An Optimization Algorithm for the Capacitated Vehicle Routing Problem Based on Ant Colony System

Majid Yousefi khoshbakht, Mohammad Sedighpour

Young Researchers Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran.
Lecturer, Hamedan Branch, Islamic Azad University, Hamedan, Iran.

Abstract: The Capacitated Vehicle Routing Problem (CVRP), a well-known combinatorial optimization problem, holds a central place in logistics management. The description of this problem appeared in the literature over 50 years ago, but has just recently attracted the attention of practitioners and researchers. In this paper, aimed at the disadvantages existed in the current Ant Colony System (ACS), a modification to pheromone evaluation model of ACS is proposed, focusing on improving the performance of traditional algorithm. This trigonometric function can avoid premature convergence of ACS and exploit more strong solutions. Comparison between this method and famous meta-heuristic algorithms shows the effectiveness of the proposed approach.

Key words: Ant Colony System, 3-Opt Algorithm, Capacitated Vehicle Routing Problem, Meta-heuristic Algorithms.

INTRODUCTION

The Capacitated Vehicle Routing Problem (CVRP) is one of the most important combinatorial optimization problems that nowadays, it has been received much attention by researchers and scientists. So, many exact, heuristic and meta-heuristic algorithms have been proposed to solve it in recent decades. The CVRP is the basic version of the Vehicle Routing Problem (VRP), where all customers are delivery customers, the demands are known, all vehicles are identical and they belong to the same central depot. The imposed constraints are related to the capacity of the vehicles, may also be restricted in the total distance it can travel and all customers must be served by a single route. In this problem, the objective is to find a set of delivery routes satisfying these requirements and giving minimal total travel cost.

To make CVRP models more realistic and applicable, there are many variety of the VRP obtained by adding constraints to the basic model. Examples of such extensions are Multi-depot VRP (MDVRP), VRP with Pickup and Delivery (VRPPD), VRP with Time Windows (VRPTW), VRP with backhauls (VRPB), Periodic VRP (PVRP), Stochastic VRP (SVRP), Site Dependent VRP (SDVRP), Split Delivery VRP, Open VRP (OVRP) and so on with different constraints (Bodin, L.D., 1983; Christofides, N., 1981).

Because this problem is known to be NP-hard problem in the strong sense (Garey, MR., Johnson, DS., 1979), and it cannot be solved to optimality within reasonable time, researchers have developed the heuristic algorithms using a permissible solution instead of the optimal solution in the last 40 years. These approaches can obtain feasible solutions within a reasonable computing time. In other words, heuristic methods cannot guarantee any specified solution quality, but many of them are known to give good results in short time also for large instances.

Heuristic algorithms for solving VRP can be categorized into following three groups:
1) construction heuristic
2) improvement heuristic
3) meta-heuristic algorithms

Construction heuristic maybe is the most well known optimization strategies for solving the combinatorial optimization problem. These methods start from null solution and generate feasible solutions by accomplishing sequences of simple steps. This procedure is continued until a full solution will be constructed and given criterion will be optimized. So, the construction heuristic approaches can find generally one feasible solution quickly, but this feasible solution may have a large disparity in compare to the best solution.

There are many celebrated algorithms in this class such as savings heuristic of Clarke and Wright, (1964) that gains single route instead of two routes according to the savings obtained by this merger. The sweep algorithm is another famous construction heuristic that is proposed by Gillett and Miller, (1974). Other renowned algorithms that belong to construction heuristic are the modified savings heuristic of Paessens (1979), the insertion heuristic of Christofides et al. (1979), and the petal algorithm of Renaud et al. (1996). More details in the survey papers can be found of Cordeau et al (2002), which presents a broad analysis of a number of famous construction heuristics.
Improvement algorithms try to find an improved solution from a poorer solution (usually generated by a construction heuristic). All improvement algorithms for the VRP nearly use replacing a set of arcs for obtaining a better solution. These operations are iterated until improvements in the value of the objective are found or performance time of algorithm is over. Solutions produced with respect to the set of solution modifications considered usually go ahead to local optima. This group as like as Construction heuristic has a lot of popular algorithms such as k-opt approach for single vehicle routes proposed by Croes (1958), local search proposed by Rochat and Taillard (1995), shift-sequence algorithm proposed by Fahle (1999). For more information, we refer the reader (1994).

