Int. J. Production Economics 141 (2013) 34–44

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

A multiple ant colony optimization algorithm for the capacitated location routing problem

Ching-Jung Ting [a,*], Chia-Ho Chen [b,1]

[a] Department of Industrial Engineering and Management, Yuan Ze University, 135 Yuan-Tung Road, Chung-Li, Taiwan 32003, ROC
[b] Splendor Game Technology Co., LTD. 4F.-5 No. 360 Jianxing Road, North District, Taichung, Taiwan 40459, ROC

ARTICLE INFO

Article history:
Received 8 November 2011
Accepted 10 June 2012
Available online 17 June 2012

Keywords:
Logistics system
Location routing problem
Multiple ant colony optimization
Nested method

ABSTRACT

The success of a logistics system may depend on the decisions of the depot locations and vehicle routings. The location routing problem (LRP) simultaneously tackles both location and routing decisions to minimize the total system cost. In this paper a multiple ant colony optimization algorithm (MACO) is developed to solve the LRP with capacity constraints (CLRP) on depots and routes. We decompose the CLRP into facility location problem (FLP) and multiple depot vehicle routing problem (MDVRP), where the latter one is treated as a sub problem within the first problem. The MACO algorithm applies a hierarchical ant colony structure that is designed to optimize different subproblems: location selection, customer assignment, and vehicle routing problem, in which the last two are the decisions for the MDVRP. Cooperation between colonies is performed by exchanging information through pheromone updating between the location selection and customer assignment. The proposed algorithm is evaluated on four different sets of benchmark instances and compared with other algorithms from the literature. The computational results indicate that MACO is competitive with other well-known algorithms, being able to obtain numerous new best solutions.

1. Introduction

The design of a logistics system is an important issue in today's competitive environment due to the significant contribution of the distribution cost to the total supply chain cost. This kind of problem is commonly solved in two phases: facility location for a long term policy and vehicle routing to satisfying customer demands for the operational decisions. These two components can be treated separately, but may lead to suboptimal solutions (Salhi and Rand, 1989). The location routing problem (LRP) integrates facility location problem (FLP), which determines the depot locations and allocates customers to each selected depot, and vehicle routing problem (VRP), which constructs the vehicle routes of the selected depot. Several real world applications can be found in the literature, for example, bill delivery (Lin et al., 2002), parcel delivery (Wasner and Zäpfel, 2004), and mobile network design (Billionnet et al., 2005). The LRP with capacities on both depots and routes is called capacitated LRP (CLRP) which is the focus of this paper.

The CLRP can be represented by a graph $G=(V, E)$, where $V=I\cup J$. $I=\{1, \ldots, n\}$ is the set of customer nodes and $J=\{1, \ldots, m\}$ denotes the set of candidate depot locations. Each customer $i\in I$ has a demand d_i. A capacity R_j and an opening cost f_j are associated with each candidate depot site $j\in J$. Associated to each edge $(i, j)\in E$ there is a routing cost c_{ij} which denotes the traveling distance or traveling cost between nodes i and j. A set K of homogeneous vehicles with capacity Q and cost C are available. Each customer must be served exactly once by only one vehicle. Each route must begin and end at the same depot and its total load cannot exceed vehicle capacity. The total load of the vehicles assigned to a depot cannot exceed the capacity of that depot. The objective is to find the optimal number and locations of the depots as well as the vehicle routes of each opened depot so as to minimize the sum of the fixed facility costs, transportation costs, and vehicle costs.

The CLRP is very difficult to solve since it encompasses two NP-hard problems: facility location problem and vehicle routing problem (Garey and Johnson, 1979). In CLRP, the location–allocation decision will influence the total cost of vehicle routes and the architecture of vehicle routes will affect the location of depots and allocation of customers. Consequently, how to deal with the interdependence between these decisions is an important issue. In this paper, we solve both location and routing problems simultaneously rather than independently with nested methods based on the ant colony optimization algorithm. We apply a hierarchical structure, with facility location as the main problem and vehicle routing as a subordinate one. To wit,

* Corresponding author. Tel.: +886 3 4638800 2526; fax: +886 3 4638907.
E-mail addresses: ietingcj@saturn.yzu.edu.tw (C.-J. Ting), Chiaho@gosmio.biz (C.-H. Chen).
[1] Fax: +886 4 22361120.

0925-5273/$ - see front matter
http://dx.doi.org/10.1016/j.ijpe.2012.06.011

we decompose the CLRP into facility location and multi-depot vehicle routing problem, while the latter problem is embedded into the first one. This concept of hierarchy is also emphasized by Balakrishnan et al. (1987) and Nagy and Salhi (1996). The proposed multiple ant colony optimization algorithm (MACO) is evaluated by four sets of CLRP benchmark instances from the literature and its computational results are compared with state-of-the-art algorithms.

The remainders of this paper are organized as follows. Section 2 provides an extensive review of LRP in the literature. The multiple ant colony optimization algorithm to tackle the CLRP is described in Section 3. In Section 4, the computational results of four groups of benchmark problems are reported. For each benchmark set we compare to the best available algorithms. Finally, conclusions are followed in Section 5.

2. Literature review

The LRP has been studied for decades, there are a few LRP surveys in the literature (Laporte, 1988; Min et al., 1998; Nagy and Salhi, 2007). Laporte (1988) reviewed early research on location routing problems and summarized different types of formulations, solution algorithms and computational results of research published prior to 1988. Min et al. (1998) synthesized the past evolution of location routing literature and explored promising research opportunities in incorporation of more realistic aspects, algorithmic design and model complexity. Recently, Nagy and Salhi (2007) surveyed the state of the art in location routing problem. They proposed a classification scheme and looked at a number of problem variants. They also investigated exact and heuristic algorithms and presented some suggestions for future research.

Most early work on LRP considers either capacitated routes or capacitated depots, but not both (Laporte et al., 1988; Chien, 1993; Srivastava, 1993; Tuzun and Burke, 1999). Recently, a number of studies have been devoted to the case with capacitated depots and routes (Wu et al., 2002; Prins et al., 2006a, 2006b; Bouhafs et al., 2006; Prins et al., 2007; Barreto et al., 2007; Duhamel et al., 2010; Yu et al., 2010). Our study also considers both depot and route capacities.

Several exact methods have been devoted to solve the LRP, but optimal solutions are only limited to medium-scale or to basic uncapacitated instances. Laporte and Norbert (1981) designed a branch-and-bound algorithm for an LRP with a single open depot, and solved instances with up to 50 customers. In Laporte et al. (1986), the solution to an LRP with vehicle capacity constraints is obtained by a branch-and-cut method. Subtour elimination constraints and chain-barring constraints guarantee that each route starts and ends at the same facility. Laporte et al. (1988) addressed an LRP with asymmetrical costs, in which vehicle capacity is replaced by a maximum route length. They elaborated a branch-and-bound algorithm that is able to solve instances with up to 40 customers, but the number of depots is small (2 or 3) and the number of routes per opened depot is limited to 2. Akca et al. (2009) presented a set-partitioning based formulation of the LRP and proposed a column generation approach to solve instances with up to 40 customers. Belenguer et al. (2011) proposed a branch-and-cut algorithm based on a zero-one linear model strengthened by new families of valid inequalities for solving the CLRP. They solved instances to optimality with up to 50 customers and five depots. Baldacci et al. (2011) proposed a branch-and-cut-and-price algorithm for solving the CLRP based on a set-partitioning-like formulation of the problem. The lower bounds produced based on dynamic programming and dual ascent methods, are used by an algorithm that decomposes the LRP into a limited set of MDVRP. The bounds provided by their model are very tight, being able to solve instances with up to 199 customers and 14 facilities.

As the LRP problem is NP-hard, most of the researches used heuristics to solve the LRP. Nagy and Salhi (2007) classified the heuristics into four different types as follows: sequential, clustering-based, iterative, and hierarchical. Sequential methods solve the location problem by minimizing the sum of facility to customer distance and the routing problem based on the selected depots sequentially. Clustering-based methods (Srivastava, 1993; Barreto et al., 2007) partition the customers into clusters and then find a depot for each cluster. The VRP is then solved for each cluster. Iterative methods (Hansen et al., 1994; Wu et al., 2002; Prins et al., 2007; Duhamel et al., 2010) decompose the LRP into two subproblems. Then, subproblems are solved iteratively by feeding information from one subproblem to the other. Hierarchical methods (Nagy and Salhi, 1996; Albareda-Sambola et al., 2005) consider the location problem as the main problem and the VRP as a subordinated problem. Nagy and Salhi (2007) believed that hierarchical methods may provide better solutions. Based on their observation, the proposed MACO in this paper is a hierarchical method.

