A Modified Ant Colony Optimization to Solve Multi Products Inventory Routing Problem

Conference Paper · July 2014

DOI: 10.1063/1.4887747

Noor Moin
University of Malaya

A modified ant colony optimization to solve multi products inventory routing problem

Lily Wong and Noor Hasnah Moin

Citation: AIP Conference Proceedings **1605**, 1117 (2014); doi: 10.1063/1.4887747
View online: http://dx.doi.org/10.1063/1.4887747
View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1605?ver=pdfcov
Published by the AIP Publishing

A Modified Ant Colony Optimization to Solve Multi Products Inventory Routing Problem

Wong, Lily and Moin, Noor Hasnah

Institute of Mathematical Sciences, University of Malaya, 50603 Kuala Lumpur, Malaysia

Abstract. This study considers a one-to-many inventory routing problem (IRP) network consisting of a manufacturer that produces multi products to be transported to many geographically dispersed customers. We consider a finite horizon where a fleet of capacitated homogeneous vehicles, housed at a depot/warehouse, transport products from the warehouse to meet the demand specified by the customers in each period. The demand for each product is deterministic and time varying and each customer requests a distinct product. The inventory holding cost is product specific and is incurred at the customer sites. The objective is to determine the amount on inventory and to construct a delivery schedule that minimizes both the total transportation and inventory holding costs while ensuring each customer's demand is met over the planning horizon. The problem is formulated as a mixed integer programming problem and is solved using CPLEX 12.4 to get the lower and upper bound (best integer solution) for each problem considered. We propose a modified ant colony optimization (ACO) to solve the problem and the built route is improved by using local search. ACO performs better on large instances compared to the upper bound.

Keywords: ant colony optimisation, inventory, routing
PACS: 02.70.-c, 02.10.Ox

INTRODUCTION

This Inventory routing problem (IRP) is a problem that involves the integration and coordination of two components of supply chain management which are the inventory management and the transportation. This usually, delivers to customers based on their demands, from a warehouse or plants, to provide the commodity on a repeated basis. The main objective of this problem is to minimize the corresponding costs (fixed and variable costs) whilst ensuring that the delivery time are met. Generally, IRP can be categorized by planning horizon, single or multi-periods, and the demand is whether deterministic or stochastic. In the literature, there are many meta-heuristic methods, such as genetic algorithms, tabu search and branch-and-cut algorithms that have been modified to suit the problems that lead to optimal or near optimal solutions. We will first present some literatures on IRP concentrating on the most recent literatures.

Federgruen and Zipkin [1] were among the first to study the inventory routing problem where the problem decomposes into a nonlinear inventory allocation problem which determines the inventory and shortage costs and a Travelling Salesman Problem (TSP) for each vehicle considered which produces the transportation costs. Chien *et al.* [2] were among the first to stimulate a multiple period planning model where the model was based on a single period approach. This is achieved by passing some information from one period to the next through inter-period inventory flow. Recently, Yu et al. [3] developed an approximate approach that incorporates Lagrangian relaxation to solve a large scale IRP with split delivery and vehicle fleet size constraint. Moin et al. [4] had proposed an efficient hybrid genetic algorithm to solve the IRP which involves multi products where each supplier supplies different products and in multi period scenario. There are some developments on modifying ACO to solve their proposed IRP models. (see Huang and Lin [5] and Calvete et al. [6]).

In this study, we develop the ACO which was initially proposed by Dorigo and coworkers (Dorigo [7], Dorigo and Blum [8], and Dorigo and Caro [9]) and was inspired by the food-foraging behavior of ant. The network consists of a warehouse that supplies multi product to geographically dispersed customers and the product are transported by a fleet of homogeneous vehicles. We assumed that the customer's demand must be met on time and we allow the customer to be served by more than one vehicle. We modified the algorithm by incorporating the inventory component in the global updating scheme which differs from the classical ACO.

This paper is organized as follows: The problem formulation including the assumptions that are made in this model and the solution procedure that is based on ACO is described in detail in the section on methodology. Then, it is followed by the computational results and discussion. Finally, the conclusion is presented in the last section.

Proceedings of the 21st National Symposium on Mathematical Sciences (SKSM21)
AIP Conf. Proc. 1605, 1117-1122 (2014); doi: 10.1063/1.4887747

METHODOLOGY

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.
- 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. 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

$\tau = \{1, 2, \ldots, T\}$ period index

$W = \{0\}$ warehouse/depot

$S = \{1, 2, \ldots, N\}$ 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 ∞ (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 customer *i* at the end of period *t*

q_{ijt} quantity transported through the directed arc (ij) in period *t*

x_{ijt} number of times that the directed arc (ij) is visited by vehicles in period *t*

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

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

subject to

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

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

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

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

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

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

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

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

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

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

The objective function (1) includes the inventory costs (I), the transportation costs (II) and vehicle fixed cost (III). The equation (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 subtours. (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.