Other new kinds of popular heuristic methods are meta-heuristic arisen in the last 20 years. Since meta-heuristic approaches try to combine basic heuristic methods in higher level frameworks aimed at exploring a search space, they are very efficient and effective for escaping local optimum values. Because meta-heuristic approaches can obtain suboptimal solution or even global optimal solutions with reasonable computation time, and they give competitive results in compare to other algorithms, most of the research efforts aimed at solving the VRP have focused on the development of various meta-heuristic algorithms and many algorithms have been applied in recent years such as tabu search (Wassan, N.A., Osman, I.H, 2002; Leung, S.C.H., 2011) have obtained the best known results to benchmark VRPs, simulated annealing (Osman, I.H., 1993), genetic algorithms (Baker, B.M., Ayecheh, M.A., 2003), evolutionary algorithms (Prins, C, 2004), large neighborhood search (Hong, I., 2012; Pisinger, D. and Ropke, S., 2010), and ant colony optimization (ACO) (Bin, Y., 2009).

Recently, many researchers have found that the employment of hybridization in optimization problems can improve the quality of problem solving in compare to heuristic and meta-heuristic approaches. These algorithm such as Genetic algorithms with a local search (Prins, C., 2009), Genetic algorithms with sweep algorithm and nearest addition method (Wang, C.H. and Lu, J.Z, 2009), ACO and greedy heuristic (Zhang, X., Tang, L, 2009), simulated annealing and tabu search (Lin, S.W., et al., 2009), neural networks and genetic algorithms (Potvin, J.Y., Dube, D. and Robillard, C, 1996), genetic algorithm and ACO (Reimann, M., Shitovba, S. and Nepomuceno, E, 2001) and others have more ability for finding a optimal solution. So, nowadays they have been considered by researchers and scientists.

The paper is organized as follows: Section 2 a formal definition of the CVRP problem is explained. In Sections 3 and 4 the ACO and a proposed algorithm are respectively discussed. Next, In Section 5, computational tests are described and compared to other existing algorithms in the literature. Finally, some overall conclusions and suggest directions for future work are drown in section 6.

**The Capacitated Vehicle Routing Problem:**

The most elementary version of the VRP is the CVRP. This problem has been celebrated problem in the field of distribution and logistics for at least 50 years. It consists of a number of customers with a known demand level that must be visited from a single depot. The problem is solved under the following constraints:

- Each customer is visited only once by a single vehicle
- Only one vehicle is allowed to visit each customer
- Each vehicle must start and end its route at the depot
- Total demand serviced by each vehicle is limited and may also be restricted in the total distance that it can travel

Possible objectives is to find a set of routes which minimizes the total distance or cost of the combined routes of a number of vehicles m that must service a number of customers n, or minimizes the number of vehicles required and the total distance traveled with this number of vehicles.

Mathematically classical VRP is described as a weighted graph $G = (V, A)$ where the vertices are represented by $V = \{0,1,\ldots,n\}$, and the arcs are represented by $A = \{(i,j) : i,j \in V, i \neq j\}$. Consider a central depot where every vehicle starts its route located at 0, and let $v' = v - \{0\}$ be used as the set of n customers. A matrix $D = (d_{ij})_{(n+1)\times(n+1)}$ of non-negative costs defined on $A$ which is measured using Euclidean computations. When $d_{ij} = d_{ji}$ for all $(i,j) \in A$, the problem is said to be symmetric and it is then common to replace $A$ with the edge set $E = \{(i,j) : i,j \in V, i < j\}$. Each node is associated with a fixed quantity $q_i$ of goods to be delivered (a quantity $q_i = 0$ is associated to the depot $v_0$), and the quantity of goods to be delivered on a route should never exceed the vehicle capacity $Q$. By defining a maximum route length, $L$, which each vehicle may not exceed, number of customers can visited by every vehicle are limited. An example of a single solution consisting of a set of routes constructed for a classical VRP is presented in Fig. 1, where $m=4$ and $n=18$. So, the solution of this example is that: 0-15-14-16-0; 0-3-2-4-10-5-0; 0-6-17-18-8-0; 0-7-9-11-12-13-0.
Various mathematical formulations of the VRP have been proposed for classical VRP, for example see (Laporte, G., 1992) for an overview.

