

# A genetic algorithm approach to the integrated inventory-distribution problem

TAMER F. ABDELMAGUID†, MAGED M. DESSOUKY\*‡

†Cairo University, Faculty of Engineering, Mechanical Design and Production Dept., Giza 12613, Egypt

‡Daniel J. Epstein Department of Industrial and Systems Engineering, 3715 McClintock Ave, University of Southern California, Los Angeles, CA 90089-0193, USA

Correspondence      \*Corresponding author. Email: [maged@usc.edu](mailto:maged@usc.edu)

We introduce a new genetic algorithm (GA) approach for the integrated inventory distribution problem (IIDP). We present the developed genetic representation and use a randomized version of a previously developed construction heuristic to generate the initial random population. We design suitable crossover and mutation operators for the GA improvement phase. The comparison of results shows the significance of the designed GA over the construction heuristic and demonstrates the capability of reaching solutions within 20% of the optimum on sets of randomly generated test problems.

*Keywords:* Inventory routing; Inventory management; Vehicle routing; GA; Lot sizing.

## 1. Introduction

In the last few years, new ideas of centralized supply chain management, such as vendor managed inventory (VMI), have been widely accepted in many supply chain environments. The idea of centralized supply chain management is that suppliers get direct access to the customers' inventory positions and make the necessary replenishment decisions. This lead to the interest of studying integrated models that combine transportation and inventory decisions. Such an integrated model is intended to optimize the replenishment decisions conducted by the supplier in order to minimize the overall inventory and transportation costs.

In their literature review article, Baita *et al.* (1998) use the term 'dynamic routing and inventory (DRAI)' to refer to the class of problems in which simultaneous vehicle routing, as a transportation problem, and inventory decisions are present in a dynamic framework. They classify the approaches used for DRAI problems into two categories. The first one operates in the frequency domain where the decision variables are replenishment frequencies, or headways between shipments. Examples in the literature include the work of Blumenfeld *et al.* (1985), Hall (1985), Daganzo (1987), and Ernst and Pyke (1993) (for more references see Daganzo, 1999).

The second category, referred to as the time domain approach, uses discrete time models to determine delivery quantities and vehicle routes at fixed time intervals. Within this category the most famous problem is the inventory routing problem (IRP), which arises in the application

of the distribution of industrial gases. The main concern for this kind of application is to maintain an adequate level of inventory for all the customers and to avoid any stockout. In the IRP, it is assumed that each customer has a fixed demand rate and the focus is on minimizing the total transportation cost; while inventory costs are generally not of concern. Examples of this application in the literature include Bell *et al.* (1983), Golden *et al.* (1984), Dror *et al.* (1985), Dror and Ball (1987) and recently Campbell *et al.* (2002).

In the literature, the integration of vehicle routing and inventory decisions with the consideration of inventory costs in the time domain approaches of the DRAI problems has taken different forms. In a few cases a single period planning problem has been addressed as found in Federgruen and Zipkin (1984) and Chien *et al.* (1989). In the multi-period problem, the decisions are conducted for a specific number of planning periods, or the problem is reduced to a single period problem by considering the effect of the long term decisions on the short term ones. Examples include Dror and Ball (1987), Trudeau and Dror (1992), Viswanathan and Mathur (1997), and Herer and Levy (1997).

In this paper, we study the integrated inventory distribution problem (IIDP), which is classified as a time domain DRAI problem. The IIDP considers multiple planning periods, both inventory and transportation costs, and a situation in which backorders are permitted. Backorder decisions are generally justified in two cases. The first is when there is insufficient vehicle capacity to deliver to a customer. The second case is when there is a transportation cost saving that is higher than the incurred backorder cost by a customer.

The IIDP is NP-hard in the strong sense due to its vehicle routing element along with its interaction with inventory decisions. Abdelmaguid (2004) proposed a construction heuristic for the IIDP, called the approximate transportation costs heuristic (ATCH). The construction solution heuristic is based on guiding principles from the uncapacitated lot-sizing problems, which assume inventory replenishment decisions only when the customer's inventory reaches zero. Using these principles, the heuristic generates solutions in which delivery schedules cover customers' exact demand requirements in future days. That is, partial fulfilment of a customer's demand in a future day is not considered by the heuristic. This approximation is reasonable when each individual customer order quantity is significantly less than the vehicle capacity since neglecting these partial demand fulfilments facilitates the decisions involved and results in significant reductions in transportation costs. However, this strategy can generate poor solutions when the customer order quantities are not significantly less than the vehicle capacity. Thus, in this paper, we introduce a genetic algorithm (GA) approach for improving the construction solution that allows for partial deliveries.

Genetic algorithm is a randomized search technique that is based on ideas from the natural selection process (Goldberg, 1989). Starting from an initial set of solutions, generations of new solutions are obtained through applying neighbourhood search operators (crossover and mutation). These operators are applied to randomly selected solutions from the current set of solutions, where the selection probability is proportional to the solutions objective function value. GAs have been successfully implemented to a wide range of combinatorial optimization problems (Gen and Cheng 1997).

This paper is organized as follows. The formal problem definition is presented in Section 2. In Section 3, the developed GA representation for the IIDP is illustrated. In Section 4, the random generation procedure of the initial population is presented. In Sections 5 and 6, the designs of the crossover and mutation operators are illustrated. The experimentation and results are provided in Section 7, followed by the conclusion in Section 8.

## 2. Problem statement

The IIDP was formulated earlier as a Mixed Integer Programming (MIP) model by Abdelmaguid (2004). Here, we present only an abbreviated version of the model with transportation costs represented as a non-linear function.

We study a distribution system consisting of a depot, denoted 0, and geographically dispersed customers, indexed  $1, \dots, N$ . Each customer  $i$  faces a different demand  $d_{it}$  per time period  $t$  (day/week), maintains its own inventory up to capacity  $C_i$ , and incurs inventory holding cost of  $h_i$  per period per unit and a backorder penalty of  $\pi_i$  per period per unit on the end of period inventory position. We assume that the depot has sufficient supply of items that can cover all customers' demands throughout the planning horizon. The planning horizon considers  $T$  periods.

The amount of delivery to customer  $i$  in period  $t$ ,  $w_{it}$ , is to be decided, and based on the delivery amounts in period  $t$ ,  $W_t = (w_{it} : i = 1, \dots, N)$ , transportation costs can be calculated. The transportation costs are represented in the form of a nonlinear function  $TR_t(W_t)$ . Appendix A shows an MIP formulation which integrates all the vehicle routing costs and constraints in the computation of  $TR_t(W_t)$ . The total planned delivery amounts for the customers in a given period are restricted by the total vehicle capacity  $Q$ . The remaining decision variables include the end of period inventory,  $I_{it}$ , and backorder,  $B_{it}$ , at period  $t$  for customer  $i$ . The following is a nonlinear programming formulation for the problem.