Many heuristics that hybrid two different heuristic approaches are proposed in the literature. Tuzun and Burke (1999) proposed a two-phase tabu search (TS) approach for the LRP. One phase seeks a good facility configuration while the other one obtains a good routing for this configuration. Wu et al. (2002) presented a combined TS and simulated annealing (SA) decomposition approach to solve the multi-depot location routing problem with multiple fleet types and limited number of vehicles for each vehicle type. Lin et al. (2002) developed a meta-heuristic approach based on threshold accepting (TA) and SA to assist in making decisions of facility location, vehicle routing and loading decision for bill delivery services in Hong Kong.

Albareda-Sambola et al. (2005) proposed another two-phase TS heuristic for the LRP which incurs not capacity constraints on vehicles. Wang et al. (2005) proposed a two-phase hybrid heuristic which decomposes the LRP into location–allocation problem and vehicle routing problem. In the location phase, the TS was applied to obtain the configuration of facility locations. For each selected facility location, a vehicle routing problem was solved by ACO in the routing phase. Bouhafs et al. (2006) proposed a hybrid algorithm which combined the SA and ant colony system (ACS) to solve the CLRP. A good configuration of facilities was first found by the SA, and then the ACS was applied to construct the routings based on the configuration. These two ACO-related heuristics construct the routing problem and feed back the information for the facility selection phase.

Prins and his coworkers conducted different heuristic methods to the LRP. Prins et al. (2006a) combined greedy randomized adaptive search procedures (GRASP) and path relinking to develop a two phase algorithm for the capacitated location routing problem. In the first phase, the GRASP and a learning process were implemented to select depots. The second phase was to generate new solutions using a path relinking. Later, Prins et al. (2006b) presented a memetic algorithm with population management (MA|PM) to solve the same problem. Prins et al. (2007) proposed a cooperative approach, which combines the Lagrangean relaxation and granular tabu search (GTS), to solve the capacitated LRP. The algorithm alternates between a location subproblem, solved by Lagrangean relaxation, and a multi-depot VRP, solved by the GTS. Duhamel et al. (2010) presented a GRASP with evolutionary location search (GRASP $\times$ ELS) approach for the CLRP.

Barreto et al. (2007) integrated several hierarchical and non-hierarchical clustering techniques in a sequential heuristic algorithm. Marinakis and Marinaki (2008) developed a hybrid algorithm (HybPSO), which combined the particle swarm optimization (PSO), multiple phase neighborhood search-greedy randomized adaptive search procedure (MPNS-GRASP), the expanding neighborhood search (ENS) and path relinking, to solve the location routing problem. More recently, Yu et al. (2010) proposed a simulated annealing algorithm to solve the LRP. Both Yu et al. (2010) and Duhamel et al. (2010) are, on average, the most effective algorithms on benchmark instances from the literature.

The LRP is generally considered as a deterministic case in the literature. A few researches have addressed stochastic versions of the LRP (Laporte et al., 1989; Chan et al., 2001). Berman et al. (1995) provided good survey on the stochastic LRP.

3. Multiple ant colony optimization for CLRP

This section introduces the proposed algorithm for solving the CLRP. The proposed multiple ant colony optimization (MACO) algorithm is based on the ant colony system (ACS) heuristic. The ant system (AS) was the first ACO algorithm proposed by Dorigo et al. (1996). Subsequently, many variants of ACO have been developed and applied extensively in the fields of the combinatorial optimization problems. Descriptions of available ACO algorithms and related literature review can be obtained in Dorigo and Stützle (2004).

The concept of multiple ant colonies was first proposed by Gambardella et al. (1999) to solve the vehicle routing problem with time windows (MACS-VRPTW). Two ant colonies were designed to successively optimize two different objective functions: the first colony is used to minimize the number of vehicles, while the second colony minimizes the traveled distances. Our MACO adopts a hierarchical ACO structure with different transition rules for different ant colonies. The upper level is for the location selection subproblem, while the lower level is for the MDVRP subproblem. The lower level is further decomposed into customer assignment and vehicle routing subproblems. The coordination between the upper and lower level is achieved through the global pheromone updating rule.

3.1. General structure

A solution of the CLRP consists in defining which depots must be opened, assigning each customer to one and only one opened depot, and determining vehicle routes to serve customers. Three ant colonies are used following this solution characteristic. The first colony (location selection) is applied to determine the location set. Then the customers are assigned to each selected facility location by a second colony (customer assignment). A VRP for each selected facility location (route construction) is solved by the third colony. These three ACOs are applied iteratively until the stopping criterion is met. Moreover, three different pheromone matrices and pheromone updating rules are used to record pheromone information for each colony, respectively.

Fig. 1 illustrates the pseudo code of the proposed MACO algorithm. The proposed algorithm consists of three inner loops. The main loop (line 6 to 25) undergoes repetition until the number of iterations for the CLRP problem is reached. Internal loop (line 8 to 17) undergoes repetition equal to number of ants (b). The third loop (line 11) which is implicitly undergoes the ant colony optimization for vehicle routing. Every ant deemed to find the best vehicle routing for the selected locations. When the iteration best ant is obtained, the local search is applied with two methods: insertion and move (lines 18 and19). Finally, when all ants generate their solutions, the global pheromone is updated in line 24.

Algorithm 1: MACO for the CLRP

1\. ***Procedure*** MACO()
2\. ***Begin***
3\. *Set_Parameters*;
4\. *Initialize_pheromone_matrix*; /* *Initialize the pheromone matrices* A_{LS}, A_{CA} */
5\. $L_b := \infty$; /* *Set initial global-best cost* (L_b)*/
6\. ***While*** (*termination criterion is not met*)
7\. $L_s := \infty$; /* *iteration-best cost* (L_s)*/
8\. ***for*** $h := 1$ *to* b ***do*** /* *Construct LRP solutions* */
9\. *Location_selection*(T_h);
10\. *Customer_Assignment*(T_h); /* *based on the selected location* */
11\. *VRP*(T_h); /* *based on the customer assignment* */
12\. *Local_Pheromone_Update*(A_{LS}, A_{CA}, T_h); /* *Update local pheromone* */
13\. ***if*** *Cost*(T_h) $< L_s$ ***then*** /* *Record the iteration-best solution* (T_s)*/
14\. $L_s :=$ *Cost*(T_h);
15\. $T_s := T_h$;
16\. ***end if***
17\. ***end for***
18\. *Insertion_Move_for_LRP*(T_s); /* *Apply local search to improve* T_s*/
19\. *Swap_Move_for_LRP*(T_s); /* *Apply local search to improve* T_s*/
20\. ***if*** *Cost*(T_s) $< L_b$ ***then*** /* *Record the global-best solution* (T_b)*/
21\. $L_b :=$ *Cost*(T_s);
22\. $T_b := T_s$;
23\. ***end if***
24\. *Global_Pheromone_Update*(A_{LS}, A_{CA}, T_b, T_s); /* *Update global pheromone*/
25\. ***end While***
26\. ***Return*** T_b, L_b;
27\. ***End***

Fig. 1. Pseudo code of the proposed MACO algorithm.

Location selection in line nine creates an ant with random number of locations. Each ant may generate different number of locations. Customer assignment in line 10 then creates a new assignment based on the selected locations. These two ant colonies will create a one-to-one mapping solution. The construction rules for both ant colonies are explained in Sections 3.2 and 3.3, respectively. VRP in line 11 creates vehicle routes for each selected facility and customer assignment. This procedure is explained later in Section 3.4 after introducing the transition rules for both location selection and customer assignment. The local search procedure is explained in Section 3.5. In addition, both local and global pheromone updates are made clear in Section 3.6. The notations used in MACO are listed in the appendix.

3.2. Location selection

The location selection will affect the customer assignment and the vehicle routing construction significantly in the CLRP. Since we consider the location selection as the main problem for the CLRP, each ant h may have different number of facilities to select in our MACO. The idea here is for ants to construct solutions diversely. The number of locations which will be selected by an ant is calculated by Eq. (1).

$$p_s^h = \left\lfloor \frac{\sum_{i=1}^{n} d_i}{\sum_{j=1}^{m} R_j/m} \right\rfloor + U(1,r) \tag{1}$$

where $p_s^h\ (\leq m)$ is the number of locations of ant h at sth iteration, $\lfloor x \rfloor$ is the largest integer smaller than or equal to x, $U(1,r)$ is a random number following the uniform distribution in [1, r], and r is a pre-specified number. The first term is the number of facilities that should be selected based on the average facility capacity. Then p_s^h locations are successively chosen from m candidate sites according to the selection rule, Eqs. (2) and (3).

$$j=\begin{cases}\arg\max_{j\in O_s^h}\left[\tau_j^s(\eta_j)^\alpha\right], & \text{if } q\le q_0\\ J, & \text{otherwise}\end{cases}\tag{2}$$

$$J:P_j^h=\frac{\tau_j^s\left(\eta_j\right)^\alpha}{\sum_{j\in O_s^h}\tau_j^s\left(\eta_j\right)^\alpha}\tag{3}$$

where O_s^h is the set of candidate sites which are not selected yet by ant h at sth iteration, τ_j^s is the pheromone level of location j at sth iteration, η_j is the ratio of capacity (R_j) to the fixed charge (f_j), α is the parameter that determines the relative influence of τ_j^s versus η_j^s ($\alpha>0$), and q is uniformly distributed within [0, 1]. With a prespecified frequency q_0, the chosen candidate is the location that gives the largest utility function $\tau_j^s(\eta_j^s)^\alpha$ among the set of not yet selected candidate sites, O_s^h ($q\le q_0$). Otherwise, it is taken from O_s^h using the probability distribution induced in Eq. (3).