**Ant Colony Optimization:**

One of the most important meta-heuristic approaches created by Dorigo et al in the last decade which proved its performance for traveling salesman problem and other problems is the ACO (Dorigo, M., 2000) that is used as same as Genetic Algorithm and Particle Swarm Optimization to solve the combinational optimization problems in 1992 (Dorigo, M, 1992). This population-based approach has been successfully applied to several NP-hard combinatorial optimization problems such as VRP (Bullnheimer, B., 1999), communications networks (Di Caro, G., Dorigo, M, 1998), sequential ordering problem (Gambardella, L.M., Dorigo, M, 1997), quadratic assignment problem (Maniezzo, V, 1999), shortest common super sequence problem (Michel, R., Middendorf, M, 1998). The ACO, inspired by the nature, simulates the natural ant treatment for food finding and applies it for solving the combinational optimization problem for which has not been found any effective algorithm yet. Studies on real ant show that despite the ants do not have the sense of seeing; they can find the shortest path from the food sources to the nest (Fig. 2). Some evaporated material called pheromone, secreted by the ants when they move from one place to another to find the shortest path. Ants secrete this chemical material first for guiding other ants, which are going to exit the nest later, secondly, for recognition of the return path to the nest. Hence the route that ants traveled is marked by this chemical material so after a few time, at the same time, more ants pass this shorter path and remain much more pheromone on this shorter path. In addition, the ants instinct with more probability select the route, which has more pheromone than others.

![Fig. 1: Sample of solving the VRP](Image)

This experience shows that the simple swarm intelligence, which is used by ants for finding food, leads to solve the hard combinational problems and reach a solution, which is near to the optimal situation.

The ACO has been developed well lately; modifying the method of updating the local and global pheromones, and the distribution of ants on the nodes, are some examples. These developments lead to more efficient algorithms like $AS_{psa}$, $AS_{rapp}$, $ACS$ and $MMAS$. On the other hand, application and efficiency of these...
algorithms has gained more attention, compared to some other meta-heuristic algorithms including Genetic Algorithm, Simulated Annealing, etc. Despite of good advantages of ACO algorithm compared to the other methods, the large distances between the solutions cause difficulties in gaining better solutions.

Many researchers have tried to modify this algorithm as much as possible to overcome these difficulties. The investigations by Stuzzle and Hoos (1997), Qinghong and Zang (1999), Bin and Zhongzhi (2001) and Yousefikhoshbakht et al (2009; 2008) are some examples of these efforts.

**Proposed Algorithm:**

In this section, first the ACS is presented and then the improvement of ACS will be explained in more detail.

**Ant Colony System:**

In 1996 Dorigo and Gambardella introduced the Ant Colony System (ACS) (Potvin, J.Y., Dube, D. and Robillard, C, 1996) which although strongly inspired by Ant System (AS), achieves performance improvements through the introduction of new mechanisms based on ideas not included in the original AS. Ant ACS differs from the other ACO instances due to its strategy of constructing an observation schedule. This strategy features two major changes to the rules employed in the AS algorithm, namely:

1. A new transition rule is introduced that favors either exploitation or exploration. From node i, the next node j in the route is selected by ant k, among the unvisited nodes $J^k$, according to the following transition rule which shows the probability of each city being visited:

   $$ P^k_i (t) = \begin{cases} 
   1 & \text{if } q \leq q_0 \text{ and } j = j' \\
   0 & \text{if } q \leq q_0 \text{ and } j \neq j' \\
   \frac{\tau^*_k (t) \eta^k_i (t) \gamma^j_q (t)}{\sum_{j' \in J} \tau^*_k (t) \eta^k_i (t) \gamma^{j'}_q (t)} & \text{Otherwise}
   \end{cases} \quad (1) $$

   Where
   
   $j' = \arg \max_{j \in J} \left( \tau_k (t)[\eta_i (t) \gamma^j_q (t)] \right)$ identifies the unvisited node in $J^k$ that maximizes $P^k_i (t)$.