[IIDP-NLP] – *Integrated inventory distribution problem – nonlinear programming model*

$$\text{Min } \sum_{t=1}^T \left[ TR_t(W_t) + \sum_{i=1}^N (h_i I_{it} + \pi_i B_{it}) \right] \quad (1)$$

subject to:

$$I_{it-1} - B_{it-1} - I_{it} + B_{it} + w_{it} = d_{it} \quad i = 1, \dots, N \text{ and } t = 1, \dots, T \quad (2)$$

$$\sum_{i=1}^N w_{it} \leq Q \quad t = 1, \dots, T \quad (3)$$

$$I_{it} \leq C_i \quad i = 1, \dots, N \text{ and } t = 1, \dots, T \quad (4)$$

$$I_{it}, B_{it}, w_{it} \geq 0 \quad i = 1, \dots, N \text{ and } t = 1, \dots, T \quad (5)$$

The objective function (1) includes transportation costs and inventory holding and shortage costs on the end inventory positions. Constraints (2) are the inventory balance equations for the customers. Constraints (3) limit the total amount delivered in a given period to the available vehicle capacity. Constraints (4) limit the inventory level of the customers to the corresponding storage capacity. Constraints (5) are the domain constraints.

### 3. Genetic representation

In GAs, each solution is represented in the form of a finite length array called chromosome. A chromosome is composed of a set of locations known as genes that assume discrete values pertaining to the problem's solution. A critical point when applying GAs to an optimization problem is to find a suitable coding scheme that transforms feasible solutions into representations amenable to a GA search and reversibly decode those representations.

#### 3.1. *The proposed genetic representation*

In the proposed genetic representation, we concentrate on the delivery schedule and leave the vehicle routing part to be solved using any efficient polynomial time heuristic such as the savings algorithm (Clarke and Wright, 1964). The delivery schedule is represented in the form of a two-dimensional matrix in which each cell contains the amount to be delivered to a customer in a given period. Each row in the matrix corresponds to a specific customer and the columns represent the planning periods from 1 to  $T$ . To simplify the genetic search process, the delivery amounts are set to be integers. This condition does not restrict the GA from reaching optimal solutions given that the customer demand values and the customer storage and vehicle capacities are integers, which is assumed without loss of generality in this study.

The proposed GA representation satisfies the necessary conditions of successful GA implementation by being as minimal as completely expressive. As shown in the nonlinear programming formulation, the delivery amounts are the key decision variables whose values can be easily used to determine the rest of the decision variables, including vehicle tours. Accordingly, the sole use of the delivery amounts in the representation suffices and satisfies the condition of being minimal. Meanwhile, the representation is capable of representing every possible solution in the search space including the optimal ones (given that the optimal solution of the VRP subproblem is obtainable).

#### 3.2. *Illustrative example*

To illustrate the proposed GA representation, consider a small sample problem in which the distribution system consists of 4 customers with storage capacity of 50 units each, and inventory holding and shortage costs given in table 1. At the beginning of the planning period, all customers have zero inventory positions. Two vehicles, each with 100 units of capacity, are available to serve the customers in every day for a 4-day planning horizon. When a vehicle is decided to be used in any period, \$10 are charged. The distribution network and transportation costs between locations are depicted in the graph shown in figure 1. The demand requirements for every day in the planning horizon are also provided in figure 1.

Costs	Customer			
	1	2	3	4
Unit holding cost (\$/unit/period)	0.09	0.09	0.13	0.1
Unit shortage cost (\$/unit/period)	2.8	3.4	2.7	3.2

Table 1. Cost information for the sample problem

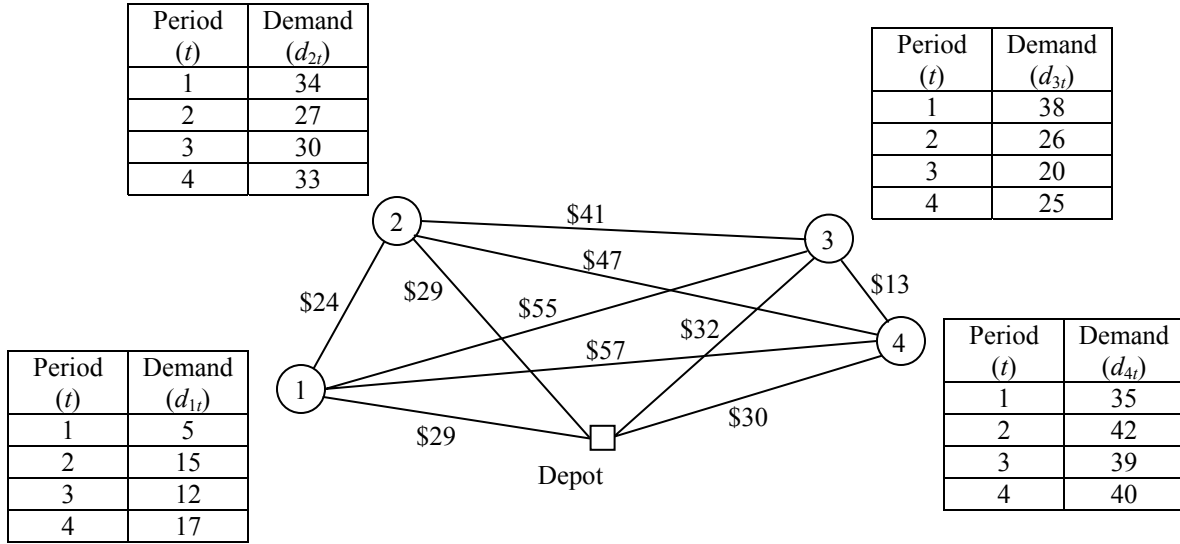


Figure 1. Distribution network and demand requirements for the sample problem

(a) Genetic representation of a sample solution

		Period			
		1	2	3	4
Customer	1	0	49	0	0
	2	61	0	63	0
	3	64	0	45	0
	4	35	81	0	40

(b) Resultant end of period inventory positions

		Period			
		1	2	3	4
Customer	1	-5	29	17	0
	2	27	0	33	0
	3	26	0	25	0
	4	0	39	0	0

(c) Generated vehicle tours based on the provided delivery amounts in the chromosome

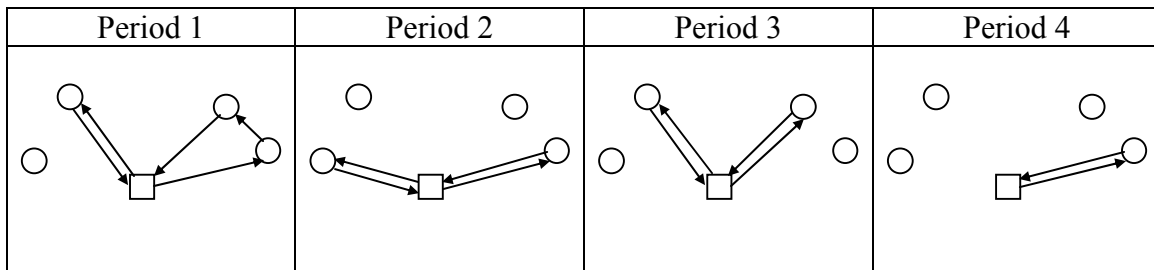


Figure 2. Genetic representation and solution interpretation for the sample solution

The genetic representation of this sample problem takes the form of a 2-dimensional matrix with four rows and four columns. Each cell in the matrix defines the scheduled delivery amount for the corresponding customer (given in the row) and the corresponding period (given in the column). Figure 2 illustrates a sample solution for this sample problem. Based on the scheduled delivery amounts, the inventory position (variables  $I_{it}$  and  $B_{it}$ ) of each customer can be easily determined as shown in figure 2b. An efficient Savings algorithm, which takes the delivery amounts provided in the solution chromosome as input, generates the necessary vehicle tours for each period in the planning horizon and hence provides values for the transportation costs. Accordingly, the value of the objective function of any given chromosome can be determined.

