

## ANT COLONY OPTIMIZATION FOR SPLIT DELIVERY INVENTORY ROUTING PROBLEM

Lily Wong<sup>1</sup>, Noor Hasnah Moin<sup>2</sup>

<sup>1,2</sup>Institute of Mathematical Sciences, University of Malaya, 50603 Kuala Lumpur, Malaysia.

Email: <sup>1</sup>lilywong84@gmail.com, <sup>2</sup>noor\_hasnah@um.edu.my (corresponding author)

### ABSTRACT

*A one-to-many inventory routing problem (IRP) network comprising of a warehouse and geographically dispersed customers is studied in this paper. A fleet of a homogeneous vehicle located at the warehouse transports multi products from the warehouse to meet customer's demand on time in a finite planning horizon. We allow the customers to be visited more than once in a given period (split delivery) and the demand for each product is deterministic and time varying. Backordering is not allowed. The problem is formulated as a mixed integer programming problem and is solved using CPLEX 12.4 to get the lower and upper bound (the best integer solution) for each problem considered. We propose a modified ant colony optimization (ACO) which takes into account not only the distance but also the inventory that is vital in the IRP. We also carried the sensitivity analysis on important parameters that influence decision policy in ACO in order to choose the appropriate parameter settings. The computational results show that ACO performs better on large instances compared to the upper bound and performs equally well for small and medium instances. The modified ACO requires relatively short computational time.*

**Keywords:** *Ant Colony Optimization, Inventory, Routing*

### 1.0 INTRODUCTION

Supply chain management is the control of supply chain to manage the flow of commodity both within and among the companies. Inventory routing problem (IRP) is a challenging NP hard problem in supply chain management which combines the vehicle routing problem where the route to visit the customers is decided and inventory management which concerns on the amount to be delivered to the customers. The main objective of IRP is to minimize both the total transportation and inventory cost over the planning horizon.

Generally, IRP can be divided by different criteria such as the type of demand, single or multi period, planning horizon, and inventory policy (order-up-to level or maximum level). From the aspect of demand, IRP can be divided into deterministic IRP (DIRP) (see Coelho and Laporte [1], Moin et al [2], Mjirda et al [3], Yu et al [4]), where the demand of customers is known in advance and stochastic IRP (SIRP), in which the demand is unknown and based on some probability functions (see example Adelman[5], Kleywegt et al. [6], Kleywegt et al. [7] and Yu et al. [8]).

Coelho and Laporte [1] recently proposed a branch-and-cut algorithm to solve the multi-product multi-vehicle IRP with deterministic demand and stock out cost is not allowed. In their paper, Coelho and Laporte [1] implemented a solution improvement algorithm after branch-and-cut identifies a new best solution. The purpose of solution improvement algorithm is to approximate the cost of a new solution resulting from the vertex removal and reinsertions. In this paper, the authors also considered additional two features: the driver partial consistency and the visiting space consistency. The results show that the visiting space helps in reducing the search space while providing a meaningful solution. This is the first paper that solves the problem with heterogeneous vehicles.

The planning horizon of the problem also can be categorized into finite or infinite horizon. We can observe that most of the earlier works concentrate on an infinite planning horizon (see for example Aghezzaf et al. [9], Anily and Bramel [10] and Campbell and Savelsbergh [11]).

IRP with finite planning horizon can be categorized as a single or multi-period scenario. Federgruen and Zipkin [12] were among the first to study the IRP with finite planning horizon. The problem was treated as a single day problem with a limited amount of inventory and the customers' demands are assumed to be a random variable. The problem decomposes into a nonlinear inventory allocation problem that determines the inventory and short-

age costs and a Travelling Salesman Problem (TSP) for each vehicle considered, which produces the transportation costs. Chien *et al.* [13] amongst the first to simulate a multiple period planning models based on a single period approach. This is achieved by passing some information from one period to the next through an inter-period inventory flow. Since then, many researchers have focused their modeling on a finite planning horizon.

Dror and Trudeau [14] first introduced the split delivery VRP (SDVRP) by relaxing a constraint of the VRP that every customer is served by only one vehicle. The authors showed that the relaxation increased the flexibility of distribution and could lead to important savings, both in the total distance traveled and in the number of vehicles used. The SDVRP remains NP hard despite this relaxation (Dror and Trudeau [15]). Several authors (see for example Moin *et al.* [2], Mjirda *et al.* [3] and Yu *et al.* [4]) have extended the concept of split delivery in the multi-period IRP.

Moin *et al.* [2] had proposed an efficient hybrid genetic algorithm to solve the IRP in a many-to-one network that comprises of a depot, an assembly plant, and retailers where the products are collected from the retailers to fulfill the demand at the assembly plant. It is assumed that each supplier supplies different products and in multi-period scenario. Split pickups are allowed in this study. The proposed hybrid genetic algorithm is based on the allocation-first-route-second strategy and takes both the inventory and the transportation costs (fixed and variable) into consideration. The computational experiments have been performed on the data sets to show the effectiveness of the proposed approach. Small, medium, and large size problems are added to the existing data sets taken from Lee *et al.* [16]. With the increase of problem size, GA based algorithms performed relatively much better compared to the best integer solutions from CPLEX.

Mjirda *et al.* [3] improved the results obtained by Moin *et al.* [2] using a two-phase Variable Neighborhood Search (VNS). The first phase developed the initial solution without considering the inventory and in the second phase, the initial solution is improved using Variable Neighborhood Descent (VND) or a VNS algorithm. Linear Programming (LP), and a heuristic method are developed for inventory management to determine the quantity of product to collect at each period during the planning horizon. The priority of rules on suppliers and vehicles are applied for this purpose. Both algorithms are implemented using the proposed seven neighborhood structures based on three elementary moves: Drop, add and change. The computational results show that the proposed method gives better results than the existing methods from the literature for both solution quality and the running time.

Yu *et al.* [4] developed an approximate approach that incorporates Lagrangian relaxation to solve a large scale IRP that delivers a single product with split delivery and vehicle fleet size constraint. This approach can quickly and near-optimally solve the problem. In the approach, the relaxed problem of the model is decomposed into inventory problem and a vehicle routing problem that are solved by a linear programming algorithm and a minimum cost flow algorithm respectively, and the dual problem is solved by using the surrogate subgradient method. The solution of the model obtained by the Lagrangian relaxation method is used to construct a near-optimal solution of the IRP by solving a series of assignment problems. Numerical experiments show that the proposed hybrid approach can find a high quality near-optimal solution for the IRP with up to 200 customers and 10 periods in a reasonable computation time.

