

Improved Ant System Algorithm and its Application for Vehicle Routing Problem

Lulu Zhai^{1,a}, Iqbal Muhammad Ather^{1,b}, Zhijie Wang^{1,c}, Qianqian Zheng¹

¹ College of Information Science and Technology, Donghua University, Shanghai, 201620, China.

^aemail:2141145@mail.dhu.edu.cn, ^bemail: 415030@mail.dhu.edu.cn, ^cemail: wangzj@dhu.edu.cn

Keywords: ACS; C-W; VNS; CVRP

Abstract. The Ant Colony System combined with Variable Neighborhood Search (VNS-ACS) was proposed to relieve the limitations of typical Ant Colony Algorithm (ACA) vulnerable to stagnation and random results. Based on the idea of ACS algorithm, VNS-ACS algorithm improves the path search strategy and the local path optimization by combining with Clark-Wright saving algorithm (C-W), roulette algorithm, VNS and segment setting parameters. Accordingly it can make deterministic and stochastic equilibrium, consequently improving the global optimization ability of the algorithm, convergence speed and stability. Several experimental results in the vehicle routing problem are presented to demonstrate its advantages over ACS in the optimal path accuracy and reliability of the algorithm.

1. Introduction

In 1959, Vehicle Routing Problem (VRP) was initially introduced by Dantzig and Ramser [1]. In 1964, Clark and Wright [2] proposed an effective heuristic algorithms Clark-Wright saving algorithm for VRP. Then VRP has become the frontier and hot topics in the field of Operations Research [3]. Up to 20% of the total cost of transportation can be saved with rational planning of transport routes. This paper mainly focuses on the Capacitated Vehicle Routing Problem (CVRP). The VRP belongs to the category of NP Hard problems, meaning that the computational effort required to solve this problem increases exponentially with the problem size.

The solution techniques to the VRP can be categorized as *exact approaches* or *heuristics*. For example, there is *Branch and bound* (Fisher 1994 [4]) that proposed to find every possible solution until one of the best is achieved in the exact approaches. There are *constructive methods* [3] and *2-phase algorithm* [5] in the heuristics, which perform a comparatively limited exploration of the search space and typically produce high quality solutions within reasonable computing times.

In the past three decades, great efforts have been made on the development of algorithms based on *metaheuristics* which perform a deep exploration of the most promising regions of the solution space and produce solutions of higher quality than the classical heuristics, mainly using two principles: local search and population search. The local search, which is an intensive exploration of the solution space, consists of *tabu search* (TS) and *simulated annealing* (SA)[6]. Hansen and Maladenovic put forward a novel track heuristic algorithm, namely *Variable Neighborhood Search* algorithm (VNS). This algorithm is simple and easy to combine with other algorithms. To elaborate further, a large number of experiments show that VNS can find the approximate optimal solution to the combinatorial optimization problem in a modest time and can be used to local search [7]. Population search consists of *genetic algorithm* (GA) [8] that recombines the good solution. GA is vulnerable to premature and convergence. The global solution of TS is poor and SA searches very slowly. The other classical population search is *ant colony system* (ACS) [9]. By comparison, Bernd Bullnheimer proved that ACS can produce more competitive results than other metaheuristic algorithms[10]. However, the ACS is vulnerable to stagnation and random results.

In this paper, for the sake of improving the solution to the CVRP quality, we present and illustrate a highly efficient metaheuristic algorithm called VNS-ACS, combining VNS with ACS algorithm and overcoming the defect of ACS. It shows that the ant paradigm can produce competitive results, meanwhile improve the stability of results at a fraction of the computational

cost of the best approaches. The remainder of the paper is organized as follows. First, we briefly describe the CVRP and the two basic ant system phases, construction of vehicle routes in section 2.1 and pheromone update rule in section 2.2. Second, new algorithm is introduced to improve the solution quality in section 3. The computational experiments in section 4 compare our VNS-ACS to ACS for the VRP *Solomon Instances* in the paper. Finally, section 5 concludes the paper.