#### 4. Initialization

The GA initialization process, in which a set of initial random solutions are generated, is conducted using a randomized version of the Approximate Transportation Costs Heuristic (ATCH) developed by Abdelmaguid (2004). The ATCH, illustrated in figure 3, is based on the idea of estimating a transportation cost value for each customer in every period in the planning horizon. The delivery amounts are determined by comparing these estimates with inventory holding and shortage costs that could be incurred as a result of delivery decisions made in a given period. The comparison between the estimated transportation costs,  $TR_{i,\tau}$  for customer  $i$  in period  $\tau$  and inventory holding and shortage costs is separated into deciding whether to have backorders on period  $t$  (subproblem SUB1) and whether to use remaining vehicle capacity to cover future customer demand (subproblem SUB2). In the randomized version of the ATCH, the solution of these subproblems is done in a random fashion as described in the following subsection.

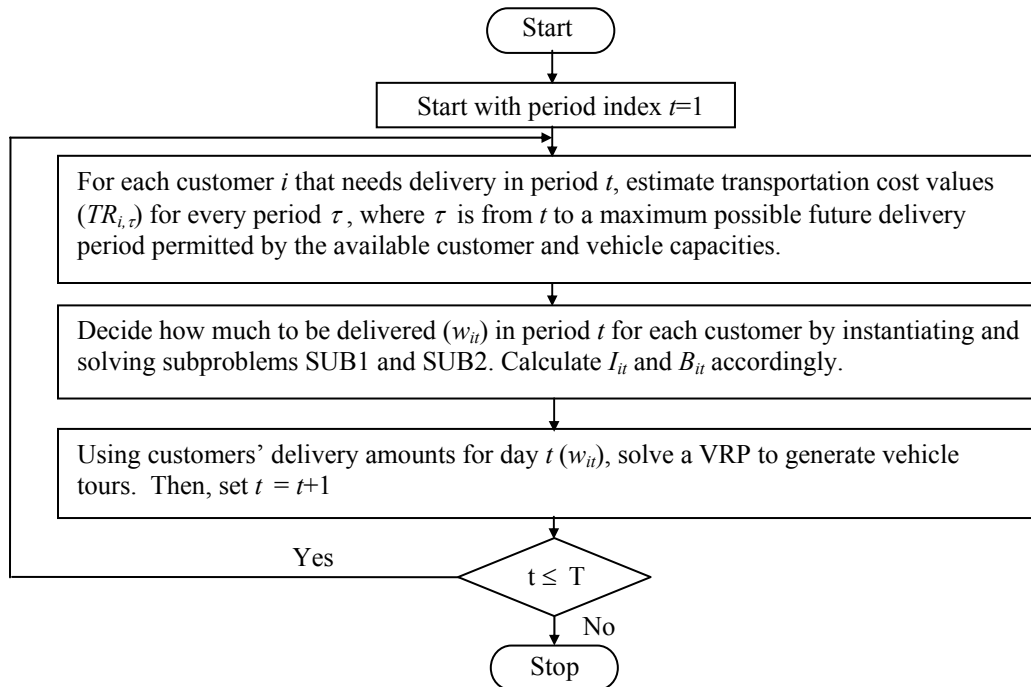


Figure 3. Outline of the ATCH

#### 4.1. Randomized solution approaches for the subproblems

Let  $\delta_{i,t} = d_{i,t} - I_{i,t-1} + B_{i,t-1}$  be the outstanding demand of customer  $i$  at the beginning of period  $t$ , and let  $ND$  be the set of customers that have  $\delta_{i,t} > 0$ . For customers that have  $\delta_{i,t} \leq 0$ , no delivery is needed, and accordingly their  $w_{it} = 0$ . The following list describes the main steps of the randomized version of the algorithm used to solve subproblem SUB1.

##### Procedure SUBALG1-R

1. For every customer  $i \in ND$ , if  $\delta_{i,t} \leq TR_{i,t} / \pi_i$  then with a probability of 0.5, randomly decide whether to remove customer  $i$  from  $ND$  or not;
  2. Update  $TR_{i,t}$  values;
  3. Let  $\Delta Q = \sum_{i \in ND} \delta_{i,t} - Q$ ;
  4. Sort customers in set  $ND$  in an increasing order of their  $\pi_i$  values;
  5. For each customer  $i$  in the ordered set  $ND$  do
    - If  $\Delta Q \geq \delta_{i,t}$ , let  $\Delta Q = \Delta Q - \delta_{i,t}$ ,  $w_{it} = 0$ , remove  $i$  from  $ND$  and update  $TR_{i,t}$  values;
    - If  $\delta_{i,t} > \Delta Q > 0$  then
      - If  $\delta_{i,t} - \Delta Q > TR_{i,t} / \pi_i$ , let  $w_{it} = \delta_{i,t} - \Delta Q$  and  $\Delta Q = 0$ ;
      - Else let  $\Delta Q = 0$ ,  $w_{it} = 0$ , remove  $i$  from  $ND$ ;
    - End-If;
    - If  $\Delta Q \leq 0$ , let  $w_{kt} = \delta_{k,t}$  for all  $k$  in  $ND$  with unassigned  $w_{kt}$  values and STOP;
- Continue;

In algorithm SUBALG1-R, the set of customers whose estimated transportation cost values are found to be higher than their shortage costs if no delivery is made in period  $t$  are to be excluded. Since this exclusion decision may not be the most appropriate decision especially when deliveries to other group of customers are considered, the algorithm performs this exclusion randomly in step 1. The rest of the algorithm does not contain any randomized decisions. After the exclusions made in step 1, if the available vehicle capacity can cover all customers' demand values, the delivery amount  $w_{it}$  for each customer is set to  $\delta_{i,t}$  and the algorithm proceeds to execute the solution algorithm for subproblem SUB2. Otherwise, delivery amounts are assigned to the customers such that the highest shortage amount is assigned to the customer with the lowest unit shortage cost first.

For the remaining vehicle capacity in period  $t$ , if any, we need to decide whether it is profitable to allocate this capacity to meet future customer demand. Here we only consider meeting future demand for customers that have planned deliveries in period  $t$  as decided in SUBALG1-R (i.e. customers that still belong to set  $ND$ ). Let  $DL$  represent the total remaining vehicle capacity after assigning the delivery amounts in the solution of SUB1, and let  $TL_i$  be the latest period where customer's  $i$  demand can be satisfied, i.e.  $TL_i = \min \left\{ \arg \max_{TL} \left( \sum_{\tau=t+1}^{TL} d_{i\tau} \leq C_i \right), T \right\}$ . The following list describes a randomized version of the algorithm used to solve SUB2. This algorithm is applied after SUBALG1-R.

### Procedure SUBALG2-R

1. For every customer  $i$  in set  $ND$ , Let  $\Delta t_i = 1$ ;
2. For every customer  $i$  in set  $ND$  calculate the value  $F_i = (TR_{i,t+\Delta t_i} - \Delta t_i h_i d_{i,t+\Delta t_i})$ ;
3. For every customer  $i$  in set  $ND$  that has  $F_i > 0$ , calculate the ratio  $P_i = F_i / \sum_{i \in ND} F_i$
4. Based on the calculated  $P_i$  values, use a roulette wheel selector to randomly select a customer and let  $j$  denotes its index;
5. If  $DL \geq d_{j,t+\Delta t_j}$  Then
  - Let  $DL = DL - d_{j,t+\Delta t_j}$  and  $w_{jt} = w_{jt} + d_{j,t+\Delta t_j}$  ;
  - Update  $TR_{i,t+\Delta t_j}$  values in period  $t+\Delta t_j$ ;
  - Let  $\Delta t_j = \Delta t_j + 1$ ;
  - If  $\Delta t_j > TL_j$ , remove customer  $j$  from set  $ND$ ;
  - End-If
  - Else remove customer  $j$  from set  $ND$ ;
6. If  $ND = \emptyset$  then stop; Else go to step 2.

