AIS-based Algorithm for Solving Vehicle Routing Problem with Simultaneous Pick-up and Delivery (VRP-SPD)

Chuhang Yu and Henry Y. K. Lau
University of Hong Kong, Hong Kong, China
Email: {ych1102, hyklau}@hku.hk

***Abstract*—Vehicle Routing Problem with Simultaneous Pick-up and Delivery (VRP-SPD) is regarded as an NP-hard problem, which takes unacceptable time to use traditional algorithms to solve. This article presents an artificial immune systems or AIS-based algorithm to solve the problem and the results shows competitive performance. This algorithm is embedded with a sweep approach to generate random initial population. For the mutation operator, a variety of local search techniques are applied to realize the diversity. The algorithm is tested with examples which used by many of other works and compared with the results obtained by an exact algorithm. Experimental results suggest that the algorithm is a valuable alternative to other metaheuristics for solving VRP-SPD.**

***Index Terms*—Artificial immune systems, pickup and delivery, vehicle routing**

I. Introduction

As an extension to classical capacitated vehicle routing problem (CVRP/VRP), a new branch of vehicle routing problem known as vehicle routing with simultaneous pick-up and delivery service (VRP_SPD) has attracted increasing attention in the research community in recent years. It is first introduced in 1989 by Min [1]. An exact algorithm based on the branch and price method is proposed by Dell'Amico [2].

The objective of VRP_SPD is to minimize the total distance traveled to serve all customers without violating the vehicle capacity. Some general constraints include [3]: Every route should start and end with the same depot v_0; Each customer can be and only be visited once, by exactly one route; All the delivery quantities are transported from depot and to the customers; all the pickup quantities are from customers and to depot; At no customer point can the total quantity on the vehicle exceed the capacity Q.

VRP_SPD is known to be an NP-hard combinatorial optimization problem. The analytical methods may take unacceptable time to solve. Therefore, many meta-heuristic algorithms are put forwards to solve the problem. The insertion heuristics are adopted by Salhi [4] to solve both VRP with backhauling and SPD. A modified insertion heuristic is used by Dethloff [5] through incorporating additional penalty or bonus into insertion criterion. The parallel heuristic based on sequential heuristic proposed by Subramanian [6] consists of various local search techniques and a variable neighborhood descent procedure. Population-based heuristics, due to their strong exploration ability, are also used by previous works relating to VRP_SPD. An ant colony system (ACS) algorithm is used to solve VRP_SPD by Gajpal [7]. An particle swarm optimization (PSO) for VRP_SPD is firstly proposed by Kachitvichyanukul [8]. A genetic algorithm with partial-mapped crossover and swap mutation is proposed by Serdar [9] that aimed to introduce an alternative approach to solve VRP_SPD.

Artificial immune system (AIS) is an engineering analogy of human biological immune systems. Emerging in the 1990s as a new branch of Computational Intelligence, AIS aims at finding solutions to NP-hard problems with high efficiency in terms of computation effort and time. The application of AIS to solving optimization problems has received notable success [10]. However, works related to vehicle routing is rather few [11], not to mention the more specific problem of VRP_SPD. Two applications of AIS in VRP are referred to in work of Potvin [11].

This article aims to fill this gap by proposing an AIS-based algorithm to solve the VRP_SPD problem by presenting its capability for solving such problems efficiently. The proposed algorithm is primarily inspired by the Clonal selection principles and Immune network theory, while implanting a sweeping approach for generating initial solutions, and providing local search techniques for the mutation processes. The advantages of proposed AIS-algorithm introduced in this article include: 1) diversity through mutation operator; 2) exploitation through memory mechanism.

II. The Representation of the Problem Under an Artificial Immune Systems Framework

The antigen is what invades vertebrates and arouses immune reaction. In VRP_SPD, the information about customers and vehicles is regarded as epitopes of antigens. One antigen may have one or more epitopes.

Manuscript received January 5, 2013; revised January 30, 2013; accepted February 28, 2013.

doi: 10.12720/jtle.1.2.174-178

Antibodies are produced by the immune system to eliminate those antigens. The epitopes is recognized by antibodies through the receptors on the surface of B-cells. The B cell and its receptor are called antibody. In our problem domain, an antibody is a solution of the VRP_SPD problem and it represents a set of feasible routes.

Activation represented by the computation of affinity is obtained by calculating objective values, which include three parts: route total distance, capacity violation, and vehicle number. The antibody with a higher affinity is selected while those with lower affinities are suppressed. The selected antibodies will proliferate for further differentiation.

