) - V</math> and <math>gene(i, j + q) = gene(i, j + q) + V</math>.</p>	
<p><b>Step 3:</b> Set <math>p = 1</math> where <math>p</math> is the number of periods since the last collection. If <math>gene(i, j - p) = 0</math>, set <math>p = p + 1</math>. Repeat the process until <math>gene(i, j - p) \neq 0</math>. Let <math>r = \sum_{k=j-p}^j a_{ik} - \sum_{k=j-p}^j d_{ik}</math> and generate randomly <math>V \in (0, r)</math>. Set <math>gene(i, j - p) = gene(i, j - p) - V</math> and <math>gene(i, j) = gene(i, j) + V</math>.</p>	

Fig. 6. The mutation operator.

We have also examined the case where the amount to be transferred, say  $r$ , is chosen deterministically instead of being randomly generated. The amount  $r$  is determined as

$r = \min\{d_{kj}, R_j \times C - \text{sum}\}$  for  $a_{kj} > d_{kj}$  and  $r = \min\{a_{kj}, R_j \times C - \text{sum}\}$  otherwise where  $\text{sum}$  refers to the total amount to be transferred in the preceding period.

**Post-optimization:** The routing within each cluster can obviously be improved using existing refinement procedures. In our implementation, the well known 2-opt originally proposed by Lin (1965) is performed on the best chromosome found. This additional task does not contribute significantly to the cpu times since the number of suppliers within a route is relatively small due to the capacity constraint.

## 5. Results and discussion

The algorithms were written in Matlab using the Genetic Algorithm Toolbox to run the program. All computations were

performed on a 2.8 GHz processor with 4 GB of RAM. The first part of the study examines the effectiveness of all the algorithms when compared to the lower bound obtained by solving the model presented in Section 2 using CPLEX 9.1.

### 5.1. Data sets

We adapt 14 data sets based on the original 4 data sets given in Lee et al. (2003). The original four data sets are S12T14, S20T21, S50T21 and S98T14 that comprises of (12 suppliers, 14 periods), (20 suppliers, 21 periods), (50 suppliers, 21 periods) and (98 suppliers, 14 periods), respectively. We note that  $S \times T$  refers to an instant with  $x$  suppliers and  $y$  periods.

The remaining 11 data sets are created from the 4 original data sets by varying the number of periods to represent small, medium and large size problems. The locations of the suppliers for S12T14, S20T21 and S50T21 are generated randomly in a square of  $100 \times 100$ . The locations of the suppliers for the S20T14 are

**Step 1:** Generate an initial population.

**Step 2:** Decode the chromosome according to the procedure previously described. Perform the following procedure (double sweep):

**Step 3a:** For each period  $j$ ,  $j=1,2,\dots,T$ , arrange the suppliers  $\{s_1, s_2, \dots, s_m\}$  and the assembly plant  $\{s_{m+1}\}$  around the depot ( $s_0$ ) such that the  $y$  coordinate of the assembly plant is the same as the coordinate of the depot. Sort the suppliers  $\{s_1, s_2, \dots, s_m\}$  in ascending order according to their new  $y$  coordinate values. Let  $s_{(i)}$  be the  $i$ th supplier after the sort. Set  $i=1$  and  $k=1$ . Open a route  $R_k = \{\}$  and set  $Q_k = 0$ , where  $Q_k$  is the total pick-up quantity assigned to cluster  $k$ .

**Step 3b:** If  $Q_k + d_{ij} \leq C$ , assign  $s_{(i)}$  to route  $R_k$ , set  $Q_k = Q_k + d_{ij}$  and  $a_{ik} = d_{ij}$ . Otherwise set  $a_{ik} = C - Q_k$  and  $Q_k = C$ . Set  $k=k+1$  and open a new route  $R_k = \{\}$ , assign  $s_{(i)}$  to  $R_k$ . Set  $a_{ik} = d_{ij} - a_{ik-1}$  and  $Q_k = a_{ik}$ . If  $i > m$ , set  $k=1$  and go to Step 2c. Otherwise, set  $i=i+1$  and repeat Step 2b.