SUBALG2-R decides how much to add to the deliveries made to the customers in period  $t$  to exploit the remaining vehicle capacity and the cost savings that can be achieved by reducing future transportation costs. To simplify the decision involved, the algorithm is based on guiding principles from the uncapacitated lot-sizing problems, which assume inventory replenishment decisions only when the customer's inventory reaches zero. Using these principles, the algorithm generates solutions in which delivery schedules cover customers' exact demand requirements in future days. That is, partial fulfillment of a customer's demand in a future day is not considered. The cost saving that can be achieved by adding a customer's future demand to its current delivery,  $F_i = (TR_{i,t+\Delta t_i} - \Delta t_i h_i d_{i,t+\Delta t_i})$ , is evaluated and used to randomly decide which customer will be selected to add its future demand. This process is repeated until the remaining vehicle capacity is consumed. The random selection process is based on a roulette wheel selector, where the selection probabilities are proportional to the  $F_i$  values.

#### 4.2. Estimating and updating individual customer transportation cost

The individual customer transportation cost values ( $TR_{i,t}$ ) are estimated by calculating the cost reduction that will result when customer  $i$  is excluded from the vehicle tour that includes it, given the VRP solutions. The VRP solutions need to be obtained for the current day  $t$  and for all future days for which customers' demands can be covered in day  $t$  while considering capacity constraints. A suitable heuristic such as the Savings algorithm can be used to generate efficient VRP tours using the appropriate demand values for each day studied. However, if the summation of customers' demands is found to exceed the available vehicle capacity, customers with the lowest unit shortage costs are assigned transportation cost values equal to the transportation costs incurred by a direct shipment from the depot.

The process of estimating transportation costs is repeatedly used as a subroutine to update these estimates during the course of the algorithm whenever a change in the delivery plan occurs.

In SUBALG1-R, the VRP solutions are generated using  $\delta_{i,t}$  values for customers in set  $ND$ . While in SUBALG2-R, the demand values  $d_{i\tau}$  in a future period  $\tau > t$  for all customers whose demand have not been added to the delivery plan in a preceding period are used.

#### 4.3. *Limitations of the construction algorithm*

It is worth noting here that the ATCH and its randomized version have two main limitations. The first limitation is due to the myopic nature of the decisions involved. This myopic nature stems from the strategy used at each period for solving the two subproblems, as it aims to provide optimal inventory allocation decisions from the studied period up to a point in the future allowed by the available capacity. This strategy does not consider the impact of such decisions on the optimality of the overall solution. The second limitation is concerned with the lack of the possibility of making partial deliveries to customers, which may provide savings in transportation and shortage costs. The genetic crossover and mutation operators are designed to overcome these limitations.

### 5. Crossover operation

Crossover operators are an essential part of GAs as they help in inheriting better characteristics from the fittest solutions among generations. Traditionally in combinatorial optimization problems, crossover operators are conducted by breaking the genetic structure (chromosome) and exchanging the broken elements between two different parent solutions. This results in two new child solutions with combined characteristics from both parent solutions.

#### 5.1. *Crossover mechanisms*

For the designed genetic representation of the IIDP, the two-dimensional matrix structure can be broken either vertically or horizontally or by a combination of them. A vertical breakdown means that delivery schedules for a selected set of periods will be exchanged between the two parent solutions. Obviously, the vertical breakdown will maintain the vehicle capacity constraints but may result in customer storage capacity violation. Moreover, solutions generated by the vertical breakdown will generally have poor inventory decisions that appear in the form of extra unnecessary inventory or backorder, which discouraged us from applying this method.

Horizontal breakdown means that delivery schedules for a selected set of customers will be exchanged between two parent solutions. In this case, the delivery schedule and inventory decisions for each customer defined in the parent solutions will remain unchanged in the resultant child solutions. However, vehicle capacity constraints may be violated. Figure 4 shows an example of how this could happen in the case of the sample problem presented earlier in this paper. Here, the second child solution is infeasible due to the vehicle capacity constraint violation occurred in the second period. There are two approaches to deal with this situation. The first one is to assign a very high cost for such candidate solutions and accordingly reduce their probability of being selected in the forthcoming generations. The second approach is to try to fix the resultant capacity violations by adjusting the delivery amounts. The advantage of the second

approach over the first one is that it is more suitable in problems that are more likely to produce vehicle capacity violations and it enables GA to investigate further points in the search space.

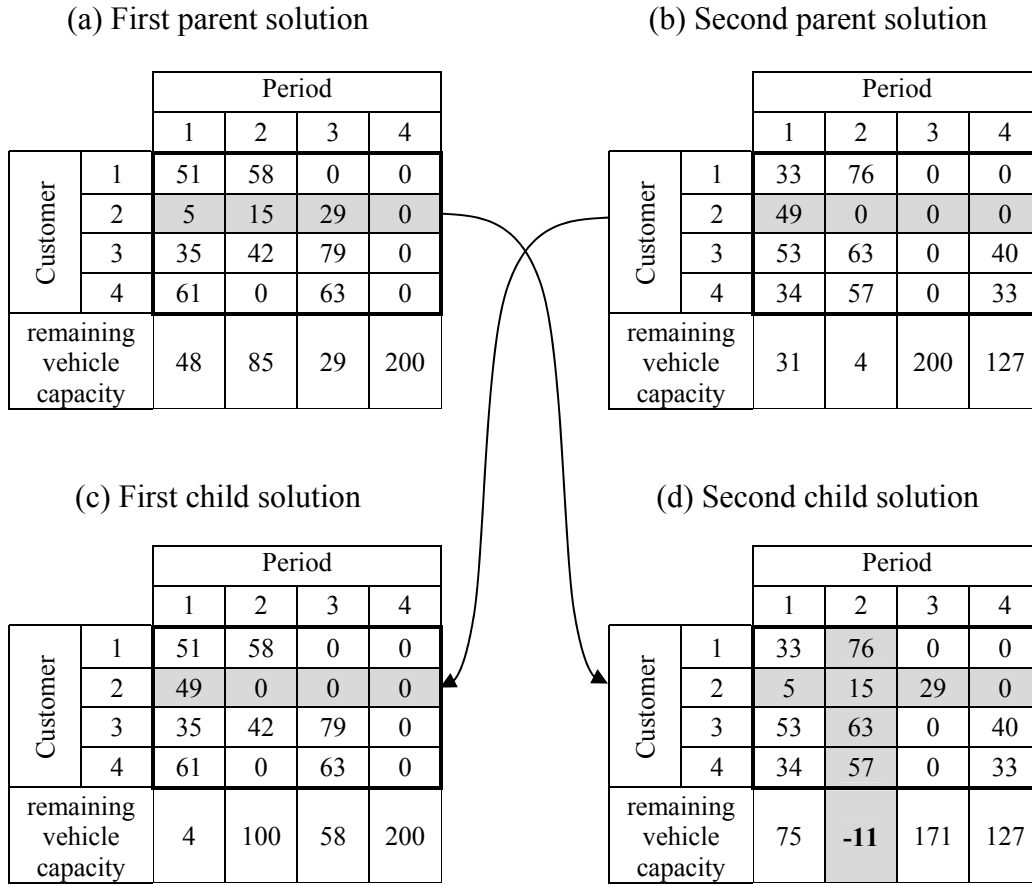


Figure 4. Illustration of horizontal exchange and resultant vehicle capacity violation

## 5.2. Fixing vehicle capacity constraint violation

The problem of fixing the vehicle capacity constraint violation can be stated as follows. Let  $S$  be the set of time periods in which vehicle capacity constraints are violated and  $D$  be the set of remaining periods in addition to a dummy period  $T+1$ . For each period  $t \in S$ , the amount of vehicle capacity constraint violation is denoted  $\sigma_t$  (negative value). For each of the remaining periods  $\tau \in D$ , The unused vehicle capacity is denoted  $\rho_\tau$  and  $\rho_{T+1} = \max\left(-\sum_{t \in S} \sigma_t - \sum_{\tau \in D} \rho_\tau, 0\right)$ .

