KKU Engineering Journal

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

An ant colony system (ACS) with a hybrid local search to solve vehicle routing problems

Suphan Sodsoon*[1], Savalee Uttra[1] and Sombat Sindhuchao[2]

[1]Logistics Engineering Program, Faculty of Agro-Industrial Technology Rajamangala University of Technology Isan, Kalasin Campus, 46000, Thailand
[2]Industrial Engineering Department, Faculty of Engineering, Ubon Ratchathani University, Ubon Ratchathani 34190, Thailand

Received May 2015
Accepted July 2015

Abstract

This research applied an Ant Colony System algorithm with a hybrid local search to solve vehicle routing problems (VRP) from a single depot when the customer requirements are known. VRP is an NP-hard optimization problem and has been successfully optimised using heuristics. A fleet of vehicles of a specific capacity are used to serve a number of customers at minimum cost, without violating the constraints of vehicle capacity. There are meta-heuristic approaches to solve these problems, such as Simulated Annealing, Genetic Algorithm, Tabu Search and the Ant Colony System algorithms. In this case, a hybrid local search was used (Cross-Exchange, Or-Opt and 2-Opt algorithm) with an Ant Colony System algorithm. The experimental design was tested on seven different problems from an online data set in the OR-Library. In five different problems, customers and the depot were randomly distributed in an approximately central location. The customers were grouped into clusters. The results were evaluated in terms of optimal routes using optimal distances. The experimental results are compared with those obtained from meta-heuristics and they show that the proposed method outperformed six meta-heuristics that have been presented in the literature.

Keywords: Vehicle routing problems, Ant colony system, Hybrid local search, Cross-exchange, Or-opt and 2-Opt algorithm

1. Introduction

The distribution problem in which vehicles based at a central facility (depot) are required to visit geographically dispersed customers during a given time period in order to fulfill known customer requirements are referred to as VRP. More than 50 years ago Dantzing and Ramser introduced Vehicle Routing Problems (VRP) [1]. They proposed the first mathematical programming formulation and algorithmic approach concerning the delivery of gasoline to service stations. Clarke and Wright [2] proposed an effective greedy heuristic to improve the Dantzig-Ramser problem. Many models and algorithms can give optimal and approximate solutions for VRP [3]. Generally, distribution or collection of goods from customers to a depot is called VRP or Vehicle Scheduling Problems. In particular, VRP is an *NP*-hard problem [4], for which exact algorithms are only efficient for small problems. Heuristics and meta-heuristics are applied because real-life problems are very complex problems. Recently, many researchers have developed hybridization algorithms to improve the quality of the solutions to such problems, for example, by using a hybrid approach with Genetic Algorithms and Ant Colony Optimization by [5]. This algorithm is a combination of these two algorithms for solving partially Dynamic-VRP in which hybrid ACS acts well for the supposed D-VRP by [6]. ACS with a local search algorithm can give better quality solutions than the original by [7] and [8], as it presents a cooperative ACO using vehicle pheromones and cooperatively organizing route guidance through pheromones.

This paper proposes ACS with a new hybrid local search method for solving VRP with multiple customers, a homogeneous fleet, limited capacity of vehicles, although the time window is not considered in this case. It can be used for redesigning the logistics network as well as improving the planning of the distribution network. The main objective of the VRP is to minimize the total distance travelled by all vehicles by means of routing and this problem can be described as assigning optimal delivery routes from a depot to a number of geographically distributed customers, subject to constraints. ACS is a population based search technique that conducts multiple searches in a particular area and which is very diversified. However, a new hybrid local search method is an improved method which is used in many local search procedures for solving VRPs such as search intensification. Therefore, we propose an improved ACS algorithm for VRP which uses different local search approaches and combines the strengths of both search heuristics. Finally, the proposed method was tested with data samples from the OR-library and compared with the performance of other meta-heuristics. The paper is organized as follows: Section 2, a formal definition of the model is

*Corresponding author. Tel.: +6683 581 3869; fax: +66 4381 3070
Email address: suphan_sodsoon@hotmail.com
doi: 10.14456/kkuenj.2016.1

given. In sections 3 and 4 a proposed algorithm and numerical analysis are discussed respectively. Next, in section 5, the computational tests are described. Section 6 compares six algorithms in the literature. Finally, some overall conclusions are given and some suggestions are made for future work in section 6.