**Step 3c:** [Routing] Sort the suppliers within route  $R_k$  according to their  $x$ -coordinate values in ascending order. Let  $s_{(i)}^k$  be the  $i$ th supplier in route  $R_k$  after the sort. Form a route that starts at the depot  $s_0$ , visit suppliers  $s_{(1)}^k, s_{(2)}^k, \dots, s_{(R_k)}^k, s_{m+1}$ , and returns to  $s_0$ . Evaluate the total objective function value as given by Equation (1).

**Step 4:** Perform crossover and mutation

**Step 5:** Repeat Step 2 – Step 3 until the maximum number of generations is attained.

Fig. 7. The overall hybrid GA (HGA).

**Step 1-2:** As Step 1-2 in Figure 7.

**Step 3:**

**Step 3A:** For each period  $j$ ,  $j=1,2,\dots,T$ , let  $DP_j = \sum_{i=1}^{|P_j|} d_{ij}$  where  $P_j \subseteq N$  is the set of suppliers visited in period  $j$  and  $DP_j$  is the total collection in period  $j$ . Calculate the percentage excess (with respect to vehicle capacity) in each period as,  $R_j = \frac{\text{rem}(DP_j, C)}{C} \times 100$  where  $C$  is the vehicle capacity.

**Step 3B:** Starting from  $j=T$  (the last period), if  $R_j < K\%$ , sort in ascending order of the holding cost, the supplier  $\{s_1, s_2, \dots, s_m\}$   $P_j$  and let  $s_{(k)}$  be the  $k$ th supplier after the sort. Otherwise set  $j=j-1$  and repeat Step 3B.

**Step 3C:** Set  $q=1$ ,  $sum=0$  and  $s=1$  where  $q$  ( $q \leq 2$ ) is the number of periods since the last collection,  $sum$  is the total amount to be transferred to the preceding period ( $s$ ) and  $s$  ( $s=|P_j|$ ) is the number of suppliers that will be visited in period  $j$ . Starting from  $s_{(k)}$ ,  $k=1$ , if  $a_{kj} > 0$  and  $a_{k(j-q)} > 0$ , then if  $a_{kj} > d_{kj}$ , set  $r = \text{random}(0, \min\{d_{ij}, R_j \times C - sum\})$ . Otherwise, let  $r = \text{random}(0, \min\{a_{kj}, R_j \times C - sum\})$ , set  $sum = sum + r$ ,  $a_{kj} = a_{kj} - r$ ,  $a_{k(j-q)} = a_{k(j-q)} + r$ ,  $s = s + 1$  and repeat until  $sum = R_j \times C$ .

**Step 4:** As in Step 3 (Steps 3a-3c) – Step 5 in Figure 7.

Fig. 8. The modified hybrid GA (MHGA).

extended from the S12T14 data sets by adding 10 new suppliers. Similarly, the S50T21 suppliers are extended from the S20T21 locations and generating randomly the locations of an additional 30 suppliers. The S98T14 is based on a real life data and the suppliers are closely located.

All the data sets, with the exception of S50T21, have demands in every period. Some suppliers in the S50T21 data set may not receive the demand for their product until the later periods. As some of the data sets are extracted from S50T21, it is possible that the demand for some of the product is zero. Data sets S50T5 and



S50T10 consist of products with zero demand. The suppliers of these products can be effectively eliminated from the representation as they will never be visited in the planning horizon.

We note that the demands for S98T14 are given in real values and the amount varies significantly between each product. The cost per unit distance, fixed cost and the vehicles' capacity are increased to 50, 200 and 400, respectively. It is also noted that the demand for each product for this data set is constant as this is a common feature in the automotive industry.

Table 1 provides the characteristics of the data sets. We also note that the depot is located at (0,0) for all the data sets.

## 5.2. Results

For all the instances, we let CPLEX run for a time limit of 3600 s when we record the lower bound and the best integer solution found. In our implementation of the GAs, the number of generations, the generation gap and the crossover rate are fixed at 300, 0.9 and 0.7, respectively, for all problems. The mutation rate for all the algorithms is fixed at 0.001 with the exception of the real representation. The mutation rate for this algorithm is fixed at 0.1 as our limited experiments indicate that this algorithm performs better with higher mutation rates. The population size is fixed at 200 individuals. We note that the maximum number of generations for the data sets with 98 suppliers is increased to 600 because of the large data size. All other parameters were kept the same.