Given the current delivery schedule  $w_{it}$ , let  $z_{t,\tau}^i$ , be the amount transferred from the deliveries made to customer  $i$  in period  $t \in S$  to the deliveries made to the same customer in period  $\tau \in D$  such that the total amounts transferred from each period  $t \in S$  equals the total amount of vehicle

capacity violation in that period, i.e.  $\sum_{i=1}^N \sum_{\tau \in D} z_{t,\tau}^i = -\sigma_t$ . In addition, the total amount transferred to any period  $\tau \in D$  should not exceed the remaining capacity in that period, i.e.  $\sum_{i=1}^N \sum_{t \in S} z_{t,\tau}^i \leq \rho_\tau$ , and the total amount to be transferred to a customer is restricted by the customer's storage capacity limit, i.e.  $\sum_{t \in S} z_{t,\tau}^i \leq C_i - I_{i\tau}$ . The last constraint that needs to be considered restricts the amount that can be transferred from customer  $i$  in period  $t \in S$  by the current amount stated in the current delivery schedule, i.e.  $\sum_{\tau \in D} z_{t,\tau}^i \leq w_{it}$ . The objective is to minimize the total inventory holding, backorder and transportation costs associated with the required delivery transfers.

The problem of fixing the vehicle capacity constraint violation is not an easily solvable problem as the cost elements are non-linear mainly due to the vehicle routing part. Accordingly, we present here a simple greedy-based search algorithm described in the following list.

#### Procedure VCF

For each time period  $t \in S$  do

While  $\sigma_t < 0$  do

For each customer  $i$  that has  $w_{it} > 0$ , calculate the change in total cost  $\Delta TC_{t,\tau}^i$  that will result from transferring *one unit* from period  $t$  to every period  $\tau \in D$  whenever customer storage and vehicle capacities permit;

Let  $j$  denote the customer that has the minimum  $\Delta TC_{t,\tau}^j$  value;

Let  $\sigma_t = \sigma_t + 1$ ,  $w_{jt} = w_{jt} - 1$ ,  $\rho_\tau = \rho_\tau - 1$  and  $w_{j\tau} = w_{j\tau} + 1$ ;

Continue;

Continue;

The vehicle capacity fixation (VCF) algorithm works by arbitrarily selecting any time period  $t$  from set  $S$ . Then it fixes the vehicle capacity violation in this period by transferring unit by unit from the deliveries scheduled in it to the other periods that belong to set  $D$ . The selection of suitable customers and periods from set  $D$  for these unit transfers is done by comparing the resultant changes in the total cost. The calculation of the change in the total cost,  $\Delta TC_{t,\tau}^i$ , considers all changes in inventory holding and shortage costs, and may also include vehicle routing costs if the vehicle routing generation algorithm is not computationally time consuming. Since we are using the savings algorithm for the determination of the vehicle routes, the computation of the transportation costs can easily be added to the change in the total cost computation,  $\Delta TC_{t,\tau}^i$ . Since the delivery amounts are set to be integers as defined in the genetic representation, the transfer of one unit at a time suffices. The reason for performing the transfer of one unit at a time is to deal with the non-linearity of the associated changes of the total cost.

### 5.3. Designed crossover operator

In our GA implementation, we use the horizontal exchange of rows between two different solutions as the crossover operator. This is done by randomly selecting any number of customers and exchanging their delivery amounts between the two solutions to generate two new child solutions as illustrated in Figure 4. The VCF is then applied whenever needed.

## 6. Mutation operation

Mutation operators are applied to each child solution resulting from the crossover operation. They help the GA to reach further solutions in the search space. The idea of the mutation operation is to randomly mutate a solution's genes (the values assigned to each cell in our two dimensional matrix structure) and hence produce a new solution that is not very far from the original one.

As presented earlier in the initialization algorithm, solution alternatives in which part of a customer's future demand is covered in a given period are not generated during the construction phase. Such partial deliveries may provide savings in transportation and shortage costs and hence overall better solutions. Furthermore, the designed crossover operator is not sufficient to investigate such solution alternatives. Hence, the mutation operator is specially designed to investigate partial deliveries.

### 6.1. Consideration of partial deliveries

To effectively modify any given solution by considering partial deliveries, we first need to look for reductions or additions to its current delivery amounts at which there are apparent savings in transportation or shortage costs. This is done for a given period in the delivery schedule at a time. Then the possible reduction/addition amounts can be transferred to/from another period whenever customer and vehicle capacity limits permit. The process of transferring part of a customer's delivery amount from one period to another is referred to as *delivery exchange*. If the quantity is to be transferred from a period to one of its successors, this delivery exchanged is called *forward delivery exchange*. Similarly if the quantity is to be transferred to a preceding period, this operation is called *backward delivery exchange*. Figure 5 illustrates a sample backward delivery exchange operation conducted over the initial delivery schedule shown in figure 2. In this delivery exchange process, the delivery quantity to customer 2 scheduled in period 3 is reduced by eight units and this amount is transferred to period 1.

The reason for selecting the amount of eight units is to reduce the number of vehicles used in period 3 and hence reduce the transportation cost in this period. Although, inventory holding cost for customer 3 will increase, the gain from reducing the transportation costs is higher than the loss from increasing the inventory costs. The final result is a reduction in the total cost and hence a better solution.

(a) Modified delivery schedule after transferring eight units

		Period			
		1	2	3	4
Customer	1	0	49	0	0
	2	69	0	55	0
	3	64	0	45	0
	4	35	81	0	40

(b) Resultant end of period inventory positions for the modified solution

		Period			
		1	2	3	4
Customer	1	-5	29	17	0
	2	35	8	33	0
	3	26	0	25	0
	4	0	39	0	0

(c) Generated vehicle tours for the modified solution

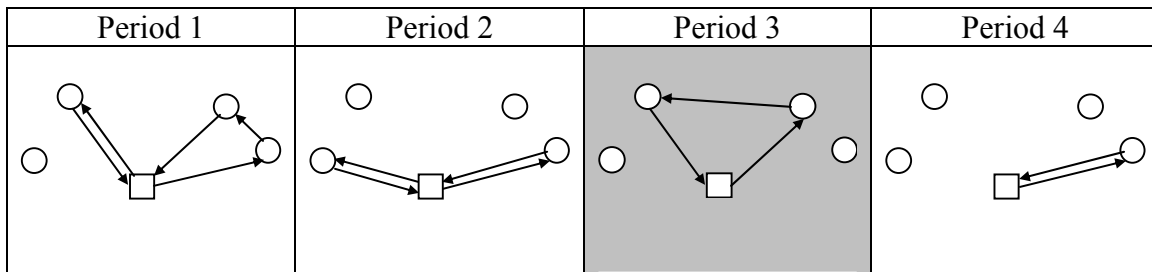


Figure 5. An illustration of a backward delivery exchange on the solution presented in figure 2

## 6.2. Delivery exchanges

It is required to determine at which period should this quantity be transferred to or from, and what would be the most economical way to conduct this transfer. To answer these questions, we develop a set of heuristic guiding rules. These rules are not inclusive and can easily be extended whenever additional rules are seen to be beneficial. The following list describes the developed guiding rules.