Ant Colony Optimisation (ACO) is inspired by the nature behavior of ants finding the shortest path between their colony and a source of food. The information collected by ants during the searching process is stored in pheromone trails. The higher density of pheromones on an arc leads to attract more ants to the arc. Therefore, an appropriate formulation associated to the model for updating pheromones trail is very crucial.

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 Figure 1 outlines the algorithm and the following definitions are required:

τ_0	the amount of pheromone deposited (an initial pheromone value assigned to all arcs).
Q	a random number generated from a uniform distribution over the interval (0, 1).
q_0	a predefined real number where $0 \leq q_0 \leq 1$.
τ_{ij}	refer to the pheromone value allocated on arc (i, j).
η_{ij}	$\frac{1}{c_{ij}}$ where c_{ij} is the length of arc (i, j).
α, β	the parameters to control the influence of the pheromone value allocated on arc (i, j) and the desirability of arc (i, j) respectively.
Ω_i	all the arcs connected to unvisited node j (unvisited customer) such that the ants in node i passing through arc (i, j) will not violate any constraint.

Nearest Neighbor algorithm (NN) which tends to find the closest customer to another customer is adopted to generate initial solutions. In this study, NN is modified to allow for split delivery. The total distance obtained by NN is embedded to initialize the τ_0 in the local pheromone updating. A simple NN is adopted since the initial solution is used to approximate the τ_0 and any other methods can be adopted as well. The route construction begins by setting the value of all the parameters α , β, τ_0, q_0 and ρ. Note the value of τ_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 which 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).

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

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

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

Local updating is used to reduce the amount of pheromone on all the visited arcs in order to simulate the natural evaporation of pheromone and it is intended to avoid a very strong arc being chosen by all the ants. After 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).

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

where ρ represent the rate of pheromone evaporation.

After a predefined number of iterations, *N_GL*, the ACO updates the pheromone allocation on the arcs to compose the current optimum route γ^{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 uses the arcs contained in the best solutions. The classical ACO takes into account the transportation cost only. Since the IRP tries to find a balanced between the transportation and inventory cost, it is natural to incorporate the inventory holding cost in the formulation. The global update rule is as follows:

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

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 and is given by

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

The routes can be further improved by adding some strategies in the procedure. Here, we adopted the route improvement strategies that focus on the split customers and they comprise 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 from the preceding vehicle or to the current selected vehicle that results in the least transportation cost is carried out. If none of the swap provides an improvement in the objective value than the solution built by ACO, the 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 merged results the improved in the objective value. We also introduces the $2-opt$ [10] heuristic as intra route optimization procedure. The purpose of this strategy is to test on all possible pair wise exchange within a vehicle to see if an overall improvement in the objective function can be attained. The heuristic calculates the distances for all pair wise permutations and compared to those distance obtained by ACO. If any of these solutions is found to improve the objective function, then it replaces the current solution.

After predefined number of iterations, *N_DEM*, the inventory level is updated. The followings are the definitions introduced for the procedure of updating inventory level:

N_moveData	maximum number of moves to be allowed for each data
N_moveTime	the 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 move generated by random number which not more than *N_moveTime* per time.

The inventory updating process is done only if the *sum_move* is less than *N_moveData*. Otherwise, the existing inventory level of the current best solution is selected to continue the routing. The available customers on a selected period p are those customers with positive demands ($q_{ijp}>0$) and the inventory has not been updated yet in the previous iterations (has not received from period $p+1$). Additional criterion imposed is that the inventory of the preceding period ($p-1$) has not been updated the present iteration. FIGURE 1 shows all the steps of the process for updating the inventory level. We note that when updating the inventory, there is no restriction imposed and may result in an increased in the number of vehicles.

Step 1: Check the availability customer on all period. If none of the period consists of 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*) <= *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.

FIGURE 1 Algorithm of updating inventory level

RESULTS AND DISCUSSION

The algorithms were written in C++ language by using Microsoft Visual studio 2008. The results of this study is compared with the lower bound (LB) and the upper bound (UB) generated by solving the formulation presented in the section of methodology using CPLEX 12.4.

The algorithm is tested on 12, 20, 50 and 100 customers, and combination with different number of periods, 5, 10, 14 and 21. The coordinates for each customer is generated randomly in the square of 100×100. The coordinates of each customer for the 20 customer instance comprise the existing 12 customer instances with additional 8 newly randomly generated coordinates. The same procedure is used to create the 50 and 100 customer instances. 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.