III. Implementation of an Artificial Immune Systems-Based Algorithm

The proposed algorithm is developed based on the Clonal Selection principle and incorporates the salient features of immune network suppression and diversity.

Experiments are conducted to solve different scale VPR_SPD problems. The objective is to obtained solutions with minimum travel distance subjected to vehicle capacity constraint at each customer.

A. Initial Solution Construction and the Mutation Operators

1) The sweep approach

The proposed algorithm adopts a "cluster and route" method to randomly construct the initial solution. The sweep approach of Gillett and Miller [12] is adopted to cluster the customers: cluster all customers into VN (VN is not decided until all customers are packed, and may varies among different initial solutions) groups and the algorithm forms the initial antibody by the following mechanisms: Firstly sort all the customers according to increasing order of polar angle; Then randomly select one customer as the starting point; Add each customer one by one into a cluster; If addition of a new customer exceeds the vehicle capacity (denoted by Q), either the total delivery demand or total pick-up demand, generates a new cluster and starts with the new customer; After that, create VN new routes, each of which contains no customers but only depot twice (origin and ending).

Customers grouped in each cluster above are inserted randomly into one of these routes (tolerating temporary the capacity violation, if exists).

An antibody is then generated composed of VN routes.

2) Mutation operators

This algorithm adopts 6 inter-route local search methods and 4 intra-route local search methods which utilizes a random neighborhood ordering (RVND) as the mutation operator. They are Shift(1,0), Swap(1,1), Shift(2,0), Swap(2,1), Swap(2,2), Cross, Opt, 2-opt, Permutation and Reverse. Details of these techniques could be found in [6]. It is showed in [6] that RVND leads to better results when compared to deterministic method in general. Each method is randomly selected for each clone. Routes are arbitrarily chosen in each solution.

B. AIS for VRP_SPD

1) Population Initialization

The process described in A .1) will be iterated for N times so that the initial population size is N .

2) Activation

Affinity for each antibody is computed. Affinity consists of three parts: total distance of all routes L .

$$L = \sum c_{ij} x_{ij} \ , i, j \in V \tag{1}$$

$x_{ij} = 1$ if edge (i, j) is included in one route, $x_{ij} = 0$ otherwise . c_{ij} is the distance of edge (i, j) .

Capacity violation accumulated in the route CV (calculated by accumulation of capacity violation quantity at each point)

$$CV = \sum_{i \in V} \max[(q_i - d_i + p_i) - Q, 0] \tag{2}$$

In which q_i is the total load on vehicle before it arrives customer i . d_i and p_i are the delivery demand and pickup demand of customer i , respectively and vehicle number VN .

3) Selection

Randomly select n antibodies as the Best population according to affinity ranking. Using lexicological sorting method containing the three elements as described before: CV , L , and VN . The capacity violation CV has the highest priority, hence, we find the solutions with no or as less capacity violation as possible.

4) Clone

Every time an antibody is chosen randomly from the Initial population for cloning; antibodies with higher affinity tend to have a higher probability to be cloned and the number of cloned offspring will be larger. Antibodies are iteratively and randomly selected in this way and compose Clone population whose size is 6 times larger than the initial population. That means, the random selection process should be repeated by 600 times.

5) Mutation

Hypermutation is operating on the Cloned population. Equal mutation frequency is assigned to each cloned antibody. This is proved to result in better solutions, when compared with inversely proportional to affinity mutation rate, through trials with Dell'Amico Data Set, and thus adopted in both data sets.

6) Suppression

The top n unique individuals from the Mutate population as well as the antibodies from Best population compose the new Best Population, whose size is $2n$. Affinity II: $Aff_2 = 1 / L$ (to simplify the process, the affinity II is only inverse proportional to the total distance L) is used for removing the same antibodies from the Best population. Memory population is updated here by adding new Best Population. Before Memory population is full (the size is set as the same as Initial population), add all the antibodies from the Best Population to the Memory population if they are different to those of the Memory population; otherwise remove the antibodies with lower affinities so as to maintain the population size of N.

7) Recruitment

The Mutate population size is shrunk to N by removing the antibodies with lower affinities. The Initial population is updated as the same of the Mutate population.

8) Iteration

All Populations are initialized except for Initial Population and Memory Population. Processes 2) to 8) are repeated until the termination condition (e.g., the predefined number of iterations) is reached.