- Delivery exchange rules when decreasing a customer's scheduled delivery to reduced transportation costs in a given period  $t$ :
  1. Implement a backward delivery exchange for a customer from period  $t$  to its predecessor period  $t-1$ . If a delivery to the customer is not scheduled in period  $t-1$ , a forward delivery exchange from the last period in which the customer has a delivery to period  $t-1$  with the amount of the customer's demand in  $t-1$  needs to be made (if the available capacity permits) to avoid unnecessary inventory holding cost.
  2. Implement backward delivery exchanges from period  $t$  to previous periods in which the customer has scheduled deliveries. In this rule, the customer's delivery amount in period  $t$  can be divided between the previous delivery periods if the available capacity in a previous period does not allow for exchanging the whole amount.
  3. Implement a forward delivery exchange for a customer from period  $t$  to its successor period  $t+1$  whenever customer and vehicle capacities permit. This rule is intended to be used to reduce inventory holding cost.

4. This rule is a combination of rules 2 and 3, where the whole delivery amount of a customer is to be exchanged with proceeding periods in which the customer has scheduled deliveries and the succeeding period  $t+1$ . The forward exchange with period  $t+1$  will be made if the total available capacity for the proceeding periods does not permit the transfer of the whole amount.
- Delivery exchange rules when increasing the delivery made to a customer to reduce its shortage cost in period  $t$ :
    5. Increase the deliveries made to the customer in previous periods in which the customer has scheduled deliveries such that the customer and vehicle capacities are not violated and the summation of the increased delivery amounts does not exceed the shortage amount of the customer.
    6. This rule is an extension to rule 5, where adding deliveries to previous periods in which the customer does not have scheduled deliveries is allowed.
    7. In the case of increasing the delivery made to a customer in period  $t$  to reduce the existing shortage cost of customer  $i$ , check for reducing the delivery made to another customer that has a scheduled delivery in period  $t$  and has the lowest unit inventory holding cost. This will result in a backward delivery exchanges for that customer, which should follow rules 1 and 2.

### 6.3. *Designed mutation operator*

The mutation operator is designed to conduct delivery exchanges in a random fashion. First a period is selected randomly and all possible sources of delivery reductions or additions at which savings in transportation and inventory shortage costs can be achieved are listed. From that list, only one reduction or addition is randomly selected. For the selected reduction or addition, possible delivery exchanges with preceding and succeeding periods are investigated using the previously mentioned guiding rules. A delivery exchange is then selected randomly and applied to modify the current solution. This process is repeated for the given solution a number of times that is twice the number of customers.

## 7. **Experimentation and results**

### 7.1. *Genetic algorithm implementation*

In our GA implementation, we use a simple GA search structure with elite preservation. The algorithm starts by generating the initial population using the randomized version of the heuristic ATCH. The size of this population remains constant throughout the application of the algorithm. Then the improvement phase of the GA follows by applying the designed crossover and mutation operators for a randomly selected pair of solutions from the current population. We use a simple roulette wheel selector with the selection probability for each solution inversely proportional to its total cost value. Specific probabilities are assigned for both crossover and mutation operators to define the frequency by which they are applied. To move from the current population to a new one, the selection process followed by the crossover and the mutation operations is repeated a number of times equals half the population size. The creation of a new

population is repeated a number of times called the number of generations. In order not to lose the best solutions found throughout the generations due to the randomized selection mechanism, a set of the best solutions found are reserved in what is referred to as the elite set. This elite set has a fixed size and used to feed the starting population of solutions in every generation.

In our experimentation, we used the following parameters. Number of generations: 300, population size: 60, elite size: 10, crossover probability: 0.8, and mutation probability: 0.8. The selection of the crossover and mutation probabilities are based on preliminary experiments, which showed that their range should be greater than or equal to 0.8.

## *7.2. Experimental design*

We use a previously developed testbed by Abdelmaguid (2004) to benchmark the developed GA against the results obtained by two versions of the ATCH. These two versions differ in the way subproblem SUB2 is solved. The first version referred to as the ATCH-G uses a greedy based search mechanism, while the other one called ATCH-DP uses a dynamic programming approach. We also compare the results obtained by the GA with previously generated lower bounds to give a measure of how far the results are from the optimal solutions. The lower bounds are generated using CPLEX 8.1, a commercial MIP solver.

Random test problems are generated to consider a situation in which backorder decisions are economical. That is, we set the parameter values so it is optimal to carry backorders. We assume that customers are allocated in a square of  $20 \times 20$  distance units and their coordinates are generated using a uniform distribution within these limits. The depot is located in the middle of the square. The transportation cost per unit distance is set to 2. Customers' unit holding costs are generated using a normal distribution with a mean of 0.1 and a standard deviation of 0.02, and each customer has a storage capacity of 120 items. The customers' unit shortage costs are generated using a normal distribution with a mean of 3 and a standard deviation of 0.5, and the customers' demands are generated using a uniform distribution from 5 to 50 items per day.

Sixty test problems have been generated by varying the number of customers ( $N$ ), the number of planning periods ( $T$ ) and the number of homogenous vehicles ( $V$ ). We generate three levels of  $N$  (5, 10 and 15), two levels of  $T$  (5 and 7), and two levels of  $V$  (1 and 2). For each combination of  $N$ ,  $T$ , and  $V$ , we randomly generate five problems. The total vehicle capacity is selected to be fixed at 150, 300, and 450 for each level of  $N$ , respectively.

The naming convention of the test problems starts with the letters 'IIDP', followed by two digits for the number of customers. The third digit represents the length of the planning horizon, and the fourth digit represents the number of vehicles. Finally, the replicate number is given at the last digit, and is separated from the former digits by a hyphen. Thus, the problem IIDP0551-1 represents the first randomly generated test problem with 5 customers, a planning horizon of 5 periods and 1 vehicle.

## *7.3. Results and discussion*

Table 2 lists the results obtained for the test problems using the two versions of the ATCH, the lower bounds generated by using a commercial mixed integer programming (MIP) solver, CPLEX 8.1, and the GA results. For each test problem, ten GA runs were conducted and the best, average and standard deviation of these ten runs are listed in table 2.

The percentage differences between the total cost obtained by each heuristic and the lower bound are used as performance indicators. These percentage differences are calculated by taking the ratio of the difference between the heuristic's total cost and the lower bound to the lower bound. A comparison against the CPLEX lower bound provides some measure of deviation from optimality. In figure 6, the calculated percentage differences are plotted against the number of binary variables in the MIP formulation of the IIDP given in Abdelmaguid (2004). The number of binary variables are taken here as a measure of the problem complexity.