3.3. Customer assignment

After determining the facility location set, the MACO allocates each customer i to a selected location k ($i\ne k$) based on the customer assignment rule by another ant colony. The location selection and customer assignment is mapped one on one. That is, we only construct one customer assignment solution based on the construction rule in Eqs. (4) and (5) for every facility location solution found in Section 3.2.

$$k=\begin{cases}\arg\max_{k\in W_s^h}\left[\xi_{ik}^s\left(\psi_{ik}^s\right)^\beta\right], & \text{if } q'\le q'_0\\ K, & \text{otherwise}\end{cases}\tag{4}$$

$$K:P_{ik}^h=\frac{\xi_{ik}^s\left(\psi_{ik}^s\right)^\beta}{\sum_{k\in W_s^h}\xi_{ik}^s\left(\psi_{ik}^s\right)^\beta}\tag{5}$$

where W_s^h is the set of selected facility locations of ant h at the location selection colony at sth iteration, ξ_{ik}^s is the pheromone level between customer i and location k at sth iteration, ψ_{ik}^s is the reciprocal of D_{ik}, which is given by

$$D_{ik}=\min_{l\in A_k^{sh}}c_{il}\tag{6}$$

where A_k^{sh} is the set of nodes including location k and those customers that have been assigned to location k of ant h at iteration s (initially only location k itself). The idea here is that the closer a customer to a selected facility, the higher possibility that the customer will be served by the facility. C_{il} is the distance between customers i and l. β is the parameter that determines the relative effect of ξ_{ik}^s versus ψ_{ik}^s ($\beta>0$), q' is a random variable uniformly distributed in [0, 1], and the parameter $q_0{}'$ determines the relative importance of exploitation Eq. (4) versus exploration Eq. (5).

After assigning the customers, the capacity restriction for each selected facility location $F\in W_s^h=\{1, 2, \ldots, p_s^h\}$ will be checked. If the capacity restriction of the location F is violated, the customer assignment for F will be revised by the following procedures.

Step 1: Sort the demand of customer assigned to the location F in descending order. Let $[e]$ be the index of the customer with the eth largest demand. Let $e=1$.
Step 2: Remove the customer $[e]$ from the current location and re-assign this customer to the nearest selected location with spare capacity.
Step 3: If the capacity constraint of the location F is met or no selected location have spare capacity, go to Step 4. Otherwise, $e=e+1$, go to Step 2.
Step 4: If the capacity constraint of the location F is still violated, let $p_s^h=p_s^h+1$ and go to the location selection phase. Otherwise, stop.

```
Algorithm 2: ACO for VRP
1.  Procedure VRP()
2.  Begin
3.    Set_Parameters;
4.    Nearest_Neighbor_heuristic(L_nn);
5.    Initialize_pheromone; /* Initialize the route construction pheromone matrix A_RC*/
6.    L_b' := ∞; /* Set initial global-best cost for VRP (L_b')*/
7.    While (Stopping criterion is not met)
8.      L_s' := ∞; /* Set initial iteration-best cost for VRP (L_s')*/
9.      for h' := 1 to b' do
10.       VRP_Solution_Construction(T_h'); /* Construct VRP solution*/
11.       Local_Pheromone_Update(A_RC, T_h'); /* Local pheromone update */
12.       if Cost(T_h') < L_s' then /* Record the iteration-best solution for VRP (T_h')*/
13.         L_s' := Cost(T_h');
14.         T_s' := T_h';
15.       end if
16.     end for
17.     2-opt_for_VRP(T_s'); /* Apply local search methods to the iteration-best solution*/
18.     Swap_for_VRP(T_s');
19.     Insertion_for_VRP(T_s');
20.     if Cost(T_s') < L_b' then /* Record the global-best solution for VRP (T_b')*/
21.       L_b' := Cost(T_s');
22.       T_b' := T_s';
23.     end if
24.     Global_Pheromone_Update(A_RC, T_b', T_s'); /* Global pheromone update */
25.   end While
26.   Return T_b', L_b';
27. End
```

Fig. 2. Pseudo code of the proposed ACO algorithm for the VRP.

The idea to remove the largest demand customer in step 2 is to repair the infeasible solution into a feasible solution in a quick way. If the capacity constant cannot be fixed, we just add one more location in step 4 and go back to the location selection to generate another ant solution.

3.4. Vehicle routing

After all customers are assigned to one of the selected facility locations, the construction of vehicle routes for each location can be regarded as an independent vehicle routing problem. In this research, we apply an ACO to solve the VRP for each selected facility location and those customers assigned to the location. Fig. 2 illustrates the pseudo code of the proposed ACO for the VRP. The number of ants for VRP is b' in line 9. Detailed procedures of our ACO for the VRP are described in the following sub-sections.

3.4.1. Route construction rule of VRP

In our ACO, when located at node i, ant h' moves to a node v by the following state transition rule.

$$v=\begin{cases}\arg\max_{v\in Z_{is'}^{h'}}\left[\zeta_{iv}^{s'}\left(\phi_{iv}^{s'}\right)^\gamma\right], & \text{if } q''\le q''_0\\ V, & \text{otherwise}\end{cases}\tag{7}$$

$$V:P_{iv}^{h'}=\frac{\zeta_{iv}^{s'}\left(\phi_{iv}^{s'}\right)^\gamma}{\sum_{v\in Z_{is'}^{h'}}\zeta_{iv}^{s'}\left(\phi_{iv}^{s'}\right)^\gamma}\tag{8}$$

where $Z_{is'}^{h'}$ is the set of nodes which are not visited yet by ant h' at node i at iteration s', $\zeta_{iv}^{s'}$ is the pheromone of edge (i, v) at iteration s', $\varphi_{iv}^{s'}$ is defined as the savings of combining two nodes i and v on one tour as opposed to serving them on two different tours at iteration s'. The $\varphi^{s}_{iv}{}'$ is calculated as follows.

$$\varphi^{s'}_{iv} = c_{i0} + c_{0v} - c_{iv} \tag{9}$$

where c_{iv} is the distance between nodes i and v, and the node 0 denotes the selected facility location. γ is the parameter that determines the relative influence of pheromone versus heuristic information ($\gamma > 0$), q'' is a random variable uniformly distributed in [0, 1], and the parameter q_0'' determines the relative importance of exploitation Eq. (7) versus exploration Eq. (8).

The starting node of each ant is randomly generated. Each ant constructs the route based on the state transition rule in Eqs. (7) and (8). When the vehicle capacity is violated, the ant will return to the facility location, and then start from the facility to build a new route. This process continues until each node is visited.

3.4.2. Local search of VRP

In the typical ACO, after the ants have constructed their solutions but before the pheromone is local updated, each ants' solution is improved by applying a local search. However, local search is a time-consuming procedure of ACO. To save the computation time, we will only apply local search to the best solution built in this iteration. The idea here is that the local optima reached from good solutions is expected to be better than those obtained from poor solutions. Moreover, our ACO involves three local search methods: 2-opt, swap and insertion (line 17 to 19 in Fig. 2). In 2-opt, two non-consecutive arcs are removed, either in the same route or in two different routes, and the two paths created are reconnected to restore feasibility. Two customers are exchanged in swap. Insertion is to move one customer from its current position to another position, in the same route or in a different route.

3.4.3. Pheromone updating rule of VRP

The pheromone updating of a typical ACO includes global and local updating rules. The ants apply a local pheromone update rule immediately after they crossed an edge (i, v) during the tour construction. The local pheromone updating rule of our ACO is

$$\zeta^{s'+1}_{iv} = (1-\rho'')\zeta^{s'}_{iv} + \rho''\zeta_0 \quad \text{if } \{\text{edge } (i,v) \in T_{h'}\} \tag{10}$$

where $T_{h'}$ denotes the routes constructed by ant h', ρ'' is the pheromone evaporation parameter in the range of [0, 1] that regulates the reduction of pheromone on the edges. The ζ_0 is the initial value of the pheromone matrix for the route construction rule, and is set to be $1/n_j L_{nn}$ (line 5 in Fig. 2), where n_j is the number of nodes in the VRP for facility location j and L_{nn} is the length of routes constructed by the Nearest Neighborhood heuristic.