Metaheuristic algorithms such as genetic algorithm (Sin *et al.* [17]), scatter search (Huacuja *et al.* [18]) and variable neighborhood search (Rasheed *et al.* [19]) have been applied to different type of combinatorial problems and one of the well-known metaheuristics, which is the Ant Colony Optimization (ACO). Recently several researchers have applied ACO algorithm metaheuristic in several variants of IRP. Huang and Lin [20] is the first paper that proposed a modification ACO for solving the multi-item inventory routing problems. The demand is assumed stochastic the algorithm chooses a delivery policy that minimizes the total costs. The algorithm was developed for the replenishment of the vending machine and this modified ACO algorithm that incorporates the stockout cost in the calculation of the pheromone values which is not included in the conventional ACO. The nodes with high stockout costs are given higher priority even though the total transportation costs are higher than other nodes. The test instances were constructed using the Solomon's [21] 56 benchmark problems created for vehicle routing problem with time windows. The results show that the modified ACO algorithm achieves highly significant improvements when compared with the conventional ACO.

Calvete *et al.* [22] are the first who studied a bilevel model in the context of hierarchical production-distribution (PD) planning, which is a variant of IRP. In this problem, a distribution company, which is the leader of the hierarchical process, controls the allocation of retailers to each depot and the routes which serve them. The

manufacturing company, which is the follower of the hierarchical process will decide which manufacturing plants will produce the orders received by the depot. In this study, ACO algorithm is developed to solve a multi-depot vehicle routing problem (MDVRP) by determining which depot is to serve which retailer and to build routes which establish the order in which retailer are served. The computational experiment is carried out to analyze the performance of the algorithm. Since the bilevel model is proposed for the first time, there is no data for comparison purposes. The computational time is reasonable, taking into account the problem sizes.

In the most recent work, Tatsis et al [23] developed a mixed integer mathematical model where a fleet of capacitated homogeneous vehicle is used to deliver distinct products from multi suppliers to a retailer to meet the demand in each period over the planning horizon. However, backlogging is allowed in this study. The ant based optimization algorithm is applied to solve the corresponding vehicle routing problem. The objective of this study is to find the best compromise between the transportation, inventory and backlogging cost. Preliminary results show that the solution gaps between the algorithm and CPLEX solutions are kept reasonably low values and offered prospective for further improvement.

Wong and Moin [24] have extended formulation proposed by Yu et al. [4] to incorporate multi products. The authors considered a network consisting of a warehouse that supplies multi-products to geographically dispersed customers, and the products are transported by a fleet of homogeneous vehicles. The customer's demand must be met on time and split delivery is allowed. Ant colony optimization (ACO), which was inspired by the food-foraging behavior of ant was initially proposed by Dorigo and co-workers (Dorigo [25], Dorigo and Blum [26], and Dorigo and Caro [27]) is modified by the authors to solve their proposed model. They have modified the algorithm by incorporating the inventory component to the global updating scheme that not only calculates the pheromone along the trail but identifies a set of feasible neighbors making use of the attractions on the nodes which differs from the classical ACO. The computational results show that ACO performs better on large instances when compared to the upper bounds, which are obtained by solving mixed integer programming formulation using CPLEX version 12.4. In addition, the proposed algorithm performs equally well for small and medium instances. This paper extends the work of Wong and Moin [24].

This paper is organized as follows. Section 2 discusses problem formulation including the assumptions that are made in this model. The solution procedure that is based on ACO is described in detail in Section 3 and it is followed by the computational results and discussion presented in Section 4. Finally, the conclusion is drawn in Section 5.

## 2.0 MODEL FORMULATION

We consider a one-to-many network where a fleet of homogeneous vehicle transports multi-products from a warehouse or depot to a set of geographically dispersed customers in a finite planning horizon. The following assumptions are made in this model.

- The fleet of homogenous vehicles with limited capacity is available at the warehouse. The number of vehicles is assumed to be unlimited.
- Customers can be served by more than one vehicle (split delivery is allowed).
- Each customer requests a distinct product and the demand for the product is known in advance but may vary between different periods.
- The holding cost per unit item per unit time is incurred at the customer sites but not incurred at the warehouse. The holding cost does not vary throughout the planning horizon.
- The demand must be met on time and backordering or backlogging is not allowed.

The problem is modeled as mixed integer programming and the following notation is used in the model:

### Indices

$\tau = \{1, 2, \dots, T\}$	period index
$W = \{0\}$	warehouse/depot
$S = \{1, 2, \dots, N\}$	a set of customers where customer $i$ demands product $i$ only

**Parameters**

$C$	vehicles capacity (assume to be equal for all the vehicles).
$F$	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)
$c_{ij}$	travel distance between customer $i$ 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 customer for product $i$ per unit product per unit time
$d_{it}$	demand of customer $i$ in period $t$

**Variables**

$a_{it}$	delivery quantity to customer $i$ in period $t$
$I_{it}$	inventory level of product $i$ at the end of period $t$
$q_{ijt}$	quantity transported through the directed arc( $i, j$ ) in period $t$
$x_{ijt}$	number of times that the directed arc( $i, j$ ) is visited by vehicles in period $t$

The model for our inventory routing problem is given as below:

$$Z = \min \underbrace{\sum_{t \in \tau} \sum_{i \in S} h_i I_{it}}_{\text{I}} + V \underbrace{\left( \sum_{t \in \tau} \sum_{j \in S} \sum_{i \in S \cup W} c_{ij} x_{ijt} + \sum_{t \in \tau} \sum_{i \in S} c_{i0} x_{i0t} \right)}_{\text{II}} + F \underbrace{\sum_{t \in \tau} \sum_{i \in S} x_{0it}}_{\text{III}} \quad (1)$$

subject to

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

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

$$\sum_{i \in S} q_{0it} = \sum_{i \in S} a_{it}, \quad \forall t \in \tau \quad (4)$$

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

$$I_{it} \geq 0, \quad \forall i \in S, \forall t \in \tau \quad (6)$$

$$a_{it} \geq 0, \quad \forall i \in S, \forall t \in \tau \quad (7)$$

$$q_{ijt} \geq 0, \quad \forall i \in S \cup W, \forall j \in S, j \neq i, \forall t \in \tau \quad (8)$$

$$q_{ijt} \leq C x_{ijt}, \quad \forall i \in S \cup W, \forall j \in S, i \neq j, \forall t \in \tau \quad (9)$$

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

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

The objective function (1) includes the inventory costs (I), the transportation costs (II) and vehicle fixed cost (III). (2), is the inventory balance equation for each product at the warehouse while (3) is the product flow conservation equations, to ensure that the flow balance at each customer and eliminating all sub-tours. (4) assures the collection of accumulative delivery quantity at the warehouse (split delivery). (5) ensures that the number of vehicles leaving the warehouse is equal to the number of vehicles returning to warehouse. (6) assures that the demand at the warehouse is completely fulfilled without backorder. Meanwhile, (9) guarantees that the vehicle capacity is respected and gives the logical relationship between  $q_{ijt}$  and  $x_{ijt}$  which allows for split delivery. This formulation is used to determine the lower and upper bounds for each data set using CPLEX 12.4.