2. Ant Colony System for CVRPs

The formal description for the CVRP is as follows: suppose that a common depot v_0 has a fixed fleet of delivery vehicles of uniform capacity Q that must service known customer demands for a single commodity at minimum transit cost, namely designing a set of delivery routes that satisfy constraints:

- i. Every customer is visited exactly once by exactly one vehicle.
- ii. Each route starts and ends at the depot.
- iii. The total demand of each route does not exceed vehicle capacity Q .

M. Dorigo et al proposed a modified Ant Colony System (ACS)[9], which achieves a better balance between exploiting the existing routes and exploring the new path with the adjustment on *state transition rule* and *pheromone updating strategy* which eventually improves the global solution. Usually VRP tour optimization is divided into two parts: the first constructing routes by using ACS, and then optimizing tour by using the *2-opt* algorithm.

2.1 Construction of vehicle routes

In the VRP ant colony system, a route is that k th ant sets out from v_0 and successively selects customer v_i to visit and return to v_0 before it exceeds the capacity. All routes guarantee that all customers are served only once. The construction of vehicle routes is done as follows: first put the unvisited customers into the candidate list cl_k , then put the visited customers into the tabu list $tabu_k$, k th ant selects next customer from cl_k according to a *pseudo-random-proportional rule*[9] that can be stated by eqs.(1)-(2).

$$j = \begin{cases} \arg \max_{s \in cl_k} \{ [\tau_{ij}]^\alpha [\eta_{ij}]^\beta \}, & \text{if } q \leq q_0 \text{ (exploitation)} \\ J, & \text{otherwise (exploration)} \end{cases} \quad (1)$$

$$J = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{u \notin tabu_k} [\tau_{iu}]^\alpha [\eta_{iu}]^\beta}, & j \notin tabu_k \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Where τ_{ij} is the pheromone left by other ants, d_{ij} is the distance between customer i and customer j , η_{ij} is heuristic function which evaluates the utility of move customer j from the current customer i . α is a parameter which determines the importance of pheromone and β is a parameter which determines the importance of distance. q is a value chosen randomly with uniform probability with an interval $[0, 1]$, $q_0 (q_0 \in [0,1])$ is a parameter which determines the relative importance of exploitation versus exploration in eq.(1), namely the probability of random selection. J is a probability distribution function given by eq.(2) and we choose next customer j from current customer i according to J and roulette algorithm[11].

2.2 Pheromone updating rule

2.2.1 Local updating (Offline update by single ant)

After sth ant has visited all customers and built a tour, pheromone is updated on all edges according to eqs.(3)-(4) in order to reduce the probability that next ant repeats tour and expand the search space, thereby avoid local optimal solution.

$$\tau_{ij}(s+1) = (1 - \rho)\tau_{ij}(s) + \rho\tau_0 \quad (3)$$

$$\tau_0 = (nL_m)^{-1} \quad (4)$$

Where $L_m = \min\{d_{ij}\}$ and τ_0 is the initial pheromone concentration of each path. ρ is the local pheromone evaporation factor and also is negative feedback factor.

2.2.2 Global updating (Offline update by ant colony)

In ant system, the global updating rule is implemented by eqs.(5)-(6). After all ants have built their tours, we only reinforce the pheromone of the shortest tour achieved so far and the pheromone of other tours gradually volatilize so as to make the later search process more guided.

$$\tau_{ij}(k) = (1 - \mu)\tau_{ij}(k-1) + \mu\Delta\tau_{ij}(k-1) \quad (5)$$

$$\Delta\tau_{ij} = \begin{cases} Q^*(L_{gb})^{-1}, & \text{if } (i, j) \in \text{global - best - tour} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where Q is pheromone intensity, L_{gb} is the globally best tour so far and μ is global pheromone evaporation factor and also is positive feedback factor ($\mu \in [0, 1]$). $\Delta\tau_{ij}$ is the pheromone increment on the edge (i, j) of this iteration.