2. Description model

The basic VRP is to route vehicles using one route per vehicle, each starting and finishing at the depot, so that all customers' requirements are satisfied and the total distance travelled is minimized. Figure 1 shows a graphic representation of a VRP example with 4 vehicles serving 12 customers forming 4 routes. So, the solutions of this example are: 0-1-2-3-0, 0-4-5-6-0, 0-7-8-9-0, and 0-10-11-12-0.

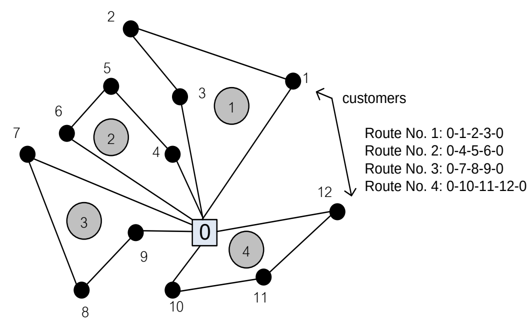


Figure 1 Typical output for VRPs

Here, we give the VRPs formulation found in the literature and the notation used in the mathematical models are as follows.

q_i Requirement of the customer $i, j \in V$

d_{ij} Distance between the customers v_i to v_j

$K = \{k_1, k_, ..., k_m\}$ represents the fleet of vehicles

Q Capacity of each vehicle $k_i \in K$ (the fleet has identical vehicles)

In order to find the order of the visits to the customers, we define the decision variables as follows:

$$x_{i,j}^k = \begin{cases} 1 & \text{if the vehicle } k \text{ visits the customer } j \text{ directly after the customer } i \\ 0 & \text{otherwise} \end{cases}$$

The objective function is

$$\sum_{i=0}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} \sum_{k=1}^{K} c_{ij} x_{ij}^k \quad (1)$$

Subject to:

$$\sum_{i=0}^{N} \sum_{k=1}^{K} x_{ij}^k = 1 \;\; \forall j \in \{1, ..., N\} \quad (2)$$

$$\sum_{j=0}^{N} \sum_{k=1}^{K} x_{ij}^k = 1 \;\; \forall j \in \{1, ..., N\} \quad (3)$$

$$\sum_{i=0}^{N} x_{ip}^k - \sum_{j=0}^{N} x_{pj}^k = 0 \quad (4)$$

$$\forall p \in \{1, ..., N\}, \;\; \forall k \in \{1, ..., K\}$$

$$\sum_{j=0}^{N} q_j (\sum_{i=0}^{N} x_{ij}^k) \le Q \;\; \forall k \in \{1, ..., K\} \quad (5)$$

$$\sum_{j=1}^{N} x_{ij}^k \le 1 \;\; \forall i \in \{1, ..., N\}, \forall k \in \{1, ..., K\} \quad (6)$$

$$\sum_{i=1}^{N} x_{ij}^k \le 1 \;\; \forall j \in \{1, ..., N\}, \forall k \in \{1, ..., K\} \quad (7)$$

$$x_{ij}^k \in \{0, 1\} \;\; \forall i, j \in \{1, ..., N\}, \forall k \in \{1, ..., K\} \quad (8)$$

The objective function of the total distance travelled minimization is expressed by Eq. (1). Constraints Eq. (2)-(3) route continuity is enforced by Eq. (4) as once a vehicle has arrived at a node, it must also leave that node. No vehicle can service a customer's requirements if they exceed the vehicle capacity in Eq. (5). Each vehicle should be scheduled for use only once by Eq. (6)-(7) and the last one in Eq. (8), all variables are binary.

3. Ant colony optimization (ACO)

Ant Colony Optimization is one of the most popular algorithms in the research field of swarm intelligence. The behavior of real ants seeks a shorter path between nest and food sources as shown in Figure 2.

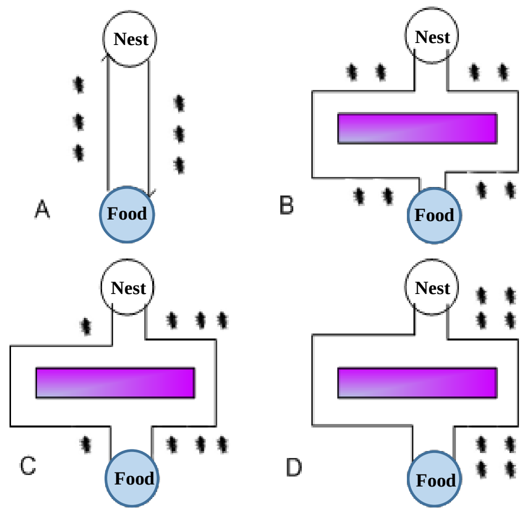


Figure 2 Behaviors of real ant colony [9]

Figure 2A shows the path of ants between a nest and a food source. Figure 2B shows an obstacle interrupting the trail. Figure 2C shows pheromone is deposited more quickly on the shorter path. And Figure 2D shows a new pheromone trail is formed along the shorter path [9]. The behavior of ants has been extensively studied and has inspired a number of methods and techniques among which the most successful is the general purpose optimization technique known as Ant Colony Optimization (ACO).The first version of ACO was called the Ant System (AS).