### 3.0 MODIFIED ACO

Ant Colony Optimization (ACO) is inspired by the nature behavior of ants finding the shortest path between their colony and food source. The information collected by ants during the searching process is stored in pheromone trails. Hence, when an ant has built a solution, the ant deposits a certain amount of pheromone proportionally (the information about the goodness of the solution) on the pheromone trails of the connection it used. The pheromone information directs search by other ants while exploring the graph. The higher density of pheromones on an arc attract more ants to the arc. Therefore, an appropriate formulation associated with the model for updating pheromones trail is very crucial. This is due to the reason that the greater amounts of pheromone it deposits on the arcs, the shorter is the path (the minimum cost).

The procedure for ACO can be divided into three main steps: the route construction, a local pheromone-update rule, and a global pheromone-update rule. These steps are described in detail in the following subsections and Fig. 1 outlines the algorithm.

#### 3.1 Initial Solution

We construct the initial solution by having all the demand met in every period. Any vehicle routing algorithms can be used to construct the route. In this study, we adopt a simple Nearest Neighbor algorithm (NN) and the algorithm is modified to allow for split delivery. The vehicle starts at the depot and repeatedly visits the nearest customer (in terms of distance) until the capacity of the vehicle is fully occupied. Then, a new vehicle is initiated and the process continues until all customers have been assigned or visited. The total distance obtained by NN is embedded to initialize the  $\tau_0$ , the initial pheromone in the local pheromone updating (discussed in Section 3.3). A simple NN is adopted since the initial solution is used to approximate  $\tau_0$  and any other methods such as savings and sweep algorithms may be adopted as well.

#### 3.2 Route Construction for ACO

The route construction begins by setting the value of all the parameters  $\alpha, \beta, \tau_0, q_0$  and  $\rho$ .  $\alpha, \beta$  are two parameters that control the influence of the pheromone value allocated on the arc  $(i, j)$  and the desirability of arc  $(i, j)$  respectively whilst  $q_0$  is a predefined real number where  $0 \leq q_0 \leq 1$  and  $\rho$  is the rate of pheromone evaporations. Note the value of  $\tau_0$ , the initial value of pheromones for each arc is obtained from the total distance of the initial solution. Starting from the depot (warehouse), each ant utilizes equation (12) to select the next customer to be visited. Ants tend to be attracted to the arc that consists of higher density of pheromones. From equation (12), if  $q$  is less than the predefined parameter  $q_0$ , then the next arc chosen is the arc with the highest attraction. Otherwise, the next arc is chosen using the biased Roulette Method with the state transition probability  $p_{ij}$  given by equation (14).

$$j = \begin{cases} \max_{j \in \Omega_i} \{Att_{i,j}\} & \text{if } q \leq q_0 \\ p_{ij} & \text{Otherwise} \end{cases} \quad (12)$$

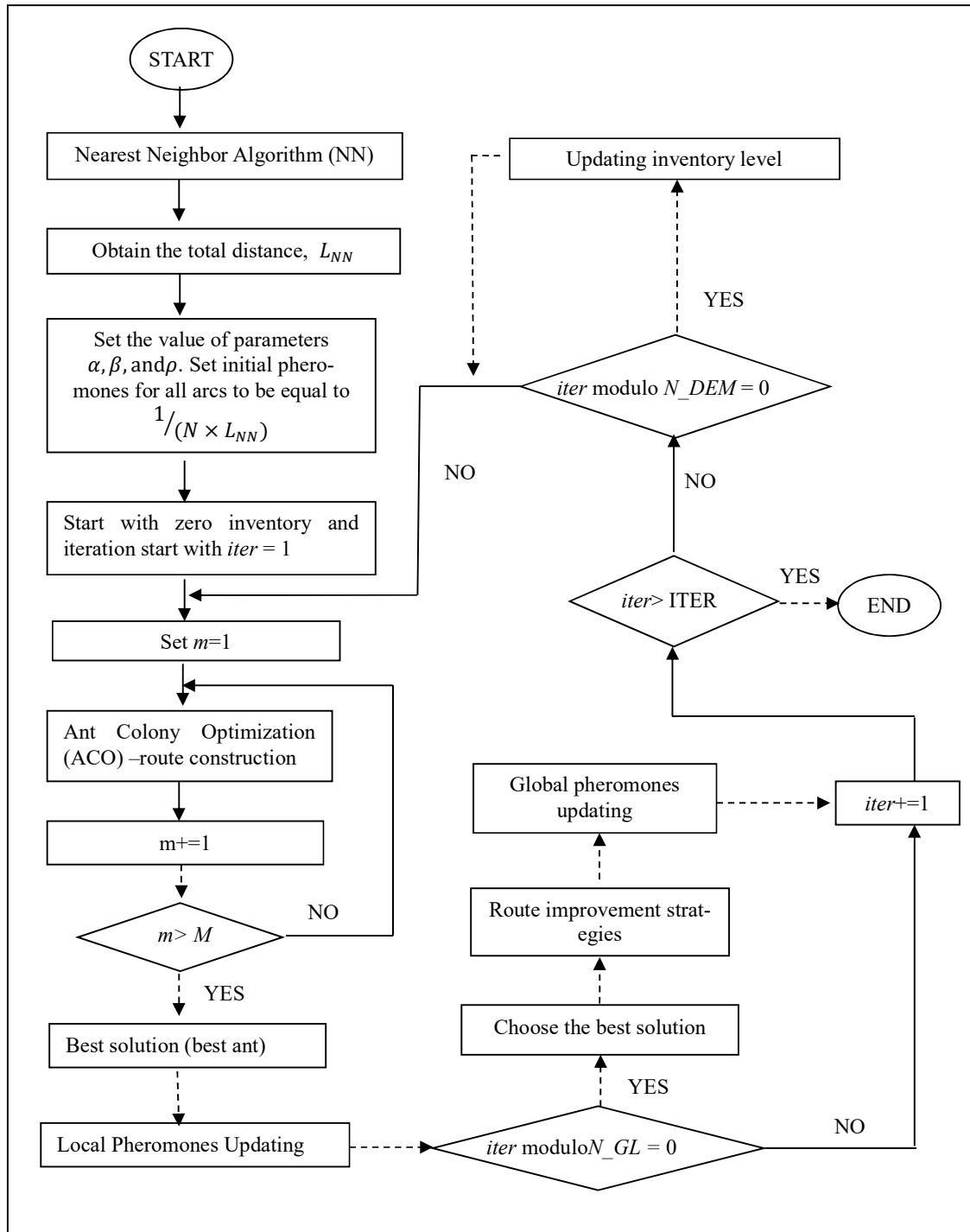


Fig. 1 Algorithm of the developed method

where  $Att_{ij} = (\tau_{ij})^\alpha (\eta_{ij})^\beta$  (13)

$$p_{ij} = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{k \in \Omega_i} (\tau_{ik})^\alpha (\eta_{ik})^\beta} & \forall j \in \Omega_i \\ 0 & \forall j \notin \Omega_i \end{cases} \quad (14)$$

$\tau_{ij}$  is the amount of pheromone deposited on arc  $(i,j)$  and  $\eta_{ij}$  is inversely proportional to the length of arc  $(i,j)$ ,  $c_{ij}$ .  $\Omega_i$  a set of unvisited customers for ant  $i$ .