Dorigo et al. (1996) and Bullnheimer et al. (1999) used the elitist strategy on the trail updating in ant system. Such a strategy will direct the search of all the other ants in probability toward a solution composed by some edges of the best tour itself. In our ACO, the best elitist tours, including the global-best tour (T'_b) and the iteration-best tour ($T'_{s'}$) of VRP, are allowed to lay pheromone on the edges that belong to them. The idea here is to balance between exploitation (through emphasizing the global-best tour) as well as exploration (through the emphasis to the iteration-best tour). The global updating rule of ACO for VRP is described as follow.

$$\zeta^{s'+1}_{iv} = (1-\rho'')\zeta^{s'}_{iv} + \rho''\Delta\zeta^{s'}_{iv} \tag{11}$$

where

$$\Delta\zeta^{s'}_{iv} = \begin{cases} [(L'_w - L'_b) + (L'_w - L'_{s'})]/L'_w & \text{if } \{(i,v) \in T'_b \text{ or } T'_{s'}\} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where L'_b and $L'_{s'}$ denote the tour length of the global-best solution and the iteration-best solution of VRP, respectively, and L'_w is the tour length of the worst solution of the current iteration. Edges that do not belong to the global-best solution and the iteration-best solution just loose pheromone at the rate ρ'', which constitutes the trail evaporation. This choice is intended to make ants searching in the neighborhood of the two best tours instead of the globally best tour to avoid the algorithm being trapped in a local optimum without finding very good solutions.

3.5. Local search of MACO

After the initial solutions of the MACO are generated by the solution construction rules, two local search approaches are applied to improve the solutions. The first one is insertion move between two opened facility locations. A customer is removed from one route of one location, and then it is inserted into one route of another location. The other one is swap move between two opened facility locations. Two customers belonging to two different locations are exchanged. For these local search approaches, the best improvement strategy is adopted. All possible moves are evaluated and then the move with largest improvements is selected. We chose insertion and swap over other heuristics available for the local search based on their trade-off between implementation simplicity, accuracy, and computational performance.

3.6. Pheromone updating rules of MACO

Every time an ant constructs a solution, the quantity of pheromone associated with the node for the location selection and the customer assignment is decreased, and the location and assignment becomes less attractive. On the other hand, global updating is used to intensify the search in the neighborhood of the best solution computed. In our MACO, the pheromone updating rules of location selection and customer assignment are different. The respective local pheromone updating rules for both selection and assignment pheromone matrices used in our MACO are as follows.

$$\tau^{s+1}_j = (1-\rho)\tau^s_j + \rho\tau^0_j \quad \text{if } \{\text{location } j \in T_h\} \tag{13}$$

$$\xi^{s+1}_{ij} = (1-\rho')\xi^s_{ij} + \rho'\xi_0 \quad \text{if } \{\text{edge } (i,j) \in T_h\} \tag{14}$$

where T_h is the solution constructed by ant h, $0 \le \rho$, $\rho' \le 1$ are the pheromone decay parameters, and τ^0_j $(=1/f_j)$ and ξ_0 (a very small value) are initial level of pheromone matrices, respectively.

Since the location selection and customer assignment is one-to-one mapping, these two ant colonies communicate with each other through the global pheromone updating. Both location and assignment colonies use independent pheromone trails but collaborate by sharing the variable global-best solution (T_b) and the iteration-best solution (T_s) of CLRP. The global pheromone updating for location selection will take into account the number of customers assigned to each location. The respective global pheromone updating rules for both selection and assignment pheromone matrices can be described as follows.

$$\tau^{s+1}_j = (1-\rho)\tau^s_j + \rho\Delta\tau^s_j \tag{15}$$

$$\xi^{s+1}_{ij} = (1-\rho')\xi^s_{ij} + \rho'\Delta\xi^s_{ij} \tag{16}$$

where

$$\Delta\tau_j^s = \begin{cases} [(L_w - L_b) + (L_w - L_s)] \times n_j / L_w & \text{if } \{j \in T_b \text{ or } T_s\} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

$$\Delta\xi_{ij}^s = \begin{cases} [(L_w - L_b) + (L_w - L_s)] / L_w & \text{if } \{j \in T_b \text{ or } T_s \text{ and } i \text{ is assigned to } j\} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where L_b and L_s denote the total cost of the global-best solution and the iteration-best solution of the LRP, respectively. L_w is the total cost of the worst solution at current iteration, and n_j represents the number of customers assigned to location j.

4. Computational results

In this section, we present the results of our computational experiments with the multiple ant colony optimization algorithm described in previous section. The proposed MACO algorithm is coded in Borland C++ Builder 5.0 and runs on a PC with an Athlon XP 2500+(1.83 Ghz) processor and 512 MB RAM, under the Windows XP operating system. We performed a set of preliminary experiments in order to find appropriate parameter settings that produce overall good results across most instances, even if they were not the optimal settings for all instances. We consider a set of parameters for each phase and then modifying one, while keeping the others fixed. The parameter tested include: $r \in \{1, 2, 3, 4, 5\}$, α, β, $\gamma \in \{1, 2, 3, 4, 5\}$, ρ, ρ', $\rho'' \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, q_0, q'_0, $q''_0 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $b \in \{4, 10, 15, 20, 25\}$, $b' \in \{n_j/10, n_j/5, n_j/3, n_j/2, n_j\}$($n_j$ is the number of customers assigned to depot j), $S \in \{15, 25, 50, 75, 100\}$ (maximum number of iterations of MACO) and $S' \in \{n/10, n/5, n/3, n/2, n\}$(maximum number of iterations of VRP). Among all the preliminary experiments we found that the parameter setting: $b=4$, $b' = \lceil n_j/5 \rceil$, $r=3$, $\alpha=1$, $\beta=1$, $\gamma=4$, $\rho=0.1$, $\rho'=0.1$, $\rho''=0.1$, $q_0=0.5$, $q_0'=0.1$, $q_0''=0.5$, $S=25$ and $S' = \lceil n/5 \rceil$, provided the best results. We then test our MACO by four different groups of benchmark instances from the literature, containing a total of 94 instances, with the same parameter setting. The four sets of benchmark instances used were designed by Perl (1983), Barreto (2004), Prins et al. (2004) and Tuzun and Burke (1999). For each instance, the algorithm is implemented for ten runs. The reported result is the best found solution over the runs with the average CPU time in seconds over the 10 runs. The best known solution in each instance was obtained based on the best results among comparing algorithms in their published version or during their parameter analyses.

4.1. Perl's instances

The first set of instances, which includes nine instances, was obtained from Perl (1983). The number of candidate sites ranges from 2 to 15 while the numbers of customers range from 12 to 85. The characteristics of Perl's instances and computational results obtained by MACO are summarized in Table 1. Columns 2–5 in Table 1 show the number of customers n, the number of candidate sites m, vehicle capacity Q, the best-known solutions (BKS). The best known solution corresponds to the best known solution as reported by previous authors in the literature. Columns 6–8 in Table 1 present the detail information of the best solution over 10 runs for each instance obtained by MACO, including the best solution found in these 10 runs C, the running time in seconds over the 10 runs (CPU), and the relative percentage deviation over the best known solution (Gap), computed as $100\% \times (C - \text{BKS})/\text{BKS}$. The set of customers and candidate sites in instances P3–P9 are the same, but the variable costs and the fixed costs of these instances are different. The computational results show that our MACO is very effective to solve the Perl's instances. The MACO reaches 1 best-known solution and finds eight new best solutions. The largest improvement is by 4.05%. For all instances, the overall average gap over best known solution is −1.37%.

The performance of MACO is compared with other methods published in the literature, including Heuristic-P in Perl (1983),

Table 1
Computational results for Perl's instances.

Instance	n	M	Q	BKS	C	CPU	Gap
P1	12	2	140	355.58	**355.58**[a]	2.171	**0.00**
P2	55	15	120	5,532.28	**5,507.25**	19.516	**−0.45**
P3	85	7	160	7,551.61	**7,466.17**	46.656	**−1.13**
P4	85	7	160	8,374.73	**8,249.97**	52.843	**−1.49**
P5	85	7	240	5,775.08	**5,648.54**	46.344	**−2.19**
P6	85	7	160	8,223.72	**8,142.64**	64.594	**−0.99**
P7	85	7	160	7,257.72	**7,176.64**	64.578	**−1.12**
P8	85	7	160	7,781.47	**7,466.17**	49.438	**−4.05**
P9	85	7	160	10,141.60	**10,049.56**	36.922	**−0.91**
Avg.				6,777.09	6673.61	42.56	**−1.37**

[a] The solution is better than or equal to the best known solution is in boldface.

Table 2
Comparison of results for Perl's instances.