   $\tau_k (t)$ is the amount of pheromone on the edge joining nodes i and j.

   $\eta_i (t)$ is the heuristic information for the ant visibility measure (e.g., defined as the reciprocal of the distance between node i and node j for the TSP).

   $\gamma^j_q (t)$ is the Savings algorithm proposed in 1964 by Clarke and Wright. the savings of combining any two customers i and j are computed as $\gamma_{i,j} = d_{i \rightarrow j} + d_{i \rightarrow j} - d_{i \rightarrow j}$

   $\alpha, \beta, \lambda$ are control parameters.

   $q$ is a uniformly distributed random number to determine the relative importance of exploitation versus exploration $q \in [0,1]$.

   $q_0$ is a threshold parameter and the smaller $q_0$ the higher the probability to make a random choice ($0 \leq q_0 \leq 1$).

2. The pheromone trial is updated in two different ways

   - Local updating: The aim of the local updating rule is to make better use of the pheromone information by dynamically changing the desirability of paths. Using this rule, ants will search in wide neighbourhood of the best previous schedule. As the ant moves between nodes i and j, it updates the amount of pheromone on the traversed edge using the following formula:

   $$ \tau_k (t + 1) = (1 - \rho) \tau_k (t) + \rho \tau_i \quad \text{if } \{\text{edge}(i, j) \in T_t\} \quad (2) $$

   Where

   $\tau_0$ is the initial pheromone level assumed to be a small positive constant distributed equally on all the paths of the network since the start of the survey. It is calculated as $\tau_0 = (nC)^{-1}$, n is the problem size (i.e., the number of nodes) and $C_i$ is the cost of the initial tour produced by a construction heuristic such as the Nearest Neighbor heuristic, and the evaporation rate,
\( \rho \) is a parameter in the range \([0, 1]\) that regulates the reduction of pheromone on the edges. The effect of local updating is that each time an ant traverses an edge \((i,j)\) its pheromone trail \( \tau_{ij} \) is reduced, so that edges becomes less desirable for the ants in future iterations. This encourages an increase in the exploration of edges that have not been visited yet. Local updating helps avoid poor stagnation situations.

- **Global updating:** When all ants have completed their schedule, the pheromone level is updated by applying the global updating rule only on the paths that belong to the best found schedule since the beginning as follows:

\[
\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho(1 / C_b) \quad \text{if } \{\text{edge}(i,j) \in T_b\}
\]

Where \( C_b \) is the cost of the best tour \( T_b \) found since the start of the algorithm. This rule is intended to provide a greater amount of pheromone on the paths of the best schedule, thus intensifying the search around this schedule. In other words, only the best ant that took the shortest route is allowed to deposit pheromone.

**Improvement of ACS:**

The pheromone updating formula was meant to simulate the change in the amount of pheromone due to both the addition of new pheromone deposited by ants on the visited edges, and the pheromone evaporation. In ACS the parameter \( \rho \) represents pheromone evaporation and deposited pheromone is discounted by a factor \( \rho \). It results in the new pheromone trail being a weighted average between the old pheromone value and the amount of pheromone deposited.

In the process of improving ACO, we found that using a trigonometric function in equation (4) gave better results than using \( \rho \). Moreover trigonometric function seems to be a good choice because its value is always in range \([0, 1]\). The change of evaporation rate is calculated using the following equation:

\[
\rho_c = 1 - \rho \cos\left(\frac{\pi t}{3n}\right)
\]

Where \( t \) represents the current time-step (iteration of the loop) and \( \rho_c \) is a small variable called the evaporation rate, which decreases with time.\( n \) are constants which depend on the size of problem and \( \rho = 0.99 \).