### 3.3 The Local Pheromone-Updating Rule

Local updating is used to reduce the amount of pheromone on all the visited arcs to simulate the natural evaporation of pheromone and it is intended to avoid a very strong arc being chosen by all the ants. After a predefined number of ants,  $m$  had completed their solutions, the best among the built solutions is chosen and the pheromone on each arc is updated using equation (15).

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (15)$$

where  $\rho$  represents the rate of pheromone evaporation.

### 3.4 The Global Pheromone-Updating Rule

After a predefined number of iterations, the ACO updates the pheromone allocation on the arcs of the current optimum route  $\gamma^{gl}$ . The global pheromone-updating rule resets the ant colony's situation to a better starting point and encourages the use of shorter routes. Moreover, it increases the probability that future routes use the arcs contained in the best solutions. In the classical ACO, only the transportation cost is taken into account in the global updating. Since the IRP tries to find a balanced between the transportation and inventory cost, it is natural to incorporate the inventory cost in the formulation. The global update rule is enhanced as follows:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \frac{\rho}{J_{\gamma^{gl}}}(i, j) \in \gamma^{gl} \quad (16)$$

where  $J_{\gamma^{gl}}$  is the weight of the best solution found where it incorporates the inventory element as well as the variable transportation costs.  $J_{\gamma^{gl}}$  is given by

$$J_{\gamma^{gl}} = \sum_{t \in \tau} \sum_{i \in S} h_i I_{it} + V \left( \sum_{t \in \tau} \sum_{j \in S \cup W} c_{ij} x_{ijt} + \sum_{t \in \tau} \sum_{i \in S} c_{i0} x_{i0t} \right) \quad (17)$$

where the first component defines the total inventory costs whilst the second component gives the total transportation cost.

### 3.5 Route Improvement Strategies

The routes can be further improved by adding route improvement strategies in the route construction procedure. In this study, there are three local searches: swap, 2 - opt\* and 2 - opt are applied to improve the solution built by ACO.

#### 3.5.1 Swap for Split Customer

The first local search is the swap algorithm focusing on the split customer and is illustrated in Fig. 2(a) and 2(b).

The swap comprises of a transfer to the selected vehicle or a swap between different vehicles. Starting from the last vehicle, the split customer is identified and we try to merge to the current selected vehicle if the respective vehicle capacity is not violated. If this fails, then the swap with the other customers either in the preceding or the current vehicle that resulted in the least transportation cost is carried out. If none of the swap provides an improvement in the objective value, then the current solution route remains unchanged. The process continues until all vehicles in every period have been examined. The aim of this method is to eliminate the split customers (merge as many as possible) if the merger results in an improved in the objective value.

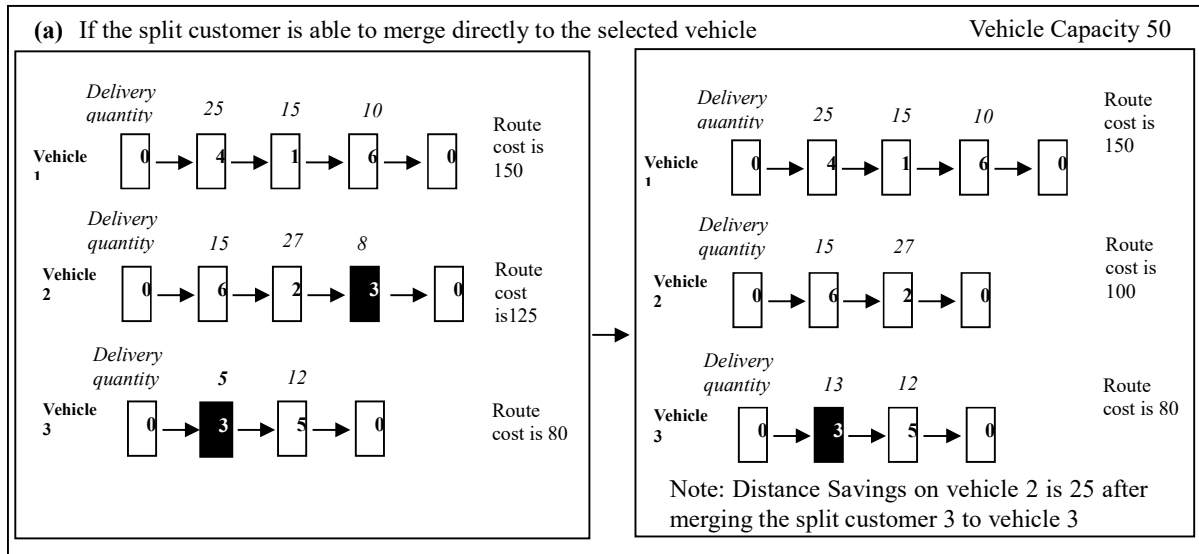


Fig. 2(a): Swap procedure if the split customer can be merged directly to the selected vehicle

### 3.5.2 2 - opt\*

We applied 2 - opt\* [28] heuristic as inter-route optimization procedure. The purpose of this strategy is to test on all possible pairwise exchange between vehicles to see if an overall improvement in the objective function can be attained. The heuristic calculates the distances for all pairwise permutations and compared those distance with the current solution. If any of these solutions is found to improve the objective function, then it replaces the current solution.

### 3.5.3 2-opt

2 - opt [29] heuristic which is an intra-route optimization procedure is implemented. The implementation is slightly different from 2 - opt\*, 2 - opt test on all possible pairwise exchange within a vehicle instead of between vehicles to see if an overall improvement in the objective function can be obtained. The current solution is replaced if the improved solution is better.