Instance	Heuristic-P	Heuristic-H	SA	TS-ACO	MACO
P1	**0.00**[a]	**0.00**	**0.00**	**0.00**	**0.00**
P2	4.76	1.54	0.00	3.61	**−0.45**
P3	3.16	0.00	3.04	1.16	**−1.13**
P4	7.65	0.00	–[b]	–	**−1.49**
P5	2.51	0.00	–	–	**−2.19**
P6	1.79	0.00	–	–	**−0.99**
P7	2.03	0.00	–	–	**−1.12**
P8	4.94	0.00	–	–	**−4.05**
P9	4.51	0.00	–	–	**−0.91**
Avg. gap (%)	3.48	0.17	1.01	1.59	**−1.37**
Avg. time[c] (sec)	136.26	147.52	4.98	4.12	42.56

[a] The best among all compared algorithms is in boldface.
[b] The instance was not tested.
[c] The average CPU time in seconds on the computer used by each algorithm.

Table 3
Computational results for Barreto's instances.

Instance	n	m	Q	BKS	C	CPU	Gap
B1	21	5	6,000	424.90*	**424.90**[a]	5.64	**0.00**
B2	22	5	4,500	585.10*	**585.10**	4.69	**0.00**
B3	29	5	4,500	512.10*	**512.10**	8.59	**0.00**
B4	32	5	8,000	562.22*	**562.22**	13.05	**0.00**
B5	32	5	11,000	504.30*	**504.30**	9.56	**0.00**
B6	36	5	250	460.40*	**460.40**	13.14	**0.00**
B7	50	5	160	565.60*	**565.60**	29.16	**0.00**
B8	75	10	140	844.40	844.88	58.72	0.06
B9	100	10	200	833.40*	836.75	83.92	0.40
B10	12	2	140	204.00	**204.00**	2.09	**0.00**
B11	55	15	120	1,112.10	1,112.58	28.81	0.04
B12	85	7	160	1,622.50	1,623.14	77.86	0.04
B13	318	4	25,000	557,275.2	560,210.81	831.34	0.53
B14	318	4	8,000	673,297.70	670,118.50	1986.33	**−0.47**
B15	27	5	2,500	3,062.00*	**3,062.00**	8.59	**0.00**
B16	134	8	850	5709.00	**5709.00**	136.63	**0.00**
B17	88	8	9000,000	355.80	**355.80**	99.53	**0.00**
B18	150	10	8000,000	43,919.90	44,131.02	166.95	0.48
B19	117	14	150	12,290.30	12,355.91	77.45	0.53
Avg.				68,639.00	68,641.00	191.69	0.08

* Proved optimal solution.
[a] The solution is better than or equal to the best known solution is in boldface.

Table 4
Comparison of results for Barreto's instances.

Instance	CH	SA-ACS	GRASP	MA|PM	LRGTS	HybPSO	GRASP × ELS	SALRP	MACO
B1	2.59	1.29	1.11	**0.00**[a]	**0.00**	1.88	**0.00**	**0.00**	**0.00**
B2	1.09	0.27	**0.00**	4.56	0.39	0.58	**0.00**	**0.00**	**0.00**
B3	**0.00**	**0.00**	0.59	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
B4	1.69	1.26	1.73	1.73	4.48	1.53	**0.00**	**0.00**	**0.00**
B5	1.41	0.36	**0.00**	6.03	0.10	1.35	**0.00**	**0.00**	**0.00**
B6	2.24	2.17	**0.00**	5.43	3.50	2.24	**0.00**	**0.00**	**0.00**
B7	3.02	–[b]	5.92	**0.00**	3.68	3.02	**0.00**	**0.00**	**0.00**
B8	4.96	–	2.04	2.57	2.26	4.96	0.76	0.43	**0.06**
B9	6.72	–	3.38	2.00	1.14	6.72	**0.00**	0.59	0.40
B10	**0.00**	**0.00**	–	–	–	**0.00**	–	**0.00**	**0.00**
B11	2.17	0.57	–	–	–	2.14	–	0.06	**0.04**
B12	2.12	1.78	–	–	–	2.12	–	**0.00**	0.04
B13	4.20	–	–	–	–	4.20	–	1.12	**0.53**
B14	11.04	–	–	–	–	11.04	–	1.61	**−0.47**
B15	**0.00**	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**	**0.00**	**0.00**
B16	9.27	8.75	4.49	4.22	1.75	9.13	0.18	**0.00**	**0.00**
B17	8.18	–	0.31	**0.00**	3.63	8.18	**0.00**	**0.00**	**0.00**
B18	6.20	–	1.61	0.21	1.06	6.20	**0.10**	2.71	0.48
B19	1.50	–	–	–	–	1.50	–	1.17	**0.53**
Avg. gap (%)	3.60	1.50	1.63	2.06	1.70	3.51	0.08	0.40	**0.08**
Avg. time[c] **(sec)**	–	–	20.28	35.62	17.58	48.48	187.62	464.07	191.69

[a] The best among all compared algorithms is in boldface.
[b] The instance was not tested.
[c] The average CPU time in seconds on the computer used by each algorithm.

Heuristic-H in Hansen et al. (1994), SA in Wu et al. (2002) and TS-ACS in Wang et al. (2005). Table 2 presents the comparison of results for Perl's instances obtained by MACO and other methods. All the percentage deviation values are computed from the original publications; a dash ('-') indicates that the instance is not tested by the algorithm. The last row provides the average computation time in seconds. The best value on each instance is indicated in boldface. The results indicate that our MACO outperforms all compared methods on average for the gap and number of best solutions found. On the other hand, the larger computational effort of our MACO is due to the multi-start of initial solution of MACO. However, its computational time is reasonable for such a problem.

4.2. Barreto's instances

The second set of 19 instances was collected from Barreto (2004) and the complete data sets are available at http://sweet.ua.pt/~iscf143. Different from the first group of instances, the vehicle variable cost is not considered in the second group of instances. These instances are either from the literature or are obtained by adding both capacitated and uncapacitated depots to classical VRP instances. The number of candidate sites ranges from 2 to 15 while the numbers of customers range from 12 to 318. Table 3 presents the basic characteristics of Barreto's instances and the computational results of MACO. The format of the table is the same as that in Table 1. An asterisk on the best known solution indicates a proven optimal solution value. As can be seen from Table 3, the MACO obtains 11 best known solutions and updates 1 best solution out of 19 instances. Our solutions are 0.08% above the best known solution on average while the largest gap is 0.53%. The computational results show that our MACO can solve the CLRP effectively. This is because the diversely sets of facility locations are adopted in the MACO. It is useful to find a better set of facility locations, when the solutions are very sensitive to the choice of the facility locations.

The comparison of algorithms for Barreto's instances is presented in Table 4. The compared algorithms are the clustering based heuristic (CH) in Barreto et al. (2007), the SA-ACS in Bouhafs et al. (2006), the GRASP in Prins et al. (2006a), the MA|PM in Prins et al. (2006b), the LRGTS in Prins et al. (2007), the HybPSO in Marinakis and Marinaki (2008), the SALRP in Yu et al. (2010), and the GRASP × ELS in Duhamel et al. (2010). Among the existing heuristics the LRGTS of Prins et al. (2007), the GRASP × ELS in Duhamel et al. (2010), and the SALRP of Yu et al. (2010) have obtained the best results in the literature. Note that some of the instances are not tested by all these methods, direct comparisons may be in many cases biased. For those 13 instances that most algorithms tested, our MACO and GRASP × ELS are the best two approaches and provide similar solution quality. Four algorithms (CH, HybPSO, SALRP, and MACO) test all 19 instances. The results show that our MACO approach outperforms the other three algorithms and provides the lowest largest gap of 0.53%. In addition, the MACO also yields the best solutions among these four algorithms in 18 out of 19 problems except instance B12.

4.3. Prins et al.'s instances

The third set of 30 instances was generated by Prins et al. (2004) which contains instances with capacitated routes and depots. The complete data sets are available at http://prodhonc.free.fr/. Different from the first two groups of instances, the vehicle fixed cost is considered in this group of instances. The number of candidate sites ranges from 5 to 10 whereas the number of customers ranges from 20 to 200. The vehicle capacity is either 70 or 150. Table 5 presents the information of Prins et al.'s instances and the computational results of the MACO. The format of the table is the same as that in Table 1. An asterisk on the best known solution indicates a proven optimal solution value. From Table 5 we can observe that the MACO reaches all 12 best-known solutions for the small size instances. The average gap of all instances is 0.36%, while the largest gap is 1.55%. The computational results show that our algorithm can effectively solve the large size LRP instances involving as many as 10 candidate facilities and 200 customers.

Moreover, the comparison of algorithms for Prins et al.'s instances is presented in Table 6. The compared algorithms are the MSLS in Prins et al. (2004), the GRASP in Prins et al. (2006a), the MA|PM in Prins et al. (2006b), the LRGTS in Prins et al. (2007), the SALRP in Yu et al. (2010), and the GRASP × ELS in Duhamel

Table 5
Computational results for Prins et al.'s instances.