2.3 Route improvement strategies

Mixed in the ant colony algorithm local optimization algorithm to improve each constructed solution can further shorten the length of the route, thus speeding up the convergence rate of ant colony algorithm. After all ants have constructed the tours completely, optimize the shortest tour based on local optimization, then achieve a new tour and update pheromone of this new tour. The *construction of vehicle routes, pheromone update and route improvement strategies*, above three steps are repeated for a given number of iterations to find the optimal tour. The route improvement strategy is that it typically adopts *2-opt heuristic* [12] to exchange the customer locations of the tour after achieving the shortest tour by ACS. Although this algorithm can achieve the best solution, the result is very random and unreliable. To improve the robustness, we propose the VNS-ACS algorithm, described in the following section.

3. VNS-ACS Algorithm

The VNS-ACS algorithm is the combination of VNS and ACS to improve the robustness. It removes the local search routine from the basic scheme of the VNS algorithm and applies the ACS to produce better solution. Firstly, improve the route search strategy of the ACS, secondly construct the initial solution, and finally local optimization of initial solution by VNS algorithm. VNS provides a higher quality solution for ACS. It guides the ACS to construct new solutions and expand the exploration of space. As a result, the number of iterations of the algorithm improves the stability and accuracy of the optimal solution.

3.1 Route search strategy

Since the start and end of every customers route are the depot, we can combine ACS with the idea of saving algorithm [2] according to the eq.(7) where s_{ij} is the saving factor. The greater the saving factor, the greater the probability that the edge is selected, therefore the shorter the total length of the path. Adding the saving factor according to the eqs.(8)-(9) could make the global solution better.

$$s_{ij} = s_{ji} = d_{i0} + d_{j0} - d_{ij} \quad (7)$$

$$j = \begin{cases} \operatorname{argmax}_{s \in cl_k} \{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta [s_{ij}]^\gamma\}, & \text{if } q \leq q_0 \\ J & \text{otherwise} \end{cases} \quad (8)$$

$$J = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta [s_{ij}]^\gamma}{\sum_{u \notin tabu_k} [\tau_{iu}]^\alpha [\eta_{iu}]^\beta [s_{iu}]^\gamma} & j \notin tabu_k \\ 0 & otherwise \end{cases} \quad (9)$$

When roulette random selection is empty, we randomly select one customer j from the cl_k and judge whether the new total weight of this route is overweight or not. If it is overweight, the vehicle returns to the depot. If not, add j to the $tabu_k$.

3.2 Parameter Settings

Generally, the number of ants in the ACS is two-thirds of customers. There are 7 main parameters affecting the performance of the algorithm, respectively as α , β , γ , q_0 , ρ , μ , Q . Among them, the greater the value of α and Q , the better the positive feedback, thus strengthening the collaboration and is vulnerable to be premature. The greater the value of β , the easier that ant search neighborhood customers to find the shortest routes but can lead to local optimization. The greater the value of γ , the better the global searching ability. With the increment in the value of α and β , convergence abilities are improved, the search space is decreased and randomness weaken that may lead the algorithm into local optimum. The smaller the value of μ , the random and global searching ability of the algorithm is enhanced, but the convergence speed is reduced. In the initial stage of the algorithm the speed of finding the best solution is very fast as proposed by Luca Maria Gambardella [13] in the ACS's classic convergence curve. The new optimal solution appears after 50% of the total number of iterations, but pheromone of stagnation emerges [13].

According to the above analysis we set the parameters by making subsections. In the initial stage, parameters setting is small in order to expand the search space, that is, $\alpha=0.8$, $\beta=2$, $\gamma=3$, $\rho=0.9$, $\mu=0.9$, $Q=10$, $q_0=0.01$. When the length of the best solution changes slightly, parameters increase in order to reduce the search space and adjust the pheromone of routes and ensure its convergence to the global optimum solution, that is $\alpha=1$, $\beta=4$, $\gamma=4$, $\rho=0.1$, $\mu=0.1$, $Q=10$, $q_0=0.2$.