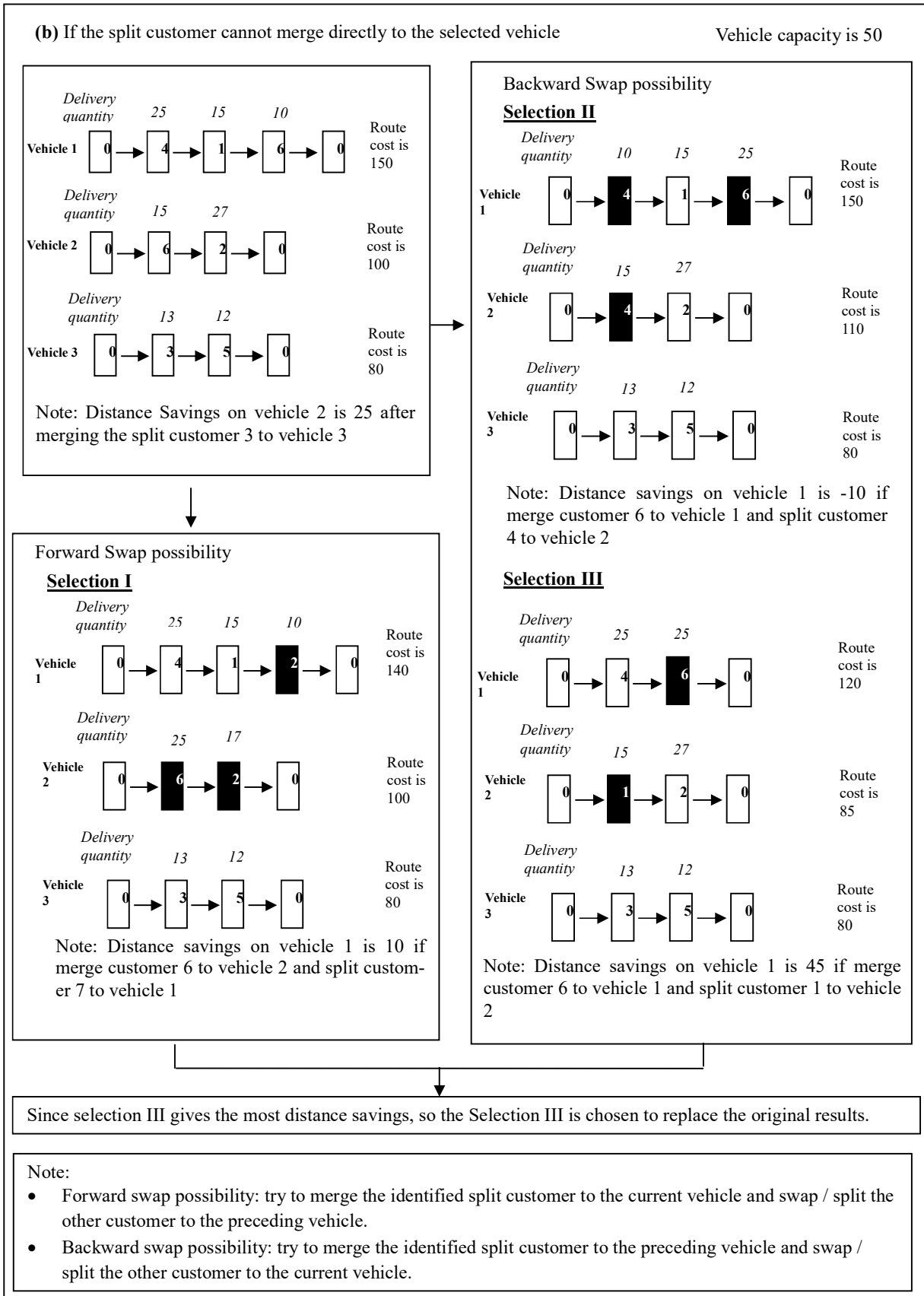


Fig. 2(b): Swap procedure if the split customer cannot be merged directly to the selected vehicle

### 3.6 Updating the Inventory Level

The inventory updating mechanism is applied after a certain predefined number of iterations has been completed. Fig.3 illustrates the process of the inventory updating. First, we randomly select the period to be moved. The number of customers to be moved is limited by some predefined maximum number of moves allowed,  $N\_moveTime$ . The available customers on a selected period  $p$  are those with positive delivery quantities ( $q_{ijp} > 0$ ) and the inventory has not been updated yet (has not received from the period  $p+1$ ). This extra constraint is to ensure that the inventory cost is not excessive. Additional criterion imposed is that the inventory of the preceding the period ( $p-1$ ) has not been updated in the present iteration. The customers who will be selected to update the inventory are those with the least inventory cost. We note that when updating the inventory, there is no restriction imposed, except for the vehicle capacity constraint and may result in an increased in the number of vehicles.

The following definitions are introduced for the procedure of updating inventory level:

$N\_moveData$	maximum number of moves to be allowed for each data
$N\_moveTime$	maximum number of move to be allowed per time
$temp\_move$	the current number of moves
$sum\_move$	the current accumulative moves that have been done
$cur\_move$	Number of moves generated by random number which not more than $N\_moveTime$ per time.

**Step 1:** Check the availability customer on all period. If none of the periods consist of the available customer, go to **Step 9**. Otherwise, go to **Step 2**.

**Step 2:** Randomly select the period,  $p$ , with the condition where there is at least one available customer. Go to **Step 3**.

**Step 3:** Randomly select the number of moves (cannot exceed  $N\_moveTime$ ),  $cur\_move$ :  
 If  $(cur\_move + sum\_move) \leq N\_moveData$   
      $real\_move = cur\_move$   
 else  
      $real\_move = N\_moveData - sum\_move$   
 Set  $temp\_move = 0$ .  
 Go to **Step 4**.

**Step 4:** Select an available customer from period  $p$ , who will give the least inventory cost.  
 Move all the quantity delivery on period  $p$  to period  $p-1$ .  
 $temp\_move++$ .  
 Go to **Step 5**.

**Step 5:** Update the availability of the customer on period  $p$ .  
 If  $(temp\_move < real\_move)$   
     Go to **Step 6**.  
 Else  
     Go to **Step 7**.

**Step 6:** Check is there any available customer on period  $p$ .  
 If yes, go to **Step 4**. Otherwise, go to **Step 7**.

**Step 7:**  $sum\_move += temp\_move$ . Go to **Step 8**.

**Step 8:** Update the inventory level and inventory cost for each customer on each period.

**Step 9:** Select the set of inventory level that had been built for the current best solution to continue with the routing.

Fig. 3: Algorithm for updating inventory level

## 4.0 COMPUTATIONAL RESULTS

The algorithms were written in C++ language by using Microsoft Visual studio 2008. The results of this study are compared with the lower bound (LB) and the upper bound (UB) generated by solving the formulation pre-

sented in Section 2.0 using CPLEX 12.4. All the computations were performed on 3.10 GHz processor with 8GB of RAM.

#### 4.1 Datasets

The algorithm is tested on 12, 20, 50 and 100 customers, and combination with a different number of periods, 5, 10, 14 and 21. The coordinates for each customer are generated randomly in the square of  $100 \times 100$ . The coordinates of each customer for the 20 customers instance comprises the existing 12 customer instance with additional 8 newly randomly generated coordinates. The same procedure is used to create the 50 and 100 customer instances. Fig.4 illustrates the distribution of the data sets. The holding cost for each customer lies between 0 and 10 while the demand for each of the customer is generated randomly between 0 and 50. The vehicle capacity is 100.