Instance	n	m	Q	BKS	C	CPU	Gap
Pr1	20	5	70	54,793*	**54,793**[a]	4.08	**0.00**
Pr2	20	5	150	39,104*	**39,104**	4.78	**0.00**
Pr3	20	5	70	48,908*	**48,908**	3.92	**0.00**
Pr4	20	5	150	37,542*	**37,542**	5.45	**0.00**
Pr5	50	5	70	90,111*	**90,111**	24.47	**0.00**
Pr6	50	5	150	63,242*	**63,242**	21.31	**0.00**
Pr7	50	5	70	88,298*	**88,298**	24.19	**0.00**
Pr8	50	5	150	67,308*	**67,308**	20.13	**0.00**
Pr9	50	5	70	84,055*	**84,055**	24.66	**0.00**
Pr10	50	5	150	51,822*	**51,822**	16.83	**0.00**
Pr11	50	5	70	86,203*	**86,203**	32.78	**0.00**
Pr12	50	5	150	61,830*	**61,830**	25.70	**0.00**
Pr13	100	5	70	274,814*	276,220	116.59	0.51
Pr14	100	5	150	213,615	214,323	134.52	0.33
Pr15	100	5	70	193,671*	194,441	237.28	0.40
Pr16	100	5	150	157,095*	157,222	144.33	0.08
Pr17	100	5	70	200,079*	201,038	178.75	0.48
Pr18	100	5	150	152,441*	152,722	151.58	0.18
Pr19	100	10	70	287,983	291,134	105.28	1.09
Pr20	100	10	150	231,763	235,348	81.81	1.55
Pr21	100	10	70	243,590*	245,263	122.47	0.69
Pr22	100	10	150	203,988*	205,524	85.03	0.75
Pr23	100	10	70	250,882	254,302	112.09	1.36
Pr24	100	10	150	204,317	204,786	79.70	0.23
Pr25	200	10	70	477,248	478,843	941.78	0.33
Pr26	200	10	150	378,351	378,865	562.05	0.14
Pr27	200	10	70	449,571	451,457	703.66	0.42
Pr28	200	10	150	374,330	374,972	403.53	0.17
Pr29	200	10	70	469,433	475,155	878.83	1.22
Pr30	200	10	150	362,817	365,401	490.92	0.71
Avg.				196,640.13	197,674.40	191.28	0.36

* Proved optimal solution.
[a] The solution is better than or equal to the best known solution is in boldface.

Table 6
Comparison of results for Prins et al.'s instances.

Instance	MSLS	GRASP	MA|PM	LRGTS	GRASP × ELS	SALRP	MACO
Pr1	1.85	0.42	**0.00**[a]	0.62	**0.00**	**0.00**	**0.00**
Pr2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Pr3	1.55	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Pr4	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Pr5	8.84	0.58	0.05	0.05	**0.00**	**0.00**	**0.00**
Pr6	14.10	2.40	**0.00**	0.02	**0.00**	**0.00**	**0.00**
Pr7	2.14	0.55	**0.00**	0.47	0.39	**0.00**	**0.00**
Pr8	2.03	1.09	0.87	0.58	**0.00**	0.05	**0.00**
Pr9	–[b]	**0.00**	**0.00**	0.15	**0.00**	**0.00**	**0.00**
Pr10	–	0.46	**0.00**	0.33	**0.00**	**0.00**	**0.00**
Pr11	12.24	1.37	**0.00**	**0.00**	**0.00**	0.29	**0.00**
Pr12	0.02	0.10	**0.00**	**0.00**	**0.00**	1.41	**0.00**
Avg. gap	4.28	0.58	0.08	0.19	0.03	0.15	**0.00**
Pr13	3.57	1.68	2.59	1.14	0.78	0.81	**0.51**
Pr14	2.08	1.19	1.42	0.59	1.05	1.12	**0.33**
Pr15	2.26	3.02	0.98	1.48	0.31	**0.23**	0.40
Pr16	1.73	1.56	0.15	0.44	0.18	**0.04**	0.08
Pr17	2.41	1.96	0.83	0.94	0.13	**0.08**	0.48
Pr18	2.25	1.41	0.58	1.49	0.06	**0.02**	0.18
Avg. gap	2.38	1.80	1.09	1.01	0.42	0.38	**0.33**
Pr19	12.31	12.22	9.93	1.36	4.67	**1.06**	1.09
Pr20	19.81	17.14	16.61	1.63	16.32	**1.06**	1.55
Pr21	19.95	4.31	0.63	1.28	**0.08**	0.91	0.69
Pr22	20.65	1.26	0.52	0.22	**0.00**	0.65	0.75
Pr23	2.81	7.95	1.11	3.10	1.05	**0.00**	1.36
Pr24	2.46	5.80	0.24	0.77	0.38	0.34	**0.23**
Avg. gap	13.00	8.11	4.84	1.39	3.75	**0.67**	0.95
Pr25	20.72	2.84	1.31	0.93	1.93	0.79	**0.33**
Pr26	6.28	10.15	0.45	0.60	1.05	1.38	**0.14**
Pr27	22.76	14.04	0.50	0.84	0.45	**0.28**	0.42
Pr28	3.91	1.51	0.18	0.81	0.22	0.63	**0.17**
Pr29	18.23	5.81	1.85	1.54	1.91	**0.95**	1.22
Pr30	23.83	7.22	0.56	0.67	0.65	**0.24**	0.71
Avg. gap	15.96	6.93	0.81	0.90	1.10	0.71	**0.50**
Avg. gap (%)	8.24	3.60	1.38	0.69	1.07	0.41	**0.36**
Avg. time[c] (sec)	176.57	96.49	76.69	17.48	258.17	422.36	191.28

[a] The best among all compared algorithms is in boldface.
[b] The instance was not tested.
[c] The average CPU time in seconds on the computer used by each algorithm.

et al. (2010). Among those comparing algorithms, SALRP is on average the most effective on this set of benchmark instances from the literature. As shown in Table 6, our MACO obtains slightly better results than those by SALRP, and it also needs less computational time. However, since the algorithms were run on different machines and different compliers were used, it is difficult to compare the run time directly. For all instances, the gaps over best known solutions obtained by the MACO range from 0.0% to 1.55%. This indicates that our MACO is robust to find a good solution. It can be observed that our MACO outperforms all the other algorithms on average in small size ($n \leq 50$) and the largest size instances ($n = 200$). The MACO takes longer computational time than the compared algorithms but the running time is quite acceptable for a strategic problem like the CLRP, even on the largest instances (less than 16 min).

4.4. Tuzun and Burke's instances

The fourth set of instances was first described in Tuzun and Burke (1999). The instances can also be found at http://prodhonc.free.fr/. Different from other three sets of instances, the Tuzun and Burke's instances do not possess capacity restrictions for facilities. Therefore, we set the facility capacity as the total customer demand in this set of instances. Table 7 provides characteristics of Tuzun and Burke's instances and computational results of the MACO. An asterisk in the best known solution indicates that the solution was proven to be optimal. The average gap of all instances obtained by the MACO is 0.90%, while the gap ranges between −0.15% and 4.21%. Our MACO reaches 4 best known solutions and updates 3 best solutions. Note that our MACO was designed to solve CLRP based on the facility capacity information to select the location. Since this set of instances does not have facility capacity limitation, the results are not as good as the other three sets of instances as we expected.

In Table 8, we present information about the comparison of our MACO with other algorithms. The compared algorithms are the TS in Tuzun and Burke (1999), the GRASP in Prins et al. (2006a), the MA|PM in Prins et al. (2006b), and the LRGTS in Prins et al. (2007), the SALRP in Yu et al. (2010), and the GRASP × ELS in Duhamel et al. (2010). GRASP × ELS is on average the most effective on this set of benchmark instances from the literature. On average, our MACO obtains very competitive average gap that is slighter better than that of the GRASP × ELS. Moreover, it is noticeable that our MACO provides better results in the medium and large size instances ($n = 100, 200$) than those by GRASP × ELS, and the largest gap in these instances is much smaller than that of the comparing algorithms. In addition, our MACO is the only algorithm that updates best solutions in three instances. With respect to computational times the MACO takes longer than the comparative algorithms but it finds good solutions within reasonable times.

Overall, we solved four groups of CLRP instances to test the performance of the MACO. The computational time grows reasonably with the problem size. For all 94 tested instances, our algorithm was

Table 7
Computational results for Tuzun and Burke's instances.