3.1 Ant Colony System

The ACS is based on the Ant System (AS) algorithm development by Dorigo and Gambardella [10]. The idea of the proposed ACS approach is to correspond the pheromone trails with the ants' adaptive memory, which represents the weight of ranking index of the constructive heuristic and is updated at run time by ants. Meanwhile, the length of an edge of the path stands for the ranking index of the applied constructive heuristic. Artificial ants (agents) use transition rules Eq. (9) and Eq. (10) to mimic real ants in choosing next target, and use trail update rules Eq. (11) and Eq. (12) to model how real ants leave pheromone on routes.

Phase I: Construction process

α and β are parameters to weigh pheromone density and the heuristic information in the probability. For simplicity, the proposed VRP uses $\eta_{ij} = 1/d_{ij}$, where d_{ij} is the travel distance between customers i and j The parameter q is a random variable following uniform distribution over [0,1]. When, q_0 is a control parameter to choose the way to select the next customers in Eq. (9) *and* Eq. (10). For example; If $q \le q_0$, the path with the highest pheromone level is chosen for the next customers with probability q_0. Otherwise, it will use J, which represents a random selection, based on values in Eq. (11).

$$j = \begin{cases} \arg\max_{l \in N_i^k} [\tau_{il}^{\alpha} \eta_{il}^{\beta}], & if\ q \le q_0 \\ J, & if\ q > q_0 \end{cases} \tag{9}$$

$$p_{ij}^k = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}^{\alpha} \eta_{il}^{\beta}]} \qquad if\ j \in N_i^k, \tag{10}$$

N_i^k is the set of all feasible customers.

Phase II: Updating pheromone trials

- Local pheromone update is used in the proposed ACS to exploit the search space globally while avoiding local convergence. The local pheromone update rule is executed every time an ant chooses its next customer, when the pheromone level will change with the equation as Eq. (11).

$$\tau_{ij} = (1-\rho_l).\tau_{ij} + \rho_l.\tau_0 \tag{11}$$

Where; ρ_l is the local evaporation parameter such that $0 < \rho_l < 1$ and τ_0 is the initial level of pheromone.

- The global pheromone update increases the pheromone level on the best-so-far path. After all the ants have completed their tours, the pheromone amount is updated by applying the follow global updating rule Eq. (12).

$$\tau_{ij} = \begin{cases} (1-\rho_g).\tau_{ij} + \rho_g.\Delta\tau_{ij} \\ if\ customer\,(i, j)\ represent\ the\ best\ tours, \\ \tau_{ij} \quad otherwise. \end{cases} \tag{12}$$

$\Delta\tau_{ij} = \dfrac{C}{l_{best}}$, where, ρ_g is the global evaporate parameter such that $0 < \rho_g < 1$, l_{best} is the best-so-far solution and C is a constant that control the amount of pheromone updates.

Phase III: ACS-Hybrid local search

This study proposes a hybrid local search approach, while it takes advantage of ACS in order to solve VRPs. Although ACS performs very well in a global search, they usually take a long time to reach the global optimum solution. Thus, in this paper we will introduce two strategies used in ACS. A hybrid local search approach is used to improve the quality solutions. We call the proposed approach the ACS-HLS algorithm. The working of the proposed algorithm is summarized in the flowchart as shown in Figure 3.

ACS-HLS Algorithm

Step 1: Set parameters: number of ants, create a vehicle and load it to maximum capacity, and exploitation with probability (q_0) =0.9 β =2.0 and ρ =0.1

Step 2: Load data samples and Initialize data structures. Obtain a feasible solution (ψ^{nn}) with the nearest heuristic neighbor. $\psi^{gb} = \psi^{nn}$, where ψ^{gb} is the best global solution, l_{gb} is route length that belongs to the best solution. Set initially $l_{gb} = l_{nn}$.