**Table 1**  
Characteristics of the data sets.

Data set	S12T14	S20T21	S50T21	S98T14
Fixed cost ( $F$ )	20	20	20	200
Cost per unit traveling distance	1	1	1	50
Vehicle capacity	10	10	10	400
Range of holding costs	[3,27]	[3,27]	[1,9]	[1,44]
Range of demand for each product	[1,4]	[1,4]	[0,9]	[0.04,393.33]
Coordinate of assembly plant	(10,20)	(10,20)	(10,20)	(42.31,83.17)

**Table 2**  
Best total costs, no. of vehicles and CPU (s) for all the algorithms.

Data set	(N, $\tau$ )	CPLEX (after 3600 s)			HGA1 <sup>a</sup>			HGA2 <sup>b</sup>			MHGA1 <sup>c</sup> (random shift)			MHGA2 <sup>d</sup> (fixed shift)		
		LB	Best integer	No. of veh.	Best objective	No. of veh.	CPU (s)	Best objective	No. of veh.	CPU (s)	Best objective	No. of veh.	CPU (s)	Best objective	No. of veh.	CPU (s)
S12T5	(12,5)	1650	1881	14	2099.31	14	46.66	<b>2096.75</b>	14	58.48	2099.31	14	48.63	2099.31	14	48.42
S12T10	(12,10)	3218	3797	28	<b>4333.27</b>	29	111.23	4350.99	29	56.70	<b>4333.27</b>	29	93.44	<b>4333.27</b>	29	101.09
S12T14	(20,14)	4709	5645	40	<b>6115.19</b>	41	120.31	6172.04	41	157.17	<b>6115.19</b>	41	123.01	6131.72	41	129.61
S20T5	(20,5)	2607	2895	+	<b>3143.39</b>	21	31.81	3170.68	21	85.52	3178.16	21	68.09	3175.46	21	133.33
S20T10	(20,10)	5227	6080	+	6543.08	44	65.81	6720.64	44	169.42	<b>6499.40</b>	43	126.03	6620.90	44	127.30
S20T14	(20,14)	7181	8772	64	<b>9208.43</b>	61	360.33	9571.85	62	237.31	9243.23	61	177.08	9287.64	62	179.45
S20T21	(20,21)	10717	14093	+	<b>13948.41</b>	92	255.83	14462.34	96	362.02	14028.48	93	273.27	14024.35	93	434.21
S50T5	(50,5)	4547	5071	46	5681.58	45	105.63	5633.37	45	217.68	<b>5618.09</b>	45	133.40	5705.55	45	125.57
S50T10	(50,10)	9289	11910	102	11906.00	95	213.92	11986.02	96	408.44	11940.23	95	269.23	<b>11642.00</b>	95	226.01
S50T14	(50,14)	13193	18264	150	17143.77	136	307.93	17477.05	137	303.75	17155.62	135	340.24	<b>16987.00</b>	135	328.07
S50T21	(50,21)	20185	29975	248	<b>26448.77</b>	209	496.72	27034.00	210	723.95	26458.80	209	563.48	26450.18	208	506.32
S98T5	(98,5)	544036	604205	53	561592.59	57	609.45	564531.95	57	113.85	561899.63	57	477.24	<b>561168.21</b>	57	476.77
S98T10	(98,10)	NA <sup>e</sup>	NA	NA	<b>1124797.57</b>	113	1307.26	1132874.15	114	214.52	1125295.96	114	1040.20	1125398.21	114	1071.79
S98T14	(98,14)	NA	NA	NA	<b>1571652.32</b>	159	1589.71	1596783.40	161	310.83	1574542.60	159	1297.93	1573987.75	159	1300.50

<sup>a</sup> Hybrid GA with binary representation.

<sup>b</sup> Hybrid GA with real representation.

<sup>c</sup> Modified hybrid GA for binary representation with randomly generated amount for  $K=40$ .

<sup>d</sup> Modified hybrid GA for binary representation with fixed amount for  $K=40$ .

<sup>e</sup> The results are not available.