The basic idea of using this function in the pheromone global updating formula can be explained as follows: According equation (3), evaporation of the deposited pheromones will be done with coefficient \( 1 - \rho \). If we define \( 1 - \rho \) as \( x \), then \( x \) is a descending function of \( \rho \). At the start of the algorithm, the evaporation has less speed because the value of \( x \) is close to 1. Then \( x \) begins to decrease, leading to an increase in the speed of the evaporation. In fact increasing the evaporation rate is somehow like to scrubbing the old pheromones deposited by ants and highlighting the new pheromones on the edge. We can name this action, encouraging the new pheromone by eliminating some of the old pheromones deposited on the edge. We should increase encouragements for the new trails (i.e., eliminate more deposited pheromones by evaporating them) in later stages of the search. If this action happens in the early stages of iterations, generally may leads to the rapid emergence of a stagnation situation and premature convergence of the algorithm toward a suboptimal region and consequently rather poor results. So encouraging should be little during the early stages of the ACO algorithm when decisions are made more randomly and for achieving more acceleration, it must be increased across the time. This mechanism reflects ants' search experience truly and implements a useful form of forgetting and enables the algorithm to forget bad decisions previously taken through decreasing pheromone values on the edges trigonometrically in the number of iterations. In this way, ants search falls in the right route and qualified solutions will be obtained.

The vast literature amount literature on VRP nowadays tells us that a promising approach to obtaining high-quality solutions is to couple a local search algorithm with a mechanism to generate initial solutions (Rochat, Y., Taillard, E, 1995). A local search approach starts with an initial solution and searches within neighborhoods for better solutions. However, it may be trapped in local optima which are significant factor affecting the quality of solutions. In the proposed approach, the local search of the 3-opt algorithm is embedded in the presented algorithm to ameliorate the search performance. This algorithm that is shown below based on to omit three arcs of tour that are not neighborhood and connect them again in other method. It is noted that there are several routes for connecting route and producing tour again, but only some states are accepted that are satisfied in problem’s constraints. So, this new tour will be only accepted in state that first, above constraints are not violated specially about each vehicle’s capacity and second, novel tour will gain better value for problem than previous solution. It also should be noted that omitting three arcs and again connecting them are iterated until no improving 3-opt is found. When all of these \( n \) arrays are gained, the best of them is selected as the best
solution and algorithm will be finished. From simulation results, the proposed algorithm can improve the quality of solution for CVRP.

**Computational Results:**

A computational experiment has been conducted to compare the performance of the proposed algorithm with some of the best techniques designed for VRP. We executed the algorithm on some of the well-known problem instances from one dataset. Then, additional simulations were conducted over another dataset and results reported accordingly.

The algorithm were coded in Matlab 7 and, using a 1.81 GHz CPU. It was first applied to the 7 vehicle routing problems (VRPs) proposed by Christofides and can be downloaded from the OR-library (http://mistic.heig-vd.ch/taillard/problems.dir/vrp.dir/taillard.dir), and which have been widely used as benchmarks. The first 5 problems (i.e. C1 to C5) have customers that are randomly distributed with the depot in an approximately central location. In the last two problems (i.e. C11 and C12), the customers are grouped into clusters. Then we applied the algorithm to the twelve instances from the second dataset which proposed by Taillard and can be downloaded from (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/).

For tuning the parameters in our algorithm, there are four parameters including $\alpha, \beta, \lambda, q_0$. The ranges of four parameters were set to $\alpha \in \{1, 2, 3\}$, $\beta \in \{2, 4, 6\}$, $\lambda \in \{1, 2, 3\}$ and $q_0 \in \{0.9, 0.95, 0.99\}$. When tuning the parameters, the instance C1 was determined as the test problem. Then, the algorithm with each parameter combination for this instance was tested 10 times. Based on the gained results, the algorithm with the smaller weight parameter ($\alpha$) of pheromone trails possesses higher performance. This may be attributed to that in the proposed algorithm the initial pheromone trails are large values. If using the large control factor of pheromone trail, the effect of visibility value is weakened and results in a premature convergence. In addition, the qualities of the solutions of the algorithms with $\beta = 2$ are better than 4 and 6.