- Initialize pheromone level for each pair $(r,s):\ \tau(r,s) = \tau_0$
  Where, $\tau_0 = (n.l_{nn})^{-1}$

Step 3: Set I-iterations is stop criterion -Cycle and ant of vehicle to be in depot

Step 4: For every unvisited customer

Do: Choose probabilistically the next node j using η_{ij} in exploitation and exploration mechanisms as Eq. (9)-(10)

- Check capacity of vehicle
If Capacity feasibility = True, Then:
-Current node: $\psi^k \leftarrow \psi^k + < j >$
- Route Length: $l_k = l_k + d_{ij}$
- Local updating according to Eq. (12).
If selected node $< j >$
Then: $load_k = load_{v,k} + d_{ij}$
Else: Return to the depot: $v \leftarrow v+1$ and set: $load_{v,k} = 0.0$
End if
Else: Return to the depot: $v \leftarrow v+1$ and set: $load_{v,k} = 0.0$

Wile: set customers $= \{\}$

Do Until stopping criterion = True

Step 5: Improvement solutions loop with ACS-HLS Algorithm

Step 6: Global updating

- Global updating according to Eq. (12)
- Where l_{gb} is global best solution

Step 7: Until a stop criterion is met (I-iterations)

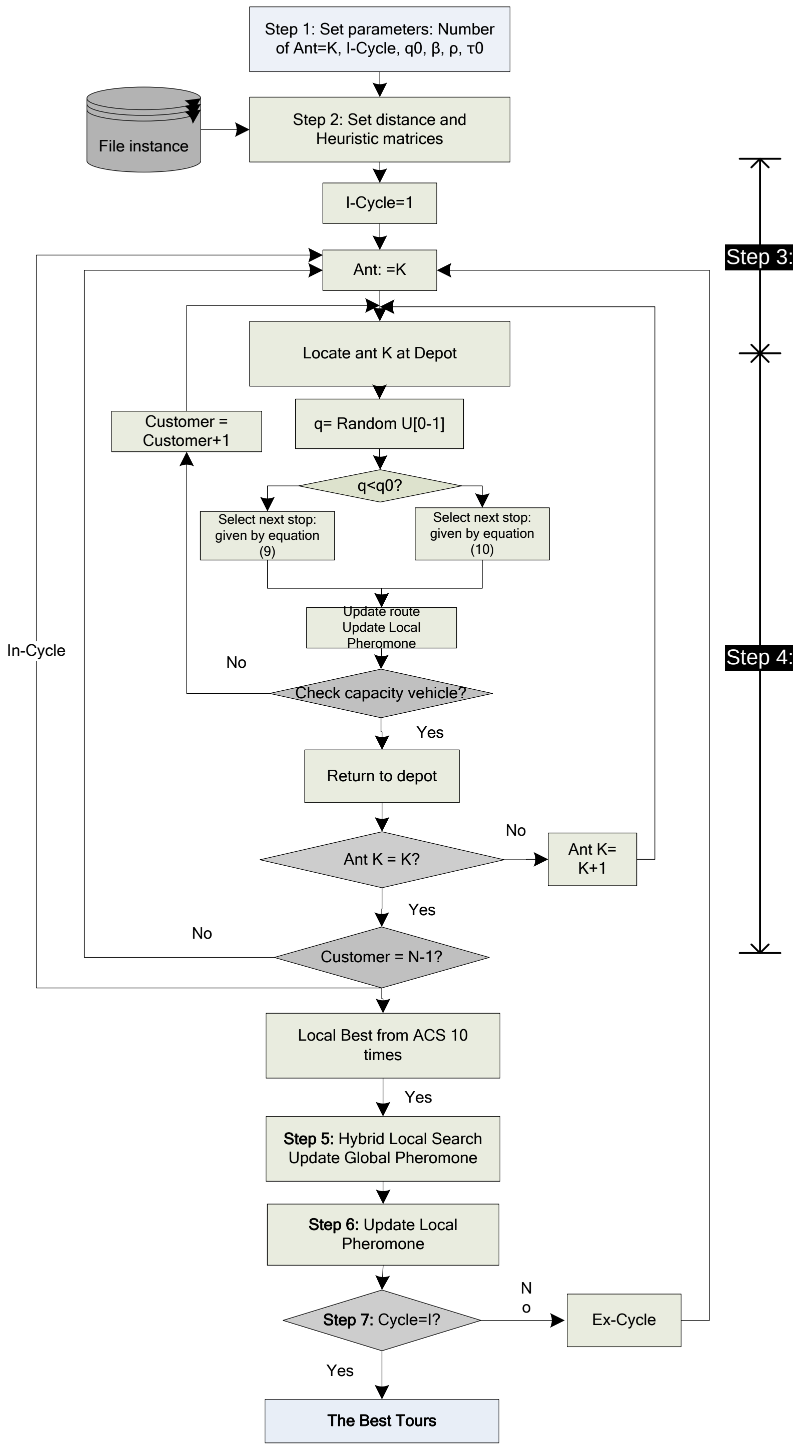


Figure 3 Flowchart of an ACS-HLS Algorithm

3.2 Hybrid local search loop

In step 5, a hybrid local search is an improved method commonly used in local search procedures for solving VRPs. After each ant constructs completed tour in step 4. The initial solutions obtained from ACS are applied to the hybrid local search approach. There are two alternatives for a hybrid local search approach. Alternative 1, starting from Inter-Routes with 2-Opt*, Cross-Exchange algorithm and Intra-Routes procedure with Or-Opt and 2-Opt algorithm, respectively. For alternative 2, starting from Inter-Route procedure with Cross-Exchange, 2-Opt* and Intra-Routes procedure with Or-Opt and 2-Opt algorithm, respectively. In this paper we suggest a hybrid local search algorithm which combines the principles of an ant colony system. This approach is an improved method commonly used in local search procedures for solving VRPs as follows:

Inter route operators:

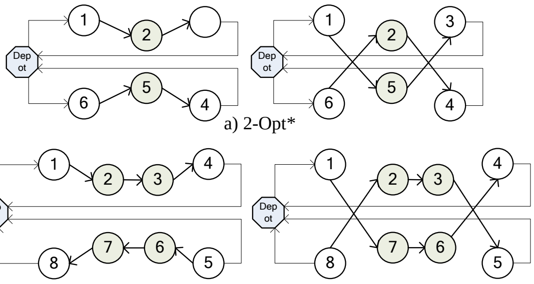
- Inter Route 2-Opt* can be considered like exchanges where 2 and 5 customers are inter route as shown in Figure 4a.

- The Cross-Exchange operator swaps two groups of customers from one route to another as shown in Figure 4b. The groups consist of one to a maximum of four customers. Bigger groups are inefficient mainly because of slow execution time. To prevent neighborhoods from being interlaced (Cross-Exchange) only two groups 2-3 or 7-6 can have only two customers in the groups.

Intra route operators:

-Intra route 2-Opt operator transforms the intersection of arcs if savings exist after changed the direction of the arcs between 1-3 and 4-2 and delete and add the appropriate arcs as shown in Figure 5a.

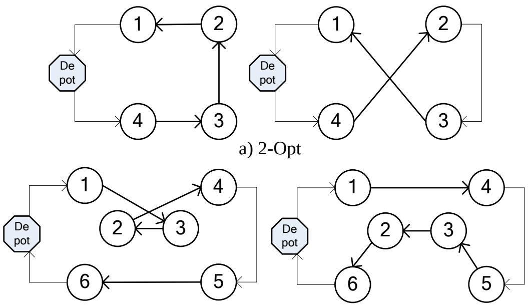
- These operators are intra-route operator with Or-Opt the intersection of arcs with reordering customers on a route as shown in Figure 5b. This operator is practical because it is very fast. The alternative slower scenario is two relocate operator execution. In Figure 6, we show an example of a hybrid local search flowchart with two alternative components.



a) 2-Opt*

b) Cross-Exchange

Figure 4 Inter-Route operators for VRPs



a) 2-Opt

b) Or-Opt

Figure 5 Intra-Route operators for VRPs

4. Numerical analysis

Table 1 shows the characteristics of 7 sample problems of [11-12]. These problems range in size from n = 50 to n = 199 customers and can be downloaded from the OR-library [13]. They have been widely used as benchmarks. The first 5 problems (p01 to p05) are when customers are randomly distributed in relation to the depot in an approximately central location. While in the last two problems (p11 and p12), the customers are grouped into clusters. The data samples of the vehicle routing problems for the seven problems are summarized in Table 1.