For all the instances, we let CPLEX 12.4 run for a limited time 9000s (2.5 hours) in order to obtain the lower bound and the best integer solution. The results which generated by the developed method are shown in Table 1. The parameters are set as follows: $\alpha = 1, \beta = 5, q_0 = 0.9, \rho = 0.1,\ \tau_0 = 1/L_{nn}$ where L_{nn} is the total distance obtained from nearest neighbor algorithm. The values of α and β of the set parameters is taken from Dorigo et al.[11]. The algorithm is run for 5000 iterations and each of the iteration 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 #veh (LB) shown in the table is calculated using the following formula:

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

We performed 10 runs for each data set. Table 1 presents the best total costs, the number of vehicles, the CPU time, the lower bound and the upper bound (best integer solutions) which are obtained from CPLEX. From Table 1, we observed that the gaps which is calculated as the ratio of the difference between the lower bound and the upper bound to the lower bound, for all the solutions is 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 shown in Table 1, we note that the total costs of 100 customers as well as 50 customers with 14 and 21 periods are less than the upper bound. The algorithm is able to obtain better results when compared with the upper bound for 100 customers as well as 50 customers with 14 and 21 periods. Moreover, the gap between the best costs of 50 customers with 5 and 10 periods and the best integer is less than 2 percent. However, the algorithm produced the gap between the best costs and the upper bound is around 10 percent for the small and medium instances. From the overall observation, the algorithm performs better for larger instances.

TABLE 1 Results for ACO

Data	LB (Objective)	UB (Best Integer)		#veh (LB)	ACO			
		Costs	# veh		Best Costs	#veh	Time (secs)	Gap* (%)
S12T5	2033	2231.96	19	16	2372.19	19	22	6.28
S12T10	4047.64	4305.33	36	31	4674.25	37	44	8.57
S12T14	6146.52	6176.02	51	44	6784.56	53	60	9.85
S20T5	3208.35	3394.78	28	26	3684.51	28	63	8.53
S20T10	6330.97	6759.71	56	52	7467.37	56	124	10.47
S20T14	8769.73	9368.08	77	71	10091.6	77	172	7.72
S20T21	12407.58	13929.21	115	104	15289.5	113	253	9.77
S50T5	7614.43	8213.22	64	58	8348.58	59	433	1.65
S50T10	13913.84	17359.2	135	120	17590.6	124	882	1.33
S50T14	19300.45	25181.61	197	171	24891.4	178	1278	-1.15
S50T21	29418.86	38626.96	311	261	38281.5	272	1956	-0.89
S100T5	13208.54	16130.13	134	120	15498.5	121	2602	-3.92
S100T10	25601.69	34388.15	293	245	31895	249	5333	-7.25
S100T14	-	-	-	348	45569.8	355	7575	-

Gap* is refer to the gap between the obtained results and UB

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 which includes the inventory component in the IRP. Transfer/swap and 2 - *op*t are applied to improve all the routes. In this study, the overall results show that the algorithm performs well in the larger instances if compared with the small and medium instances. In future research, more powerful route improvements strategies can be incorporated in the algorithm in order to produce better outcomes.

ACKNOWLEDGMENT

The first author would like to thank the Ministry of Higher Education, Malaysia and Universiti Malaya for payment of the registration fees as well as the travelling expenses via Fundamental Grantt FRGS (F041/2010B) and Postgraduate Research Fund (PG020-2012B).

REFERENCES

1. A. Federgruen and P. Zipkin, *Oper Res*, **32** , 1019–1036 (1984).
2. T. W. Chien, A. Balakrishnan and R.T. Wong, *Transport Sci,*, **23** (2), 67–76 (1989).
3. Y. G. Yu, H. X Chen and F. Chu, *Eur J Oper Res*, **189**, 1022-1040 (2008).
4. N. H., Moin, S. Salhi and N. A. B. Aziz, *Int J Prod Econ*, **133**, 334-343 (2011).
5. S. H. Huang and P. C. Lin, *Transport Res E*, **46**, 598 – 611 (2010).
6. H. I. Calvete, Galé, C. and M. J. Oliveros, *Comput Oper Res*, **38**, 320 – 327 (2010).
7. M. Dorigo, “Learning and natural algorithms”, Ph.D. dissertation, Politecnico di Milano, 1992.
8. M. Dorigo and C. Blum, *Theor Comput Sci*, **344** (2-3), 243 – 278 (2005).
9. M. Dorigo and G. D. Caro, *IEEE C Evol Computat,* **2**, 1470-1477 (1999).
10. S. Lin and B. W. Kernighan, *Oper Res*, **21**, 498-516 (1973).
11. M. Dorigo, V. Maniezzo and A. Colorni, *IEEE T Syst Man Cy B*, **26** (1), 29–41 (1996).