We performed 10 runs for each algorithm on each data set. Table 2 summarizes the best total costs, the number of vehicles and the cpu time for each of our algorithms along with the lower bound and the best integer solutions (upper bound) obtained from CPLEX. The solutions in bold are the best of the 4 algorithms and an '\*' shows that the solutions are better than the upper bound obtained by CPLEX. The mean and standard deviation of the total cost over the 10 runs are also shown in Table 3.

We note that CPLEX terminates prematurely in 3 of the data sets (indicated by '+') due to memory usage. CPLEX did not provide any solution for the 2 large problems (S98T10 and S98T14), even after 7200 s (2 h) of CPU time. In addition, it was not able to find the optimum solution in any of the instances within the 3600 s time limit. It is observed that the gaps, calculated as the ratio of the difference between the upper bound and the lower bound to the lower bound, for all the solutions obtained by CPLEX are more than 10%. This ratio grows drastically as the number of periods and the number of suppliers increase. Therefore it is very hard to judge the quality of the lower bound obtained by CPLEX as this may be because the lower bound is really loose or the upper bound found is rather poor.

In small instances the upper bound found by CPLEX within the time limit outperforms the GAs results. Good solutions (the gap between the upper bound and the lower bound is less than 15%) are obtained in cases where the number of periods is 5. However, the best solutions for all our algorithms were found in significantly less cpu times. As expected, GA based algorithms performed relatively much better for larger instances. In addition, in almost all problem instances except for the S12T5 data set, the binary representation produced better solutions than the ones generated using the real representation. The solutions obtained by the binary representation HGA1 and the modified algorithm (MHGA1 and MHGA2) are not significantly different from each other, with the modified algorithms outperforming HGA1 slightly on 5 instances (2 by MHGA1 and 3 by MHGA2). All our algorithms produced significantly good solutions as the gap between the best solution and the lower bound for S98T5 is less than 3.5% besides requiring a computational time which is less than 30 min

**Table 3**

The mean and standard deviation of the total costs over 10 runs.

Data Set	HGA1		HGA2		MHGA1 (random shift)		MHGA2 (fixed shift)	
	Average objective	Std. Dev.	Average objective	Std. Dev.	Average objective	Std. Dev.	Average objective	Std. Dev.
S12T5	2122.50	17.14	2129.24	12.50	2122.57	14.19	2124.04	13.90
S12T10	4358.07	20.41	4403.37	43.77	4360.40	21.61	4355.35	22.09
S12T14	6150.06	24.12	6221.20	31.42	6151.08	16.50	6173.24	24.25
S20T5	3238.61	64.56	3222.87	33.05	3260.82	75.63	3284.2	65.24
S20T10	6674.98	60.46	6784.50	70.13	6660.13	72.34	6706.88	44.68
S20T14	9347.35	78.83	9659.57	88.47	9373.75	85.74	9354.89	59.89
S20T21	14160.60	115.51	14659.93	158.35	14136.47	92.08	14163.00	78.79
S50T5	5831.02	91.73	5686.09	48.22	5711.63	79.82	5773.86	65.98
S50T10	12059.41	97.60	12168.11	113.83	12076.03	73.97	12128.50	81.16
S50T14	17294.73	99.98	17652.97	153.28	17321.55	114.46	17337.33	113.44
S50T21	26678.06	127.75	27294.92	126.47	26625.07	117.25	26591.69	85.04
S98T5	563839.80	1464.09	5567351.21	3666.85	563271.00	1381.35	563741.31	1795.23
S98T10	1129545.27	3028.44	1141130.82	5452.87	1128558.29	2700.65	1128072.18	2334.86
S98T14	1580344.45	3273.84	1600426.66	4241.72	1580339.90	1260.47	1581465.51	5548.88

(1800 s). Table 3 shows that the standard deviations over the 10 runs for the four algorithms are comparatively small except for the larger instances that can be due to the maximum number of generations being not sufficiently large.

## 6. Conclusion and future work

Both transportation and inventory costs ought to be considered concurrently in the logistic planning functions as these two areas might lead to significant gains and more competitive distribution strategies. In this paper, we present two representations using GA for the inventory routing problem on a finite horizon, multi-period and multi-product problem. We presented two types of modifications for the binary representations in order to overcome some of the limitations which we encountered. The binary representation and the modified binary representations outperformed the real representation in 13 out of 14 instances studied. With the increase of problem size especially for the problems with 98 suppliers, the performance of the GA based algorithms increases and the results were obtained in significantly less computational times. In these particular instances, the suppliers are closely located which is most appropriate for consolidated transportation strategy.