Table 1 Characteristics data instances of the VRPs problem

Problem Instance	n = number of nodes	Q = capacity of vehicle	BKS
p01	50	160	524.61
p02	75	140	835.26
p03	100	200	826.14
p04	150	200	1028.42
p05	199	200	1291.45
p11	120	200	1042.11
p12	100	200	819.56

BKS = the best known solutions from Tabu Search method [14], [15]

5. Experiments test

In our algorithm, we coded in visual studio C++ program with version 6.0 and all of the problems were conducted on a Pentium Intel(R) Core™2Dual CPU 2.20 Ghz 0.99 Gb of RAM. By setting the values for the parameters as: $\alpha = 1$, $\beta = 2$ and $\rho = 0.10$ each problem was solved 5 times so that the results were replicated as shown in Table 2. The %Gap closed is computed as Eq. (13).

$$\frac{C(s^b) - C(s^a)}{C(s^a)} \text{x100} \qquad (13)$$

Where $C(s^b)$ is the best solution by the algorithm for a given instance and $C(s^a)$ is the overall best known solution (BKS). Tables 2 and 3 show the results expressed in terms of the average total distance and the one solution obtained from ACS-HLS performance gap exceeding about 1.362% and 0.08%, respectively (for all instances) of the BKS results.

6. Comparison of ACS-HLS with other meta-heuristics

In this section we present the performance of the ACS-HLS algorithm and compare it with the results of the other meta-heuristic (TAG[16], STO[17], TST[14], GAB[18], ASZ[19] and ACS[20]) in terms of total distance, as shown in Table 4. The numbers in bold indicate the best solution from the proposed method is equal to the solutions from the other methods in terms of total travel distances and compare them in terms of gap, as shown in Table 4, the numbers in bold indicate that the best solution is equal to the best result from BKS. The average gap of all benchmark problems obtained by ACS-HLS is the lowest from the six algorithms. When comparing the gaps of the total travelled distance of the six algorithms, the computational results of the ACS-HLS algorithms applied to seven problem instances are summarized in Table 4.

The method which gives the best solution outperforms in [16], [17], [18] and [20] in terms of the %Gap from the BKS. TST [14] and ACS-HLS can be a percentage of gaps as well. However, the new hybrid local search with ACS algorithms are the most powerful local search in this research study which can improve the solutions for both inter-route and intra-route at the same time. Figure 7 shows a comparison between the gap values of the meta-heuristic algorithms, where the gap is defined as the percentage of gap from the BKS in the literature. The results show that the ACS-HLS method provides better solutions than the previous methods.

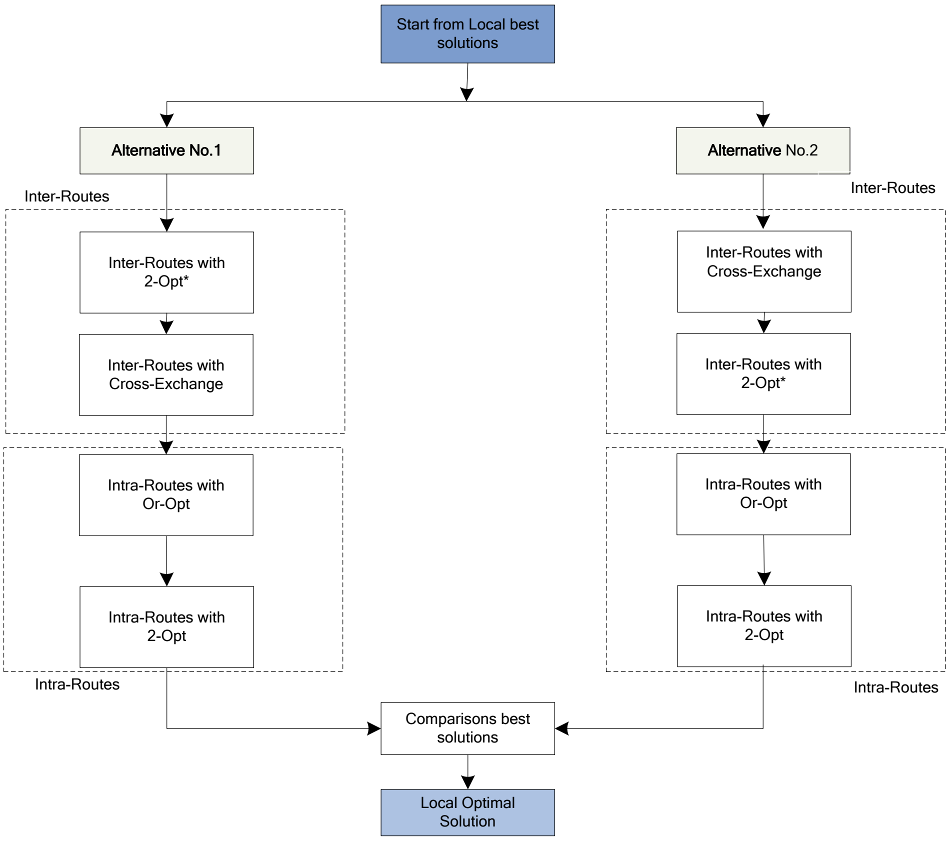


Figure 6 Hybrid Local Search (HLS)