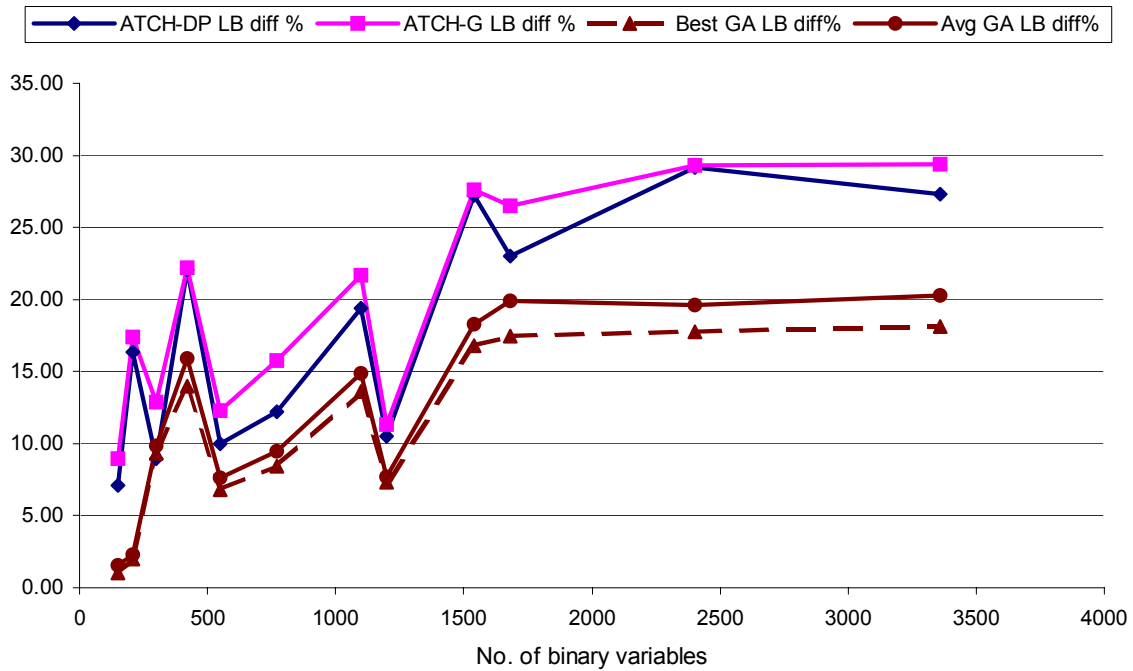


Figure 6. Percentage differences against lower bounds vs. number of binary variables

As illustrated in figure 6, on average the GA outperforms both versions of the constructive heuristic. And it is capable of generating solutions that are very close to the optimal for small problems, and stay within 20% from the optimal for larger ones.

## 8. Conclusion

In this paper, a genetic algorithm approach for solving the integrated inventory distribution problem is developed. A suitable genetic representation that focuses on the delivery schedule represented in the form of a 2-dimensional matrix is designed. In the GA construction phase, a randomized version of a previously developed construction heuristic is used. In the GA improvement phase which involves two random neighbourhood search mechanisms, the crossover and mutation operations, suitable designs are developed. The main concern in designing the mutation operator was to develop a suitable mechanism that allows for deliveries to customers that cover part of their demand requirements, which is referred to as partial deliveries. Partial deliveries can provide savings in transportation and shortage costs and hence provide better solutions.

The experimental results show the significance of the developed GA approach. On average, GA outperforms the previously developed construction algorithm and generates solutions that are within 20% from the optimal solution.

Problem	CPLEX LB	ATCH-DP	ATCH-G	GA		
				Best	Average	S.D.
IIDP0551-1	649.8*	687.83	702.09	668.93	668.93	0.00
IIDP0551-2	468*	537.27	537.27	468.16	480.63	8.50
IIDP0551-3	400*	406.85	430.07	400.61	400.61	0.00
IIDP0551-4	475.29*	475.95	480.52	475.95	475.95	0.00
IIDP0551-5	426.01*	481.87	483.46	433.82	434.22	1.26
IIDP0571-1	522.97*	640.65	634.37	539.49	539.49	0.00
IIDP0571-2	557.89*	580.81	610.47	557.89	557.89	0.00
IIDP0571-3	434.86*	510.92	510.92	435.94	437.59	1.95
IIDP0571-4	536.42*	647.07	646.07	567.36	570.19	4.55
IIDP0571-5	498.08*	582.22	588.52	501.28	505.82	5.08
IIDP0552-1	509	550.13	563.53	540.52	540.78	0.34
IIDP0552-2	933.76	991.78	991.78	1059.78	1059.78	0.00
IIDP0552-3	497.98	578.56	638.11	540.75	545.08	5.82
IIDP0552-4	519.91	555.34	581.41	547.74	547.74	0.00
IIDP0552-5	536.52	576.96	576.96	605.71	615.66	5.26
IIDP0572-1	789.04	980.06	980.06	849.08	854.07	3.20
IIDP0572-2	943.43	1042.9	1042.86	1081.20	1091.06	4.16
IIDP0572-3	793.38	960.04	960.04	949.17	973.76	15.04
IIDP0572-4	738.55	936.11	937.51	843.10	872.66	17.72
IIDP0572-5	728.76	930.36	935.16	830.02	837.36	7.89
IIDP1051-1	509.59	572.84	572.84	531.78	537.39	3.25
IIDP1051-2	423.78	495.26	515.41	488.23	491.67	4.72
IIDP1051-3	660.23	698.7	698.7	698.70	698.70	0.00
IIDP1051-4	445.86	463.22	474.7	456.00	457.12	0.63
IIDP1051-5	546.62	607.43	629.08	581.37	592.23	6.38
IIDP1071-1	728.48	814.52	814.08	779.22	782.28	2.48
IIDP1071-2	730.1	831.05	828.35	813.93	816.91	2.02
IIDP1071-3	668.8	769.15	807.83	722.35	729.68	4.66
IIDP1071-4	799.72	888.57	936.66	880.54	891.03	7.63
IIDP1071-5	712.32	779.75	825.19	753.45	766.03	5.84
IIDP1052-1	758.39	831.04	831.04	838.31	843.47	2.73
IIDP1052-2	566.94	697.34	746.66	648.14	659.29	9.69
IIDP1052-3	659.22	769.63	774.93	729.67	734.16	4.14
IIDP1052-4	509.46	695.17	693.8	626.30	641.59	6.05
IIDP1052-5	718.07	797.42	814.33	786.13	786.13	0.00
IIDP1072-1	808.46	1013.1	1003.49	941.84	949.07	6.57
IIDP1072-2	1029.21	1275.3	1275.34	1212.60	1223.74	7.89
IIDP1072-3	857.06	1118.8	1129.75	987.62	1002.86	11.40
IIDP1072-4	896.78	1173	1177.59	1068.11	1077.89	7.89
IIDP1072-5	750.97	944.37	951.84	867.47	885.53	13.30
IIDP1551-1	736.42	806.16	823.12	782.69	784.80	2.47
IIDP1551-2	725.42	785.8	775.67	758.69	762.09	4.35
IIDP1551-3	666.94	740.41	741.97	731.82	731.82	0.00
IIDP1551-4	608.49	714.51	729.72	689.24	694.93	4.32
IIDP1551-5	971.66	1033.3	1038.44	999.66	999.66	0.00
IIDP1571-1	747.3	942.16	948.35	885.62	909.00	11.22
IIDP1571-2	660.8	837.18	888.73	789.76	813.00	13.84
IIDP1571-3	800.45	1023.9	1037.83	955.38	976.24	9.74
IIDP1571-4	803.99	1004	1053.25	972.59	989.76	9.77
IIDP1571-5	1130.8	1237.4	1247.24	1230.17	1242.41	6.73
IIDP1552-1	620.96	848.64	832.27	748.52	764.72	8.99
IIDP1552-2	595.9	746.02	757.33	695.22	698.79	2.63
IIDP1552-3	679.235	965.73	968.45	840.42	859.01	13.42
IIDP1552-4	923.82	1052	1052.04	1019.53	1031.76	6.04
IIDP1552-5	729.65	932.91	940.11	858.59	872.35	8.76
IIDP1572-1	881.63	1189.7	1195.65	1069.78	1093.85	12.91
IIDP1572-2	972.09	1189.1	1201.01	1137.76	1149.25	6.24
IIDP1572-3	1042.43	1309.4	1356.36	1244.33	1270.60	16.49
IIDP1572-4	920.04	1233.9	1263.63	1102.91	1128.25	16.76
IIDP1572-5	1117.42	1335.8	1344.32	1261.34	1281.69	12.05