#### 4.2 Results and discussion

In this study, we let CPLEX version 12.4 run for a limited time 9000s (2.5 hours) to obtain the lower bound and the best integer solution for all the instances.

##### 4.2.1 Sensitivity analysis

Since the two most important parameters in any ACO are  $\alpha$  and  $\beta$  that control on the decision policy in the selection of customers, we conduct the computational experiments to test on the different combination of parameters  $\alpha = [1, 2, 3]$  and  $\beta = [1, 2, 3, 4, 5]$  to determine the appropriate values of  $\alpha$  and  $\beta$ . The performance of the modified ACO is measured for each data set, and averaged over 5 runs. Table 1 shows the mean and standard deviation over 5 runs of the different combinations of parameter  $\alpha$  and  $\beta$  and parameter  $\alpha$  is represented by alphabet A (A1, A2 and A3) while B(B1, B2, B3, B4 and B5) refers to parameter  $\beta$ . The results show that the combination of  $(\alpha, \beta) = (1, 5)$  gives the best average in the larger data set. However, the best combination of  $\alpha$  and  $\beta$  for S12 and S20 are  $(\alpha, \beta) = (2, 1)$  and  $(\alpha, \beta) = (2, 3)$  respectively. Extra computations are done to compare between using existing parameters  $(\alpha, \beta) = (1, 5)$  and the best parameter settings for instances of S12 and S20 and the results are tabulated in Table 2 (S12) and Table 3 (S20). The improvement in the mean between  $(\alpha, \beta) = (1, 5)$  and  $(\alpha, \beta) = (2, 1)$  of S12 and  $(\alpha, \beta) = (2, 3)$  of S20 are small which is less than 1.7%. Therefore we set the parameter values for  $(\alpha, \beta) = (1, 5)$  for all data sets and the results are tabulated in Table 4.

##### 4.2.2 Results of ACO

The parameters for both versions of ACO (ACO and ACO2) are set as follows:  $q_0 = 0.9, \rho = 0.1, \tau_0 = 1/(N \times L_{NN})$  where  $L_{NN}$  is the total distance obtained from the nearest neighbor algorithm. The algorithm is run for 5000 iterations and each of the iterations consists of 25 ants to build a solution.  $N_{moveData}$  is determined by  $\{\frac{1}{12} \times T \times N\}$  while  $N_{moveTime}$  is set to be equal to 3. The number of vehicles (LB) is calculated by the following formula:

$$\frac{\text{Total demand for all periods}}{\text{capacity of the vehicle}}$$

We performed 10 runs for each data set. Table 1 shows the results of ACO, which only applies Swap and 2-optas local search and ACO2 which includes 2 - opt\* as inter route optimization procedure. Table 4 presents the best total costs, the number of vehicles, the CPU time, the lower bound and the upper bound (best integer solutions) obtained from CPLEX. From Table 4, we observed that the gaps that are calculated as the ratio of the difference between the lower bound and the upper bound to the lower bound, for all the solutions are greater than 10%. This ratio increases as the periods and the number of customers increase. Thus, it is hard to justify the quality of the lower bound obtained by CPLEX. This may due to the lower bound is really loose or the upper bound is rather poor.

From the results, we note that the total costs of the data sets with 100 and 50 customers are less than the upper bound, which mean that the algorithm is able to obtain better results when compared with the upper bound. However, ACO gives less than 9 percent gaps for the small and medium instances with 12 customers and 20 customers. Meanwhile, ACO2 performs equally well for the small and medium instances and producing the gaps of less than 4.5 percent between the results and the best integer solutions. We can conclude that ACO2 out performs ACO in all data sets.

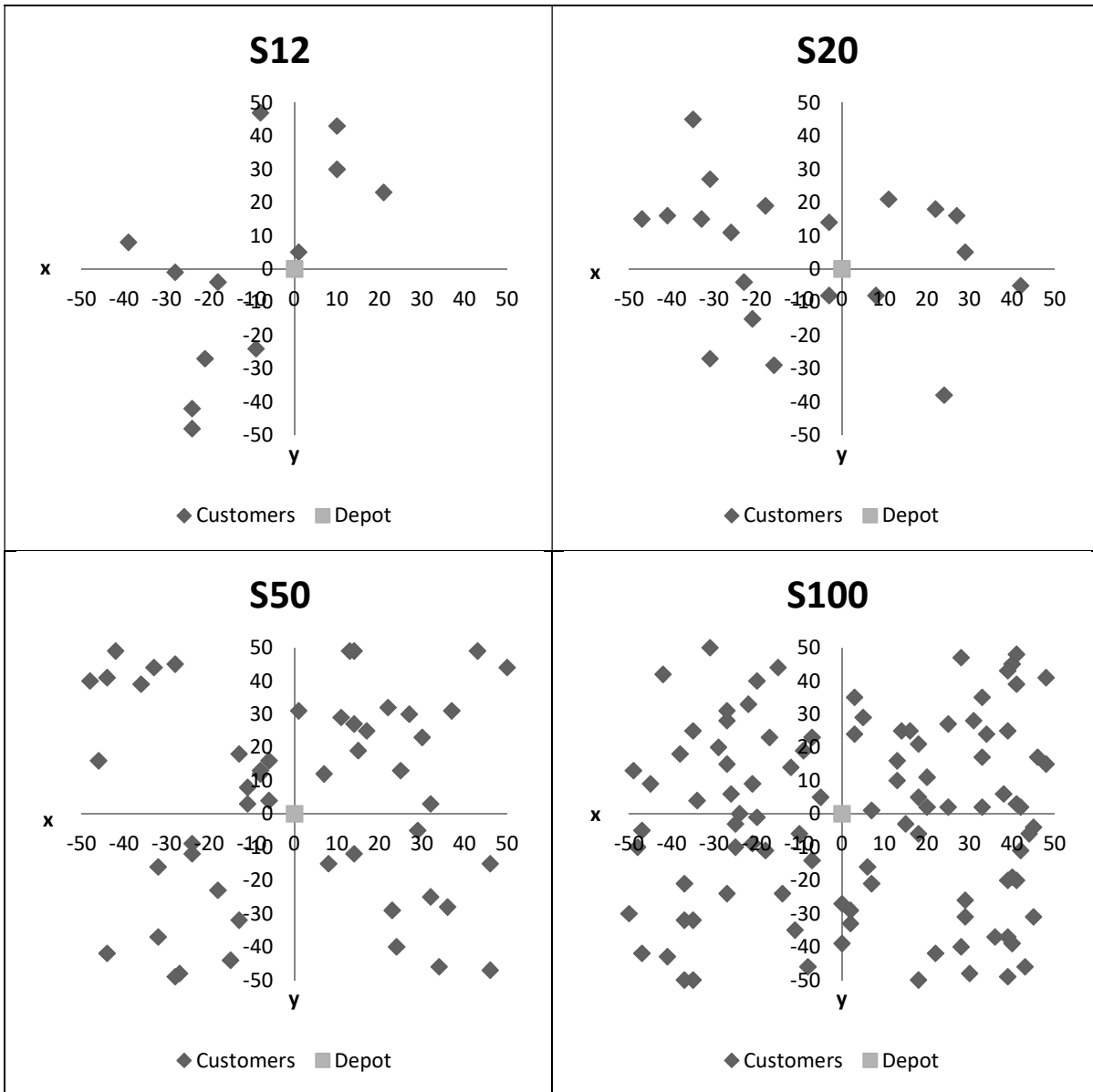


Fig. 4: Distribution of the data sets

Table 1: The results of mean and standard deviation for different parameters settings over 5 runs.