Table 2 The best solutions from ACS-HLS

Problems	1		2		3		4		5		Avg.	BKS	%Gap
	TD	CPU.T	TD	CPU.T	TD	CPU.T	TD	CPU.T	TD	CPU.T			
p01	524.61	146.30	528.29	135.61	524.61	150.15	527.68	15.62	527.68	139.85	526.57	524.61	0.374
p02	835.26	362.97	854.87	314.70	856.17	318.57	835.26	321.26	837.63	355.61	843.84	835.26	1.027
p03	837.63	1447.75	834.06	1512.35	831.47	1438.70	826.14	1450.66	826.14	1460.95	831.09	826.14	0.599
p04	1056.14	3541.83	1028.42	3550.76	1060.95	3812.65	1054.12	3957.81	1053.98	3926.73	1050.72	1028.42	2.169
p05	1366.08	5143.38	1371.38	5254.23	1361.57	5521.36	1361.57	5312.56	1298.31	5234.56	1351.78	1291.45	4.672
p11	1048.17	2422.63	1046.39	2452.72	1056.11	2460.52	1053.98	2435.69	1042.11	2440.55	1049.35	1042.11	0.695
p12	819.56	937.39	819.56	930.19	819.56	940.65	819.56	950.16	819.56	945.75	819.56	819.56	0.000
%Gap over all of instances													**1.362%**

Note: TD = Total distance, CPU.T = Computational time (sec.) and Avg. = Average distance

Table 3 The bester solutions of ACS-HLS

Problem	ACS-HLS	BKS	%Gap
p01	**524.61**	524.61	0.00
p02	**835.26**	835.26	0.00
p03	**826.14**	826.14	0.00
p04	**1,028.42**	1,028.42	0.00
p05	**1298.31**	1,291.45	0.53
p11	**1,042.11**	1,042.11	0.00
p12	**819.56**	819.56	0.00
%Gap of the bester solutions from ACS-HLS			**0.08**

Table 4 Comparison of ACS-HLS with six meta-heuristics

Problem	TAG		STO		TST		GAB		ASZ		ACS		ACS-HLS	
	Best known	%Gap	Best known	%Gap	Best known	%Gap	Best known	%Gap	Best known	%Gap	Best known	%Gap	Best known	%Gap
p01	524.61	0	524.61	0	524.61	0	524.61	0	524.61	0	524.61	0	***524.61****	***0***
p02	835.26	0	835.26	0	835.26	0	835.26	0	835.26	0	835.26	0	***835.26****	***0***
p03	826.14	0	829	0.35	826.14	0	829	0.35	830.14	0.48	829.45	0.4	***826.14****	***0***
p04	1031.07	0.26	1044	0	1028.42	0	1034	0.54	1038.2	1.1	1039.78	1.1	***1028.42****	***0***
p05	1311.35	1.54	1334	3.29	1298.31	0.53	1327	2.75	1307.18	1.22	1298.31	0.53	**1298.31**	***0.53***
p11	1042.11	0	1042.11	0	1042.11	0	1046	0	1044.12	0.19	1042.11	0	***1042.11****	***0***
p12	819.56	0	819.56	0	819.56	0	819.56	0	824.31	0.58	819.56	0	***819.56****	***0***
%Gap		0.26		0.52		*0.08**		0.52		0.51		0.29		***0.08****

Notes: TAG: tabu search [16] GAB: genetic algorithm [18]
STO: simulated annealing and tabu search [17] ASZ: ACO and scatter search [19]
TST: tabu search [14] ACS: Ant Colony System [20]