\* Optimal solution found

Table 2. Experimental results

## Appendix A

The transportation costs function can be expressed as follows:

$$TR_t(W_t) = \min \sum_{j=1}^N \sum_{v=1}^V f_t x_{0jt}^v + \sum_{i=0}^N \sum_{\substack{j=0 \\ j \neq i}}^N \sum_{v=1}^V c_{ij} x_{ijt}^v \quad (6)$$

Subject to:

$$\sum_{\substack{j=0 \\ j \neq i}}^N x_{ijt}^v \leq 1 \quad i = 0, \dots, N \text{ and } v = 1, \dots, V \quad (7)$$

$$\sum_{\substack{k=0 \\ k \neq i}}^N x_{ikt}^v - \sum_{\substack{l=0 \\ l \neq i}}^N x_{lit}^v = 0 \quad i = 0, \dots, N \text{ and } v = 1, \dots, V \quad (8)$$

$$y_{ijt}^v - q_v x_{ijt}^v \leq 0 \quad i, j = 0, \dots, N, i \neq j \text{ and } v = 1, \dots, V \quad (9)$$

$$\sum_{\substack{k=0 \\ k \neq i}}^N y_{ikt}^v - \sum_{\substack{l=0 \\ l \neq i}}^N y_{lit}^v \leq 0 \quad i = 1, \dots, N, \text{ and } v = 1, \dots, V \quad (10)$$

$$\sum_{v=1}^V \left( \sum_{\substack{l=0 \\ l \neq i}}^N y_{lit}^v - \sum_{\substack{k=0 \\ k \neq i}}^N y_{ikt}^v \right) = w_{it} \quad i = 1, \dots, N \quad (11)$$

$$x_{ijt}^v \geq 0 \text{ and } x_{ijt}^v = 0 \text{ or } 1 \quad i, j = 0, \dots, N, i \neq j \text{ and } v = 1, \dots, V \quad (12)$$

This transportation cost function definition represents an MIP model for a capacitated vehicle routing problem (VRP) that need to be solved for any given delivery amounts  $W_t = (w_{it} : i = 1, \dots, N)$ . In this model, deliveries to customers  $1, \dots, N$  are to be made by a capacitated heterogeneous fleet of  $V$  vehicles, each with capacity  $q_v$  starting from the depot at the beginning of each period. In the IIDP-NLP model  $Q = \sum_{v=1}^V q_v$ . The binary decision variables  $x_{ijt}^v$  are used to represent the decision if vehicle  $v$  travels from location  $i$  to location  $j$  in period  $t$  or not. The continuous variables  $y_{ijt}^v$  represent the corresponding material flow, which is restricted by the available vehicle capacity  $q_v$ . In the objective function (6), transportation costs include  $f_t$  a fixed usage cost per vehicle, which depends on the period  $t$ , and  $c_{ij}$  a variable transportation cost between  $i$  and  $j$ , which satisfies the triangular inequality. Constraints (7) make sure that a vehicle will visit a location no more than once in a time period, and constraints (8) ensure route continuity. Constraints (9) serve for two purposes. The first one is to ensure that the amount transported between two locations will always be zero whenever there is no vehicle moving between these locations, and the second is to ensure that the amount transported is less than or equal to the vehicle's capacity. Constraints (10) are necessary to eliminate sub-tours. The relationship between the flow variables and the delivery amount decision variables are defined in constraints (11). Constraints (12) are the domain constraints.

## References

- Abdelmaguid, T. F., 2004, Heuristic approaches for the integrated inventory distribution problem, PhD Dissertation, University of Southern California, Los Angeles.
- Baita, F., Ukovich, W., Pesenti, R., and Favaretto, D., 1998, Dynamic routing-and-inventory problems: a review. *Transportation Research*, **32**, 585-598.
- Bell, W. J., Dalberto, L. M., Fisher, M. L., and Greenfield, A. J., 1983, Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, **13**, 4-23.
- Blumenfeld, D. E., Burns, L. D., Diltz, J. D., and Daganzo, C. F., 1985, Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Research*, **19**, 361-380.
- Campbell, A., Clarke, L., and Savelsbergh, M., 2002, Inventory routing in practice. *The Vehicle Routing Problem*, SIAM, Philadelphia, 309-330.
- Chien, T. W., Balakrishnan, A., and Wong, R. T., 1989, An integrated inventory allocation and vehicle routing problem. *Transportation Science*, **23**, 67-76.
- Clarke, G., Wright, J., 1964, Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**, 568-581.
- Daganzo, C. F., 1987, The break-bulk role of terminals in many-to-many logistic networks. *Operations Research*, **35**, 543-555.
- Daganzo, C. F., 1999, *Logistics Systems Analysis*, (Berlin: Springer).
- Dror, M., Ball, M., and Golden, B., 1985, A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, **4**, 2-23.
- Dror, M., and Ball, M., 1987, Inventory/routing: reduction from an annual to a short-period problem. *Naval Research Logistics*, **34**, 891-905.
- Ernst, R., and Pyke, D. F., 1993, Optimal base stock policies and truck capacity in a two-echelon system. *Naval Research Logistics*, **40**, 879-903.
- Federgruen, A., and Zipkin, P., 1984, Combined vehicle routing and inventory allocation problem. *Operations Research*, **32**, 1019-1037.
- Gen, M., and Cheng, R., 1997, *Genetic Algorithms and Engineering Design* (New York: Wiley).
- Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning* (New York: Addison-Wesley).
- Golden, B., Assad, A., and Dahl, R., 1984, Analysis of a large scale vehicle routing problem with an inventory component. *Large Scale Systems*, **7**, 181-190.
- Hall, R. W., 1985, Determining vehicle dispatch frequency when shipping frequency differs among suppliers. *Transportation Research Part B*, **19**, 421-431.
- Herer, Y. T., and Levy, R., 1997, The metered inventory routing problem, an integrative heuristic algorithm. *International Journal of Production Economics*, **51**, 69-81.
- Trudeau, P., and Dror, M., 1992, Stochastic inventory routing: route design with stockouts and route failures. *Transportation Science*, **26**, 171-184.
- Viswanathan, S., and Mathur, K., 1997, Integrating routing and inventory decisions in one-warehouse multicustomer multiproduct distribution systems. *Management Science*, **43**, 294-312.