Instance	n	m	Q	BKS	C	CPU	Gap
T1	100	10	150	1467.68*	1489.68	71.08	1.50
T2	100	20	150	1449.20	1453.89	46.33	0.32
T3	100	10	150	1394.80*	1407.78	60.98	0.93
T4	100	20	150	1432.29	1433.42	54.05	0.08
T5	100	10	150	1167.16*	1208.04	79.64	3.50
T6	100	20	150	1102.24	**1102.24**[a]	64.53	**0.00**
T7	100	10	150	791.66*	792.90	95.38	0.16
T8	100	20	150	728.30	**728.30**	65.14	**0.00**
T9	100	10	150	1238.24	1265.27	77.17	2.18
T10	100	20	150	1245.31	1256.95	50.03	0.93
T11	100	10	150	902.26*	**902.26**	60.56	**0.00**
T12	100	20	150	1018.29	**1018.29**	68.56	**0.00**
T13	150	10	150	1866.75	1945.43	226.63	4.21
T14	150	20	150	1833.95	1853.22	100.88	1.05
T15	150	10	150	1965.12	1991.44	200.81	1.34
T16	150	20	150	1801.39	1812.34	140.59	0.61
T17	150	10	150	1443.33*	1499.05	206.06	3.86
T18	150	20	150	1441.98	1446.63	162.73	0.32
T19	150	10	150	1205.09	**1204.76**	218.19	−**0.03**
T20	150	20	150	930.99	931.73	149.86	0.08
T21	150	10	150	1699.92	1724.02	226.11	1.42
T22	150	20	150	1400.01	1401.05	122.86	0.07
T23	150	10	150	1199.51	1217.29	240.59	1.48
T24	150	20	150	1152.18	1158.03	129.94	0.51
T25	200	10	150	2259.87	2304.67	460.73	1.98
T26	200	20	150	2185.41	2187.65	230.83	0.10
T27	200	10	150	2234.78	**2231.46**	427.89	−**0.15**
T28	200	20	150	2241.04	2275.70	233.89	1.55
T29	200	10	150	2089.77	2098.56	569.08	0.42
T30	200	20	150	1709.56	1711.25	276.17	0.10
T31	200	10	150	1466.62	1472.93	544.30	0.43
T32	200	20	150	1084.78	1087.57	316.20	0.26
T33	200	10	150	1970.44	1978.74	386.44	0.42
T34	200	20	150	1918.93	1959.71	229.63	2.13
T35	200	10	150	1771.06	1782.94	405.49	0.67
T36	200	20	150	1393.16	**1392.70**	268.45	−**0.03**
Avg.				1505.64	1520.22	201.88	0.90

[a] The solution is better than or equal to the best known solution is in boldface.

Table 8
Comparison of results for Tuzun and Burke's instances.

Instance	TS	GRASP	MA|PM	LRGTS	GRASP × ELS	SALRP	MACO
T1	6.06	3.92	1.79	1.58	**0.39**[a]	0.65	1.50
T2	5.71	5.36	1.53	1.56	**0.00**	1.50	0.32
T3	3.49	2.06	1.72	1.24	**0.13**	0.99	0.93
T4	5.52	3.49	4.20	0.75	**0.00**	**0.00**	0.08
T5	5.48	2.83	0.52	1.75	**0.00**	0.86	3.50
T6	2.70	1.94	1.19	1.24	**0.00**	0.74	**0.00**
T7	4.23	2.82	0.29	2.73	0.05	**0.00**	0.16
T8	1.68	2.68	0.30	2.01	**0.00**	0.50	**0.00**
T9	6.36	2.82	1.94	2.40	0.17	**0.02**	2.18
T10	2.34	2.22	0.48	0.87	**0.06**	0.16	0.93
T11	2.05	1.10	0.17	1.20	**0.00**	**0.00**	**0.00**
T12	2.35	0.41	0.46	0.71	**0.00**	0.56	**0.00**
Avg. gap	4.00	2.64	1.22	1.50	**0.07**	0.50	0.80
T13	7.19	7.50	4.96	4.25	**4.17**	4.67	4.21
T14	3.21	3.00	2.60	2.28	1.65	3.55	**1.05**
T15	2.90	3.50	**0.97**	2.31	1.39	4.70	1.34
T16	2.97	3.04	2.99	1.03	1.88	**0.00**	0.61
T17	7.79	4.50	**0.34**	0.37	0.72	0.69	3.86
T18	2.55	1.03	1.24	3.53	**0.15**	0.94	0.32
T19	2.18	2.93	0.19	0.50	1.23	0.10	−**0.03**
T20	1.86	1.05	0.41	0.64	1.59	0.39	**0.08**
T21	3.68	2.18	1.20	1.73	**0.72**	1.23	1.42
T22	6.31	1.84	2.09	1.76	0.21	1.13	**0.07**
T23	5.43	2.02	**0.33**	1.40	1.28	1.44	1.48
T24	2.61	6.87	0.55	0.87	**0.33**	0.60	0.51
Avg. gap	4.06	3.29	1.49	1.72	1.28	1.62	**1.24**
T25	5.29	5.49	**1.51**	1.62	1.59	2.84	1.98
T26	1.20	4.70	4.21	1.01	0.83	3.33	**0.10**
T27	2.39	1.72	1.78	1.17	0.52	1.14	−**0.15**
T28	5.12	4.64	6.03	**0.82**	1.09	3.81	1.55
T29	3.29	2.26	0.79	1.48	0.80	1.09	**0.42**
T30	4.53	5.72	3.62	1.65	4.06	0.79	**0.10**
T31	5.67	2.05	**0.06**	1.50	0.52	0.17	0.43
T32	2.60	1.03	0.30	0.54	**0.08**	0.36	0.26
T33	4.35	3.77	**0.14**	0.69	1.72	1.20	0.42
T34	4.35	8.96	3.13	3.52	2.37	**0.68**	2.13
T35	6.00	1.00	0.63	0.89	**0.44**	0.45	0.67
T36	1.56	1.11	0.22	0.57	4.35	0.23	−**0.03**
Avg. gap	3.86	3.54	1.87	1.29	1.53	1.34	**0.66**
Avg. gap (%)	3.97	3.15	1.53	1.50	0.96	1.15	**0.90**
Avg. Time[b] **(sec)**	11.50	195.60	203.10	21.20	606.64	826.42	201.88

[a] The best among all compared algorithms is in boldface.
[b] The average CPU time in seconds on the computer used by each algorithm.

able to reach 28 best known solutions and obtain 12 new best solutions. The computational results show that our MACO algorithm can solve diverse instances effectively. Moreover, the performance of our algorithm is also compared with other algorithms in the literature. For those six algorithms that test the last three sets of benchmark instances in Tables 4, 6, and 8, our MACO is the only one that is able to update the best known solutions. The best two algorithms that tested these three sets of instances from the literature are the GRASP × ELS by Duhamel et al. (2010) and SALRP by Yu et al. (2010). The results show that our MACO approach is comparable to these two algorithms. It is worth observing that our MACO provides the lowest average gap among all algorithms for the last three sets of 85 instances. Furthermore, the average gap by MACO in the largest size instances with number of customers more than 200 ($n \geq 200$) is much lower than the other algorithms. This indicates that our MACO can find good solutions in the problems close to the real and practical situation.

5. Conclusions

In this paper we proposed a multiple ant colony optimization (MACO) heuristic, which adopts a nested mechanism with three hierarchical solution construction rules, to solve the capacitated location routing problem (CLRP). The CLRP is to solve each of the three decisions in LRP: location selection, customer assignment and route construction. We decompose the CLRP hierarchically into facility location problem (FLP) and multiple depot vehicle routing problem (MDVRP), while the first one is the main problem and the latter problem to be a subordinate one. Thus, the MDVRP is solved embedded in the facility location problem. In each iteration, two ant colonies (location selection and customer assignment) communicate with each other through the global pheromone updating rule.

The computational experiments are carried out on four sets of instances from the literature. Our MACO is able to obtain optimal or near-optimal solutions in a reasonable computation time for most benchmark instances. Our MACO reaches 28 and updates 12 best known solutions, respectively, in 94 instances considered in this study. The results show that the MACO can obtain good solutions on various kinds of CLRP instances within reasonable computational times. The MACO is especially very effective to solve the largest size LRP instances ($n \geq 200$) and the overall performance is competitive with other algorithms in the literature.

Our MACO checks the facility capacity constraint after finishing the customer assignment, in the future we could check this during the customer assignment construction to reduce the computational time. Currently, each colony has its own control parameters in the MACO approach. It would be interesting to study the effect of using the same control parameters in each colony in the future. Another research direction could be the study on reliable location routing problem, in which some facilities are subject to failure. This could be happened in the disaster relief network. Other possible perspective could be applying the MACO to other combinatorial optimization problems that contain multiple-level decisions.

Acknowledgements

This research was supported by a grant (NSC-100-2221-E-155-022) from the National Science Council of Taiwan. The authors would also like to thank two anonymous referees for their helpful comments and suggestions on improving the presentation of this paper.

Appendix

The following notations are used for the MACO algorithm.