Combination of parameters	MEAN				STDEV			
	S12T14	S20T21	S50T21	S100T14	S12T14	S20T21	S50T21	S100T14
A1B1	6501.44	14397.58	37401.46	45549.38	4.880	49.947	214.086	198.486
A1B2	6480.71	14477.42	36978.60	44780.64	23.131	38.739	76.765	81.299
A1B3	6494.63	14534.18	36825.98	44494.96	26.374	46.013	48.109	57.527
A1B4	6509.66	14545.96	36788.92	44314.14	9.268	20.261	125.782	72.611
A1B5	6502.01	14554.66	36729.76	44216.72	22.996	45.268	77.270	43.506
A2B1	6394.06	14383.48	37162.56	45341.20	20.631	85.372	225.680	341.760
A2B2	6502.29	14336.86	37038.28	45032.00	16.981	46.222	89.375	224.243
A2B3	6505.26	14309.66	36789.50	44693.04	12.281	56.123	180.622	110.700
A2B4	6503.11	14372.46	36919.64	44400.80	13.745	51.106	109.498	155.846
A2B5	6506.18	14545.30	36820.14	44282.44	5.637	70.642	60.905	100.351
A3B1	6410.164	14437.34	37369.62	45386.00	15.728	96.595	139.770	289.770
A3B2	6461.848	14349.34	37116.32	45304.10	42.698	58.585	117.479	117.729
A3B3	6489.428	14357.42	37050.30	44631.50	24.138	96.583	83.243	317.133
A3B4	6502.286	14315.22	36930.08	44531.80	9.043	26.120	106.103	186.044
A3B5	6500.422	14411.86	36842.48	44561.46	30.014	52.133	70.393	47.792

Table 2: Comparison of the results for S12 with two different parameters

Data Sets	LB	UB (Best Integer)	A1B5				A2B1			
			Best costs	Gaps ** (%)	Mean	STDEV	Best costs	Gaps ** (%)	Mean	STDEV
S12T5	2033	2231.96	2290.38	2.62	2296.98	7.682	2254.04	0.99	2272.04	9.976
S12T10	4047.64	4305.33	4453.58	3.44	4512.78	30.855	4409.91	2.43	4459.50	24.550
S12T14	5329.58	6196.35	6462.09	4.29	6505.47	17.702	6361.24	2.66	6403.95	17.333

Gaps\*\* refers to the difference between the obtained results and the CPLEX Upper Bound

Table 3: Comparison of the results for S20 with two different parameters

Data Sets	LB	UB (Best Integer)	A1B5				A2B3			
			Best costs	**Gaps (%)	Mean	STDEV	Best costs	**Gaps (%)	Mean	STDEV
S20T5	3208.35	3394.78	3527.00	3.89	3551.21	11.566	3431.84	1.09	3456.66	16.788
S20T10	6330.97	6759.71	7046.34	4.24	7114.85	36.759	6924.58	2.44	6946.81	25.406
S20T14	8769.73	9368.08	9707.08	3.62	9783.33	52.582	9609.24	2.57	9678.84	39.873
S20T21	12407.58	13929.21	14514.10	4.20	14598.30	82.124	14262.00	2.39	14325.30	53.532

Gaps\*\* refers to the difference between the obtained results and the CPLEX Upper Bound

Table 4: Results for ACO

Data	LB (Objective)	UB (Best Integer)		#veh (LB)	ACO				ACO2			
		Costs	# veh		Best Costs	#veh	Time (secs)	Gap* (%)	Best Costs	#veh	Time (secs)	Gap* (%)
S12T5	2033	2231.96	19	16	2353.04	19	16	5.42	2290.38	19	15	2.62
S12T10	4047.64	4305.33	36	31	4604.56	37	30	6.95	4453.58	36	30	3.44
S12T14	5329.58	6196.35	52	44	6665.05	52	42	7.56	6462.09	52	41	4.29
S20T5	3208.35	3394.78	28	26	3617.39	28	47	6.56	3527	28	47	3.89
S20T10	6330.97	6759.71	56	52	7293.06	56	90	7.89	7046.34	56	91	4.24
S20T14	8769.73	9368.08	77	71	9982.36	77	126	6.56	9707.08	77	128	3.62
S20T21	12407.58	13929.21	115	104	15093.5	113	184	8.36	14514.1	113	188	4.20
S50T5	7614.43	8213.22	64	58	8176.18	59	317	-0.45	8115.38	61	324	-1.19
S50T10	13913.84	17359.2	135	120	17205.7	124	653	-0.88	16935.4	124	664	-2.44
S50T14	19300.45	25181.61	197	171	24357.1	176	942	-3.27	23969.1	178	941	-4.82
S50T21	29418.86	38626.96	311	261	37485.6	272	1438	-2.95	36620.4	273	1432	-5.19
S100T5	13208.54	16130.13	134	120	15247.6	122	1709	-5.47	15117	122	1734	-6.28
S100T10	25601.69	34388.15	293	245	31407.6	249	3527	-8.67	30963.9	249	3517	-9.96
S100T14	-	-	-	-	44610.5	355	4960	-	44155	355	4956	-

Gaps\* refers to the difference between the obtained results and the CPLEX Upper Bound

## 5.0 CONCLUSION

The integration of inventory and transportation plays an important role in supply chain management. This paper presents the formulation of the model that consists of multi-products and multi-periods IRP as well as the development of a modified ACO that includes the inventory component in the pheromone updating. A new transfer/swap aims at combining the split customers is also proposed. This is carried out in order to obtain the improvement in term of the transportation costs. Consequently, the number of vehicles, when compared to the number of vehicles for the best integer, is much reduced especially for large instances. For small instances the number of vehicles is comparable. The lower bound for the number of vehicles is calculated using the formula

$$\left\lceil \frac{\sum_{t=1}^T \sum_{i=1}^n d_{it}}{C} \right\rceil$$

that is the total demands in all periods divide by the vehicle capacity. This lower bound is used to gauge the number of vehicles obtained by the CPLEX and the algorithms. Generally, our algorithms, ACO and ACO2 performs well in large instances and less than 5% difference for small instances.