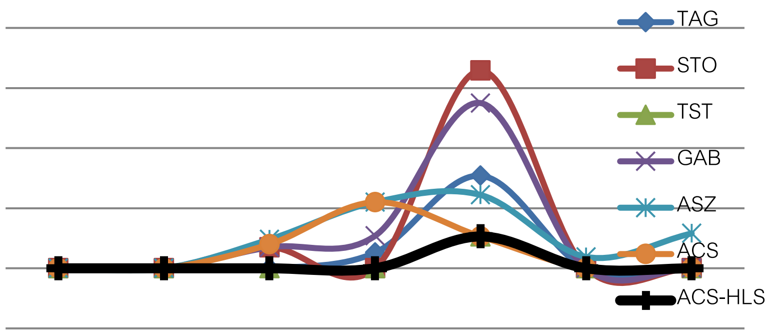


Figure 7 Comparison of ACS-HLS with other meta-heuristics

7. Conclusions

This paper presents a new ACS that is different from the common ACS. The main idea is that a modified ACS algorithm can find good solutions efficiently. ACS combined with many local search techniques is used for solving VRPs. We call this method a hybrid local search algorithm. Hybrid Local Search also tries to improve both the performance of the algorithm and the quality of solutions for the VRPs by using many local search algorithms for both Inter-Routes and Intra-Routes, respectively.

The computational results show that better solutions can be obtained from the ACS-HLS algorithm than the six other approaches proposed in the literature. The proposed methods are tested in terms of the average total distance only and the solution obtained from the ACS-HLS %Gap exceeds 1.362% and 0.08%, respectively. In the future, further research will be conducted to improve the local search in order to obtain better solutions and other local search algorithms will be examined to enhance the performance of this method in improving the solutions.

8. References

[1] Dantzig GB, Ramser JH. The truck dispatching problem. Manag Sci. 1959;6:80-91.
[2] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Oper Res. 1964;12:568-81.
[3] Toth P, Vigo D. The Vehicle routing problem in SIAM monographs discrete mathematics and applications. Philadelphia: Society for Industrial & Applied; 2002.
[4] Lenstra J, Rinnooy Kan A. Complexity of vehicle routing and scheduling problems. Networks. 1981;(11):221-7.
[5] Li N, Wang S, Li Y. A Hybrid approach of GA and ACO for VRP. J Comput Inform Syst. 2011;7(13): 4939-46.
[6] Rashidi H, Farahani RZ. A hybrid ant colony system for partially dynamic vehicle routing problem. Am J Oper Res. 2012;2(4):31-44.
[7] Gambardella LM, Montemanni R, Weyland D. Coupling ant colony systems with strong local searches. Eur J Oper Res. 2012;220:831-43.
[8] Tan KC, Lee LH, Ou K. Hybrid genetic algorithms in solving vehicle routing problems with time window constraints. Asia Pac J Oper Res. 2001;18:121-30.
[9] Dorigo M. Optimization, learning and natural algorithms [dissertation]. Milan: Politecnico di Milano; 1992.
[10] Taillard E, Badeau P, Gendreau M, Guertin F, Potvin JY. A new neighborhood structure for the vehicle routing problem with time windows. Technical report CRT-95-66. Montréal, Centre de recherche sur les transports, Universit´e de Montr´eal; 1995.
[11] Christofides N, Mingozzi A, Toth P. Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations. Math Program. 1981;20:255-82.
[12] Taillard ÉD. Parallel iterative search methods for vehicle routing problems. Networks. 1993;23:661-73.
[13] Automatic Control [Internet]. West Lafayette: School of Electrical and Computer Engineering, Purdue University. Available from: http://www.ece.purdue.edu/ECE/Research/ARS/ARS2000/PART_I/Section1/1_19.whtml.

[14] Taillard RE. Parallel iterative search methods for vehicle routing problems. Networks. 1993;23: 661-73.
[15] Rochat Y, Taillard E. Probabilistic diversification and intensification in local search for vehicle routing. J Heuristics. 1995;1:147-67.
[16] Gendreau M, Hertz A, Laporte G. A tabu search heuristic for the vehicle routing problem. Manag Sci. 1994;40:1276-90.
[17] Osman IH. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann Oper Res. 1993;41:421-51.
[18] Baker BM, Ayechew MA. A genetic algorithm for the vehicle routing problem. Comput Oper Res. 2003;30: 787-800.
[19] Zhang X, Tang L. A new hybrid ant colony optimization algorithm for the vehicle routing problem. Pattern Recogn Lett. 2009;30:848-55.
[20] Majid YK, Mohammad S. An optimization algorithm for the capacitated vehicle routing problem based on ant colony system. Aust J Basic Appl Sci. 2011;5(12): 2729-37.