The current algorithms do not incorporate powerful route improvement procedures though a simple 2-opt procedure is implemented at the end. The current algorithms give more emphasis on the benefit of consolidating the transportation to reduce the overall transportation cost and the inventory cost. However, reducing the routing cost can significantly reduce the total cost as it constitutes a large part of the total cost. It may therefore be interesting to dynamically use post-optimization at various generations and on specific chromosomes. This adaptive strategy is worth exploring further. The studied problem and the developed GA based heuristics can also provide interesting insights for solving other problems, especially in an outbound logistics where the demand pattern from one period to the other changes significantly.

## Acknowledgements

The authors would like to thank the referees for their constructive comments that improve the content as well as the presentation of the paper. The first author is also grateful to the

Ministry of Higher Education, Malaysia, and University Putra Malaysia for payment of the registration fees through Fundamental Grant FRGS (02-10-07-320FR) and University of Malaya for payment of traveling expenses.

The revised work was carried out while the first author was on sabbatical at the University of Kent, United Kingdom.

## References

- AbdelMaguid, T.F., Dessouky, M.M., 2006. A genetic algorithm approach to the integrated inventory–distribution problem. *International Journal of Production Research* 44, 4445–4464.
- Bertazzi, L., Speranza, M.G., Ukovich, W., 1997. Minimization of logistic costs with given frequencies. *Transportation Research—B* 31 (4), 327–340.
- Bertazzi, L., Paletta, G., Speranza, M.G., 2002. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science* 36 (1), 119–132.
- Campbell, A., Savelsbergh, M., Clarke, L., Kleywegt, A., 1998. The inventory routing problem. In: Crainic, T.G., Laporte, G. (Eds.), *Fleet Management and Logistics*. Kluwer Academic Publishers, pp. 95–112.
- Chien, T.W., Balakrishnan, A., Wong, R.T., 1989. An integrated inventory routing allocation and vehicle routing problem. *Transportation Science* 23 (2), 67–76.
- Cotta, C., van Hemert, 2008. *Studies in Computational Intelligence: Recent Advances in Evolutionary Computation for Combinatorial Optimization*, 153. Springer-Verlag.
- Dror, M., Ball, M., 1987. Inventory/routing: reduction from an annual to a short period problem. *Naval Research Logistics Quarterly* 34 (6), 891–905.
- Dror, M., Ball, M., Golden, B., 1985. Computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research* 4, 3–23.
- Dror, M., Trudeau, P., 1989. Savings by split delivery routing. *Transportation Science* 23, 141–145.
- Federgruen, A., Zipkin, P., 1984. A combined vehicle routing and inventory allocation problem. *Operations Research* 32, 1019–1036.
- Federgruen, A., Prastacos, G., Zipkin, P.A., 1986. An allocation and distribution model for perishable products. *Operations Research* 34, 75–82.
- Fisher, M., Greenfield, A., Jaikumar, R., Kedia, P., 1982. Real-time scheduling of a bulk delivery fleet: practical application of lagrangean relaxation. Technical Report, October 1982, The Wharton School, University of Pennsylvania, Department of Decision Sciences.
- Goldberg, D.E., 1986. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company Inc.
- Lee, C.-H., Bozer, Y.A., White III C.C., 2003. A heuristic approach and properties of optimal solutions to the dynamic inventory routing problem. Working Paper, University of Toronto, Toronto, Canada.
- Lin, S., 1965. Computers solutions of the traveling salesman problem. *Bell System Technical Journal* 44, 2245–2269.
- Moin, N.H., Salhi, S., 2007. Inventory routing problems: a logistical overview. *Journal of Operational Research Society* 58, 1185–1194.
- Ribeiro, R., Laureço, H.R., 2002. Inventory-routing model, for a multi-period problem with stochastic and deterministic demand. UPF Economics and Business Working Paper.
- Trudeau, P., Dror, M., 1992. Stochastic inventory routing: route design with stockouts and route failures. *Transportation Science* 26 (3), 171–184.
- Yu, Y., Chen, H., Chu, F., 2008. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research* 189, 1022–1040.