We note that both ACO and ACO2 are more likely to give savings in term of transportation costs instead of inventory costs even though we did embedded the inventory cost component in the global pheromones updating. We observed that this is due to larger component of the total transportation costs than the inventory costs. Therefore, improvement can be obtained by selecting appropriate weights to balance between transportation and inventory costs. However selecting appropriate weights is very complex.

## 6.0 ACKNOWLEDGMENT

This work has been supported by Exploratory Research Grant ERGS (ER004-2013A) and Postgraduate Research Fund (PG020-2012B).

## REFERENCES

- [1] L. C. Coelho, and G. Laporte, "A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem". *International Journal of Production Research*, Vol. 5, No. 23-24, 2013, pp. 7156-7169.
- [2] N.H. Moin, S.Salhi and N.A.B.Aziz, "An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem". *Int. J. Production Economics*, Vol. 133, 2011, pp. 334-343.
- [3] A. Mjirda, B. Jarboui, R. Macedo, S. Hanafi and N.Mladenović, "A two phase variable neighborhood search for the multi-product inventory routing problem". *Computers and Operations Research*, <http://dx.doi.org/10.1016/j.cor.2013.06.006i>, 2013.
- [4] Y. Yu, H. Chen, and F. Chu, "A new model and hybrid approach for large scale inventory routing problems". *European Journal of Operational Research*, Vol. 189. No. 3, 2008, pp. 1022-1040.
- [5] D.Adelman, "A price-directed approach to stochastic inventory/routing". *Operations Research*, Vol. 52, No. 4, 2004, pp. 499-514.
- [6] A. J.Kleywegt, V. S. Nori, and M. W. Savelsbergh, "The stochastic inventory routing problem with direct deliveries". *Transportation Science*, Vol. 36, No.1, 2002, pp. 94-118.
- [7] A. J. Kleywegt, V. S. Nori, and M. W. Savelsbergh, "Dynamic programming approximations for a stochastic inventory routing problem". *Transportation Science*, Vol. 38, No. 1, 2004, pp. 42-70.
- [8] Y. Yu, C. Chu, H. Chen, and F.Chu, "Large scale stochastic inventory routing problems with split delivery and service level constraints". *Annals of Operations Research*, Vol. 197, No. 1, 2012, pp. 135-158.
- [9] E. H. Aghezzaf, B.Raa, and H.Van Landeghem, "Modeling inventory routing problems in supply chains of high consumption products". *European Journal of Operational Research*, Vol. 169, 2006, pp. 1048-1063.
- [10] S.Anily, and J.Bramel, "A Probabilistic Analysis of a Fixed Partition Policy for the Inventory-Routing Problem". *Naval Research Logistics*, Vol. 51, 2004, pp. 925-948.DOI: 10.1002/nav.20031.
- [11] A.M. Campbell and M.W.P Savelsbergh, "A decomposition approach for the inventory-routing problem". *Transportation Science*, Vol. 38, No. 4, 2004, pp. 488-502.

- [12] A. Federgruen and P. Zipkin, "A combined vehicle routing and inventory allocation problem". *Operations Research*, Vol. 32, 1984, pp. 1019–1036.
- [13] T.W.Chien, A.Balakrishnan, and R. T. Wong, "An integrated inventory routing allocation and vehicle routing problem". *Transportation Science*, Vol. 23, No. 2, 1989, pp. 67–76.
- [14] M.Dror and P.Trudeau, "Savings by split delivery routing". *Transportation Science*, Vol. 23, 1989, pp. 141-145.
- [15] M. Dror and P.Trudeau, "Split delivery routing". *Naval Research Logistics*, Vol. 37, 1990, pp. 383-402.
- [16] C. H. Lee, Y. A. Bozer and C. C. White III, "A heuristic approach and properties of optimal solutions to the dynamic inventory routing problem". *Toronto, Ontario, Canada*, 2003.
- [17] O. C. Sin, N. H. Moin, and M. Omar, "Multi Parents Extended Precedence Preservative Crossover for Job Shop Scheduling Problems". *Malaysian Journal of Computer Science*, Vol. 26, No. 3, 2013.
- [18] H. J. F. Huacuja, G. C. Valdez, R. A. P. Rangel, J. G. Barbosa, L. C. Reyes, J. M. C. Valadez, H. J. P. Soberanes and D. T. Villanueva, Scatter Search with Multiple Improvement Methods for the Linear Ordering Problem. *Malaysian Journal of Computer Science*, Vol. 25, No. 2, 2012.
- [19] A. Rasheed, S. Ajmal and A. Qayyum, "Adaptive Routing Update Approach for Vanet Using Local Neighbourhood Change Information". *Malaysian Journal of Computer Science*, 27, 4, 2014.
- [20] S. H. Huang and P. C. Lin, "A modified ant colony optimization algorithm for multi-item inventory routing problem with demand uncertainty". *Transportation Research Part E*, Vol. 46, 2010, pp. 598 – 611.
- [21] M.M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints". *Operations Research*, Vol. 35, No. 2, 1987, pp. 254-265.
- [22] H. I. Calvete, C. Galé, and M. J. Oliveros, "Bilevel model for production-distribution-planning solved by using ant colony optimization". *Computers and Operation Research*, Vol. 38, 2010, pp. 320 – 327.
- [23] V. A. Tatsis, K. E. Parsopoulos, K. Skouri, and I. Konstantaras, "An Ant-Based Optimisation Approach for Inventory Routing". M. Emmerich et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV*, Advances in Intelligent Systems and Computing 227, DOI: 10.1007/978-3-319-01128-8\_8, c\_ Springer International Publishing Switzerland 2013, 2013
- [24] L. Wong, and N. H. Moin. "A modified ant colony optimization to solve multi products inventory routing problem", in Proceedings of The 21st National Symposium on Mathematical Sciences (SKSM21): *Germination of Mathematical Sciences Education and Research towards Global Sustainability*, Vol. 1605, July 2014, pp.1117-1122. AIP Publishing.
- [25] M. Dorigo, "Learning and natural algorithms.", Ph.D. dissertation, Politecnico di Milano, 1992.
- [26] M.Dorigo, and C.Blum, "Ant Colony Optimization Theory: A Survey". *Theoretical Computer Science*, Vol. 344, No. 2-3, 2005, pp. 243 – 278.
- [27] M. Dorigo, and G. Di Caro, "Ant colony optimization: a new metaheuristics", in Proceeding of the Congress on Evolutionary Computation, Vol. 2, 1999, pp. 1470–1477.
- [28] J-Y Potvin, and J-M Rousseau, "An Exchange Heuristic for Routing Problems with Time Windows". *Journal of the Operational Research Society*, Vol. 46, 1995, pp.1433-1446.
- [29] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem". *Operations Research*, Vol. 21, 1973, pp. 498-516.