- A_k^{sh} the set of nodes including location k and those customers that have been assigned to location k of ant h at sth iteration
- b, b' the number of ants for the MACO and VRP, respectively
- c_{il} the distance between customers i and l
- C the vehicle acquisition cost
- D_{ik} the minimum distance between customer i and all customers served by location k
- f_j fixed charge of location j
- L_b, L_s the total cost of the global-best solution and the iteration-best solution of the LRP, respectively
- L'_b, $L'_{s'}$ the tour length of the global-best solution and the iteration-best solution of VRP, respectively
- L_w the total cost of the worst solution at current iteration
- L'_w the tour length of the worst solution at current iteration
- n_j the number of nodes in the VRP for facility location j
- O_s^h the set of candidate sites which are not selected yet by ant h at sth iteration
- p_s^h the number of locations of ant h at sth iteration
- q, q', q'' random variables uniformly distributed in [0, 1]
- Q the vehicle capacity
- R_j the capacity of location j
- T'_b, $T'_{s'}$ the global-best tour and the iteration-best tour of VRP, respectively
- T_h the facility location solution and customer assignment constructed by ant h
- $T_{h'}$ the VRP solution constructed by ant h'
- $U(1,r)$ a random number following the uniform distribution in $[1, r]$
- W_s^h the set of selected facility locations of ant h at the location selection colony at sth iteration
- $Z_{is'}^{h'}$ the set of nodes which are not visited yet by ant h' at node i at iteration s'
- α the parameter that determines the relative influence of τ_j^s versus η_j^s
- β the parameter that determines the relative effect of ξ_{ik}^s versus ψ_{ik}^s
- γ the parameter that determines the relative influence of $\zeta_{iv}^{s'}$ versus $\varphi_{iv}^{s'}$
- η_j the ratio of capacity and the fixed charge for location j
- ρ, ρ', ρ'' the pheromone evaporation parameter
- $\varphi_{iv}^{s'}$ the savings of combining two nodes i and v into one tour
- ψ_{ik}^s the reciprocal of D_{ik}
- τ_j^s the pheromone level of location j at sth iteration, τ_j^0 is the initial level
- ξ_{ik}^s the pheromone level between customer i and location k at sth iteration, ξ_0 is the initial level
- $\zeta_{iv}^{s'}$ the pheromone of edge (i, v) at iteration s', ζ_0 is the initial level

References

Akca, Z., Berger, R.T., Ralphs, T.K., 2009. A branch-and-price algorithm for combined location and routing problems under capacity restrictions. In: Chinneck., J.W., Kristjansson, B., Saltzman, M.J. (Eds.), Operations Research and Cyber-infrastructure, 47. Springer, pp. 309–330.

Albareda-Sambola, M., Diaz, J.A., Fernandez, E., 2005. A compact model and tight bounds for a combined location–routing problem. Computers & Operations Research 32 (3), 407–428.

Balakrishnan, A., Ward, J.E., Wong, R.T., 1987. Integrated facility location and vehicle routing models: recent work and future prospects. American Journal of Mathematical and Management Sciences 7 (1–2), 35–61.

Baldacci, R., Mingozzi, A., Wolfler Calvo, R., 2011. An exact method for the capacitated location–routing problem. Operations Research 59 (5), 1284–1296.

Barreto, S., Ferreira, C., Paixão, J., Santos, B.S., 2007. Using clustering analysis in capacitated location–routing problem. European Journal of Operational Research 179 (3), 968–977.

Barreto, S.S., 2004. Análise e modelização de problemas de localização-distribuição [Analysis and Modelling of Location–Routing Problems]. Dissertation. University of Aveiro, Aveiro, Portugal.

Belenguer, J.M., Benavent, E., Prins, C., Prodhon, C., Wolfler Calvo, R., 2011. A banch-and-cut method for the capacitated location–routing problem. Computers & Operations Research 38 (6), 931–941.

Berman, O., Jaillet, P., Simchi-Levi, D., 1995. Location–routing problems with uncertainty. In: Drezner, Z. (Ed.), Facility Location: A Survey of Applications and Methods New York. Springer, pp. 427–452.

Billionnet, A., Elloumi, S., Djerbi, L.G., 2005. Designing radio-mobile access networks based on synchronous digital hierarchy rings. Computers & Operations Research 32 (2), 379–394.

Bouhafs, L., Hajjam, A., Koukam, A., 2006. A combination of simulated annealing and ant colony system for the capacitated location–routing problem. Lecture Notes in Computer Science 4251, 409–416.

Bullnheimer, B., Hartl, R.F., Strauss, C., 1999. A new rank based version of the ant system—a computational study. Central European Journal of Operations Research 7 (1), 25–38.

Chan, Y., Carter, W.B., Burnes, M.D., 2001. A multiple-depot, multiple-vehicle, location–routing problem with stochastically processed demands. Computers & Operations Research 28 (8), 803–826.

Chien, T.W., 1993. Heuristic procedures for practical-sized uncapacitated location-capacitated routing problems. Decision Sciences 24 (5), 995–1021.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics Part B 26 (1), 29–41.

Dorigo, M., Stützle, T., 2004. Ant Colony Optimization. Bradford Books. MIT Press, Cambridge, MA.

Duhamel, C., Lacomme, P., Prins, C., Prodhon, C., 2010. A GRASP $\times$ ELS approach for the capacitated location–routing problem. Computers & Operations Research 37 (11), 1912–1923.

Gambardella, L.M., Taillard, E., Agazzi, G., 1999. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. In: Corne, D., Dorigo, F., Glover, F. (Eds.), New Ideas in Optimization. McGraw-Hill, London, pp. 63–76.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co, New York.

Hansen, P.H., Hegedahl, B., Hjortkjær, S., Obel, B., 1994. A heuristic solution to the warehouse location–routing problem. European Journal of Operational Research 76 (1), 111–127.

Laporte, G., 1988. Location routing problems. In: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam, pp. 163–197.

Laporte, G., Louveaux, F., Mercure, H., 1989. Models and exact solutions for a class of stochastic location–routing problems. European Journal of Operational Research 39 (1), 71–78.

Laporte, G., Nobert, Y., Taillefer, S., 1988. Solving a family of multi-depot vehicle routing and location–routing problems. Transportation Science 22 (3), 161–172.

Laporte, G., Norbert, Y., 1981. An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research 6 (2), 224–226.

Laporte, G., Norbert, Y., Arpin, D., 1986. An exact algorithm for solving a capacitated location–routing problem. Annals of Operations Research 6 (9), 293–310.

Lin, C.K.Y., Chow, C.K., Chen, A., 2002. A location–routing-loading problem for bill delivery services. Computers & Industrial Engineering 43 (1), 5–25.

Marinakis, Y., Marinaki, M., 2008. A particle swarm optimization algorithm with path relinking for the location routing problem. Journal of Mathematical Modelling and Algorithms 7 (1), 59–78.

Min, H., Jayaraman, V., Srivastava, R., 1998. Combined location–routing problems: a synthesis and future research directions. European Journal of Operational Research 108 (1), 1–15.

Nagy, G., Salhi, S., 1996. Nested heuristic methods for the location–routing problem. Journal of the Operational Research Society 47 (9), 1166–1194.

Nagy, G., Salhi, S., 2007. Location–routing: issues, models and methods. European Journal of Operational Research 177 (2), 649–672.

Perl, J., 1983. A Unified Warehouse Location–Routing Analysis. Ph.D. dissertation, Northwestern University, Illinois, USA.

Prins, C., Prodhon, C., Wolfler Calvo, R., 2004. Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. In: Dolgui, A., Dauzère-Pérès, S. (Eds.), MOSIM'04, 2, Lavoisier. Ecole des Mines de Nantes, France, pp. 1115–1122.

Prins, C., Prodhon, C., Wolfler Calvo, R., 2006a. Solving the capacitated location–routing problem by a GRASP complemented by a learning process and a path relinking. 4OR 4 (3), 221–238.

Prins, C., Prodhon, C., Wolfler Calvo, R., 2006b. A memetic algorithm with population management (MA|PM) for the capacitated location–routing problem. Lecture Notes in Computer Science 3906, 183–194.

Prins, C., Prodhon, C., Wolfler Calvo, R., 2007. Solving the capacitated location–routing problem by a cooperative Lagrangean relaxation–granular tabu search heuristic. Transportation Science 41 (4), 470–483.

Salhi, S., Rand, G.K., 1989. The effect of ignoring routes when locating depots. European Journal of Operational Research 39 (2), 150–156.

Srivastava, R., 1993. Alternate solution procedures for the locating–routing problem. OMEG 21 (4), 497–506.

Tuzun, D., Burke, L.I., 1999. A two-phase tabu search approach to the location routing problem. European Journal of Operational Research 116 (1), 87–99.

Wang, X., Sun, X., Fang, Y., 2005. A Two-phase hybrid heuristic search approach to the location–routing problem. 2005 IEEE International Conference on Systems, Man and Cybernetics, 4, 3338–3343.

Wasner, M., Zäpfel, G., 2004. An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. International Journal of Production Economics 90 (3), 403–419.

Wu, T.H., Low, C., Bai, J.W., 2002. Heuristic solutions to multi-depot location–routing problems. Computers & Operations Research 29 (10), 1393–1415.

Yu, V.F., Lin, S.W., Lee, W.Y., Ting, C.J., 2010. A simulated annealing heuristic for the capacitated location routing problem. Computers & Industrial Engineering 58 (2), 288–299.