3.3 VNS local path optimization

The variable neighborhood search procedure (VNS) guides local search and escape local minima. VNS mainly adopts local operations, namely *or-opt* neighborhood, *swap* neighborhood, *2-opt** neighborhood and *cross* the neighborhood[14], and combines them randomly to expand the search space and increase the optimization speed and higher convergence ratio. Apply the VNS to optimize the initial solution generated by ACS, compare the length of solution optimized with the best solution achieved so far and update the global pheromone of the shorter solution. After a given number of iterations, we can get the best solution.

3.4 Algorithm implementation

Step 1: Initialization parameters, the number of iteration N_{max} , the number of ants m , the currency iteration time $N = 0$, the length of best solution $L_{gb} = \infty$. Input data and calculate d_{ij} , s_{ij} , τ_0 according to eq.(3), then store in the respective array.

Step 2: Put all ants in the depot, $tabu_k=1$, the l th ant construct tour according to eqs.(8)-(9) and load, then update $tabu_k=1$ and local pheromone until all customers are visited.

Step 3: Repeat Step 2 until all ants visit all customers. Apply the VNS to adjust the construction of the shortest tour in this iteration so that it shortens the tour, if the length of new tour is shorter than the length of best solution as a result replace it and update its pheromone and L_{gb} . Otherwise, update the pheromone of the best solution.

Step 4: $N = N + 1$, if $N < N_{max}$, clear $tabu_k$ and load and jump to Step2 and Step 3. Else iteration ends and input the best solution.

4. Experiment simulation and results analysis

4.1 The effectiveness of the algorithm

In this section, we verify the effectiveness of VNS-ACS algorithm by the instances applied: Solomon Instance, such as *eil22*, *eil30*, *r101*, *rc101*. The running environment is in the 64 bit Window8.1 system, Intel (R) Core (TM) i7-4700CPU@3.60GH processor, Matlab 2012a. In order to verify effectiveness and robustness, we compared ACS with VNS-ACS. The parameters of VNS-ACS are as follows. $N_{max}=50$, the section node is a quarter of N_{max} , other parameters are set according to section 3.2. The parameters of ACS are, $\alpha=1$, $\beta=4$, $\rho=0.5$, $\mu=0.1$, $Q=10$, $q_0=0.2$, $N_{max}=100$. Simulate the above cases 1000 times with both algorithms. Table 1 and 2 show the results, where t is program running time for each case, ε is error and r is robustness, which are given by eqs.(10)-(11).

$$\varepsilon = (best - ave) / best * 100\% \quad (10)$$

$$r = m / n * 100\% \quad (11)$$

In the formula, $best$ is the optimal length and ave is the average length of iteration. m is the number of times finding the optimal solution and n is equal to 1000.

Tab 1 The simulation result of VNS_ACS

case	best(km)	ave(km)	$\varepsilon(\%)$	r(%)	t(s)
eil30	505.0111	505.0111	0	100	10
eil22	375.2798	375.2798	0	100	4
r101	335.2688	335.3107	0.01	90	6
rc101	294.9944	294.9944	0	100	4

Tab 2 The simulation result of ACS

case	best(km)	ave(km)	$\varepsilon(\%)$	r(%)	t(s)
eil30	505.0111	514.1918	1.8	0.8	5
eil22	375.2798	377.5934	0.6	18.5	3
r101	335.2688	336.5712	0.38	26	3.5
rc101	294.9944	295.0237	0.01	96.6	3.5

4.2 Results analysis

Compare Table 1 with Table 2, shows that the robustness(r) of the new algorithm is better, error(ε) is smaller and reliability is higher.

From the Fig.1, the length of optimal solution generated by VNS-ACS is superior to reference [16], the accuracy of optimal solution improves 3%, consequently the convergence is improved.