From the test results, it can be found that by setting the $\lambda$ to 2, the proposed algorithm can yield better solutions. Finally, the PA, in which $q_0$ is set to 0.99, can provide better solutions in compared with other values.

As a result, the pack of optional parameters obtained is as follows:

$$\alpha = 1, \beta = 2, \lambda = 2 \text{ and } q_0 = 0.99$$ (5)

Because the proposed approach is a meta-heuristic algorithm, the results for those two datasets are reported for twenty independent runs, and in each run the algorithm was executed until the best solution was iterated 10 times.

Table 1 illustrates the characteristics of the seven problem instances of Christofides et al. these problems range is size from $n=50$ to $n=199$ customers. All problems are Euclidean and distances are compared with real numbers. Columns 3-8 show the problem size $n$, the number of vehicles $m$, the vehicle capacity $Q$, the maximum route length $L$, the best known solutions BKS, the authors and the methods used by the authors.

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>n</th>
<th>Q</th>
<th>BKS</th>
<th>Reference</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>randomized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>50</td>
<td>160</td>
<td>524.61</td>
<td>Taillard(1993)</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>C2</td>
<td>75</td>
<td>0</td>
<td>140</td>
<td>835.26</td>
<td>Taillard(1993) Tabu Search</td>
</tr>
<tr>
<td>C3</td>
<td>100</td>
<td>200</td>
<td>826.14</td>
<td>Taillard(1993)</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>C4</td>
<td>150</td>
<td>2</td>
<td>200</td>
<td>1028.42</td>
<td>Taillard(1993) Tabu Search</td>
</tr>
<tr>
<td>C5</td>
<td>199</td>
<td>6</td>
<td>200</td>
<td>1291.45</td>
<td>Rochat and Taillard(1993) Tabu Search</td>
</tr>
<tr>
<td>grouped</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C11</td>
<td>120</td>
<td>200</td>
<td>1042.11</td>
<td>Taillard(1993)</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>C12</td>
<td>100</td>
<td>0</td>
<td>200</td>
<td>819.56</td>
<td>Taillard(1993) Tabu Search</td>
</tr>
</tbody>
</table>

Table 2 shows the comparison of our algorithm with published results. The first column describes the various instances, whereas the columns 2-6 specify well-known published best results obtained using meta-heuristic algorithm. Finally, the following column refers to the best result of our method for these instances. The proposed algorithm has shown to be competitive with the best existing methods in terms of solution quality. Moreover, during the experiment, some solutions are as close as the best solution published so far.
Table 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>524.61</td>
<td>524.61</td>
<td>524.61</td>
<td>524.61</td>
<td>524.61</td>
<td>524.61</td>
</tr>
<tr>
<td>C2</td>
<td>835.26</td>
<td>835.26</td>
<td>835.26</td>
<td>835.26</td>
<td>835.26</td>
<td>835.26</td>
</tr>
<tr>
<td>C3</td>
<td>826.14</td>
<td>829</td>
<td>826.14</td>
<td>829</td>
<td>830.14</td>
<td>829.45</td>
</tr>
<tr>
<td>C4</td>
<td>1031.07</td>
<td>1044</td>
<td>1028.42</td>
<td>1034</td>
<td>1038.20</td>
<td>1039.73</td>
</tr>
<tr>
<td>C5</td>
<td>1311.35</td>
<td>1334</td>
<td>1298.31</td>
<td>1327</td>
<td>1307.18</td>
<td>1298.31</td>
</tr>
<tr>
<td>C11</td>
<td>1042.11</td>
<td>1042.11</td>
<td>1042.11</td>
<td>1046</td>
<td>1044.12</td>
<td>1042.11</td>
</tr>
<tr>
<td>C12</td>
<td>819.56</td>
<td>819.56</td>
<td>819.56</td>
<td>819.56</td>
<td>824.31</td>
<td>819.56</td>
</tr>
</tbody>
</table>