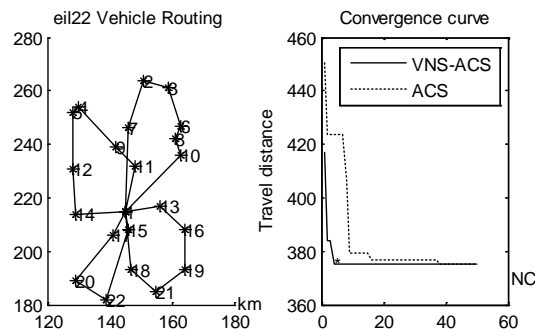


Fig.1. Experimental Result of *eil22* case

5. Discussion and Conclusions

ACS algorithm is vulnerable to stagnation and produces random results. In this paper, we apply VNS to guide ACS to search routes in the more larger space and presented VNS-ACS a novel approach to CVRP, which overcome the shortcomings of ACS. Although the results presented in

this paper show the robustness of VNS-ACS is better than the ACS and the error rate is smaller. However, the calculation time increases. VNS-ACS is the foundation of solving the large scale VRP. In the future we will research how to shorten the calculation time and solve the VRP.

Acknowledgement

In this paper, the research was supported by the National Natural Science Foundation Project (Project No.11572084).

References

- [1] Dantzig GB, Ramser JH, The truck dispatching problem[J], *MngtSci.* 1995 (6) 80-91.
- [2] Clarke G, Wright J, Scheduling of vehicles from central depot to a number of delivery points[J], *Operations Research.* 1964 (12) 568-581.
- [3] Paolo Toth, Daniele Vigo, The Vehicle Routing Problem[M], Society for Industrial and Applied Mathematics philadelphia. 2002.
- [4] Fisher ML, Vehicle routing. In: Ball MO, Magnanti TL, Monma CL and Nemhauser GL(eds)[J], *Network Routing , Handbooks in Operations Research and Management Science.* 1994 (8) 1-33.
- [5] J. Renaud, F.F. Boctor, A Sweep-Based Algorithm for the Fleet Size and Mix Vehicle Routing Problem[J], *European Journal of Operational Research.* 2002: (140): 618-628.
- [6] Osman I H. Meta-strategy simulated annealing an Tabu search algorithms for the vehicle routing problem[J], *Annals of Operations Research.* 1993: 41(4): 421-451.
- [7] P Hansen, N Mladenovic, Variable neighborhood search: Principles and applications[J], *European Journal of Operational Research.* 2001: 130(3): 449-467.
- [8] W Wang, Z Wang, F Qiao, An Improved Genetic Algorithm for Vehicle Routing Problem with Time-Window[J], *International Conference on Intelligent Computing & Cognitive Informatics.* 2009: 8(6): 203-306.
- [9] M Dorigo, LM Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem[J], *IEEE Transactions on Evolutionary Computation.* 1997: 1(1): 53-66.
- [10] Bernd Bullnheimer, Richard F, Hartl and Christine Strauss, An improved ant system algorithm for the vehicle routing problem[J], *Annals of Operations Research.* 1999:(89): 319-328.
- [11] A Lipowski, D Lipowska, Roulette-wheel selection via stochastic acceptance[J], *Chemometrics and Intelligent Laboratory Systems.* 2014 (133) 136-148.
- [12] MARTINOVIC,G. , BAJER,D. Elitist Ant System with 2-opt Local Search for the Traveling Salesman Problem[J], *Advances in Electrical and Computer Engineering.* 1996: (96): 622-627.
- [13] Luca Maria Gambardella , Marco Dorigo. Solving symmetric and asymmetric TSPs by ant colonies[J], *Proc. IEEE Int. Conf. Evolutionary Computation.* 2012 (12) 25-32.
- [14] Behnam Vahdani, Reza Tavakkoli-Moghaddam, Mostafa Zandieh, et al. Vehicle routing scheduling using an enhanced hybrid optimization approach[J], *Journal of Intelligent Manufacturing.* 2012: 23(3): 759-774.
- [15] Glover F, Gary A Kochenberger. *Handbook of metaheuristics*[M]. Norwell Academic Publisher. 2003.
- [16] Tang Liansheng, Cheng Wenming, Zhang Zeqiang et al. Vehicle routing simulation based on an improved ant colony algorithm[J], *Comptuter Simulation.* 2007: 24(04): 262-264.