TAG: tabu search by Gendreau (1994)
STO: simulate annealing and tabu search by Osman (1993)
TST: tabu search by Taillard (1993)
ASZ: ACO and scatter search by Zhang (2009)

Besides, Fig. 3 shows comparison between the gap values of the meta-heuristic algorithms, where the gap is defined as the percentage of deviation from the BKS in the literature. The gap is equal to \(100 \left( \frac{c(s^*) - c(s^t)}{c(s^t)} \right)\), where \(s^t\) is the best solution found by the algorithm for a given instance, and \(s^*\) is the overall best known solution for the same instance on the Web. A zero gap indicates that the best known solution is found by the algorithm.

Fig. 3: Computational results of the meta-heuristics

Table 3 shows the results obtained for the second problem instances and presents the comparison of best results of previous methods and our algorithm. The results show that our approach could find the best solutions for eight instances. Furthermore, the best, worst and average values over the twenty runs for each problem are shown in Table 3.

![Graph showing computational results of meta-heuristics](attachment:image.png)

### Table 3

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>Size</th>
<th>Previous best result</th>
<th>Our best result</th>
<th>worst</th>
<th>Average value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tai25a</td>
<td>75</td>
<td>1618</td>
<td>1618</td>
<td>1666</td>
<td>1684</td>
</tr>
<tr>
<td>Tai25b</td>
<td>75</td>
<td>1344</td>
<td>1344</td>
<td>1370</td>
<td>1362</td>
</tr>
<tr>
<td>Tai25c</td>
<td>75</td>
<td>1291</td>
<td>1291</td>
<td>1325</td>
<td>1311</td>
</tr>
<tr>
<td>Tai25d</td>
<td>75</td>
<td>1365</td>
<td>1365</td>
<td>1400</td>
<td>1384</td>
</tr>
<tr>
<td>Tai100a</td>
<td>100</td>
<td>2041</td>
<td>2101</td>
<td>2142</td>
<td>2131</td>
</tr>
<tr>
<td>Tai100b</td>
<td>100</td>
<td>1940</td>
<td>1940</td>
<td>1984</td>
<td>1965</td>
</tr>
<tr>
<td>Tai100c</td>
<td>100</td>
<td>1406</td>
<td>1410</td>
<td>1460</td>
<td>1445</td>
</tr>
<tr>
<td>Tai100d</td>
<td>100</td>
<td>1581</td>
<td>1585</td>
<td>1631</td>
<td>1612</td>
</tr>
<tr>
<td>Tai50a</td>
<td>150</td>
<td>3055</td>
<td>3055</td>
<td>3109</td>
<td>3095</td>
</tr>
<tr>
<td>Tai50b</td>
<td>150</td>
<td>2727</td>
<td>2727</td>
<td>2885</td>
<td>2795</td>
</tr>
<tr>
<td>Tai50c</td>
<td>150</td>
<td>2341</td>
<td>2355</td>
<td>2666</td>
<td>2531</td>
</tr>
<tr>
<td>Tai50d</td>
<td>150</td>
<td>2645</td>
<td>2625</td>
<td>2970</td>
<td>2899</td>
</tr>
</tbody>
</table>
Conclusion:

In this paper we presented a new ACS that is different with common ACS in the coefficient evaporation. At the end of every iteration, the hybrid algorithm also tries to improve both the performance of the algorithm and the quality of solutions for the VRP by using a local search algorithm in the name of 3-Opt. As a result, the intuition is that nodes which are near to each other will probably belong to the same group of the graph and thus will probably belong to the same route in VRP. Future work will be conducted to improve the proposed algorithm, and examined other local search algorithms to enhance the performance of the method for improving solutions. Additional improvements might lie on the combination of our approach with the other meta-heuristic like genetic algorithm, tabu search, simulated annealing and so on.

REFERENCES

http://mistic.heig-vd.ch/taillard/problems.dir/vrp.dir/taillard.dir
http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/