IV. Implementation And Results

The proposed algorithm was implemented in Java. To evaluate the effectiveness of proposed algorithm, two benchmark data sets that have also been adopted by several previous literatures were used in the experimentation. In the following sections, these two benchmark data sets are introduced and corresponding computational results are presented.

A. Benchmark Data Sets

The two data sets adopted have different sizes and characteristics, which provide a representative test data for benchmarking the performance of the proposed algorithm.

1) Dell'Amico Data Set

The first adopted benchmark data set is introduced by Dell'Amico [2]. The test instances are adapted from Solomon's instances R101, C101, and RC101. The scale of the problems presented is relatively small. Only 20 and 40 customers are considered in the data. Here, vehicle capacity is set to 100. The delivery demands are the same as original demands, while pickup demands are generated according to the function $\lfloor (1-r)*demand \rfloor$ for even customers and $\lfloor (1+r)*demand \rfloor$ for odd customers. The values of $\gamma = 0.2$ and $\gamma = 0.8$ are considered.

2) Nagy and Salhi Data Set

The second data set is from Nagy and Salhi [4, 13]. The problems are originally proposed by Christofides et al.[14] for CVRP. Nagy and Salhi modified the problems to become a VRP_SPD according to the followings: for each customer (city) a, first calculates a ratio $\gamma_a = \min(x_a / y_a, y_a / x_a)$, where x and y are coordinates. Then the delivery demand of a is computed by $d_a = \gamma_a * demand$, the pick-up demand is $p_a = (1-\gamma_a)*demand$, where $demand$ is original demand in CVRP data set. The first seven VRP_SPD problems (denoted by CMT1X, CMT2X, CMT3X, CMT4X, CMT5X, CMT11X, CMT12X) are constructed as aforementioned. They are adopted in this experiment since they are commonly referenced by other analogous works. The other seven problems (denoted by CMT1Y, CMT2Y, CMT3Y, CMT4Y, CMT5Y, CMT11Y, CMT12Y) are generated by swapping d and p values for every customer (city). All swapped d and p values are rounded to the nearest integer.

B. Computational Results

1) Dell'Amico data set instances

The population size N is set to 100 and Best population size n is set to 10. The results are presented in Table 1. The results from the first data set are compared with the results given by Dell'Amico [2] and the average results are compared with those generated by the particle swarm optimization (PSO) algorithm proposed by Ai and Kachitvichyanukul [8].

For example, C101_40_0.2 represents the data of choosing the first 40 customers from Solomon's C101 with $\gamma = 0.2$. The number of iteration performed in the experiment is 5000. 30 trials were run for each experiment. Exact (UB) means the particular upper bound is obtained using branch and bound algorithm. This data set consists of 12 instances.

TABLE I. Comparison of AIS, Dell'Amico et al. and PSO

Instance	No. vehicle	AIS		Exact(UB)	%Deviation	PSO
		Average	Best			
C101_20_0.2	4	265	265	272	-0.02574	
C101_40_0.2	8	547	541	551	-0.00726	
R101_20_0.2	3	317	317	329	-0.03647	
R101_40_0.2	7	578	577	601	-0.03827	
RC101_20_0.2	5	419	419	428	-0.02103	
RC101_40_0.2	9	874	870	886	-0.01354	
C101_20_0.8	4	271	271	279	-0.02867	
C101_40_0.8	8	566	557	569	-0.00527	
R101_20_0.8	4	326	326	342	-0.04678	
R101_40_0.8	8	612	600	629	-0.02703	
RC101_20_0.8	5	453	450	458	-0.01092	
RC101_40_0.8	9	918	910	926	-0.00864	
average		512.2	509	522.5	-0.01978	524.7

In order to compare the solution obtained with other solutions, the percentage of deviation from previous solution is used, which is calculated by:

$$\%Deviation = \frac{z - z_p}{z_p} * 100\% \tag{3}$$

where Z is the values obtained by AIS and Zp is the values obtained by other methods.

The number of customers in this instance is relatively small. Table 1 shows the proposed AIS-based algorithm performs well in solving VRP_SPD when compared to the other methods. Each test instance is run for 30 independent with small negative percentages of deviations, which means the obtained results are below the upper bound of Dell'Amico. Average result of particle swarm optimization (PSO) algorithm is also used for comparison. The results obtained by AIS-based algorithm are a little better than those of PSO.

2) Nagy and Salhi instances

The size of this instance is relatively larger. The results are showed in Table 2. The performance is not as good as in the case of Dell'Amico when compared with existing literatures.

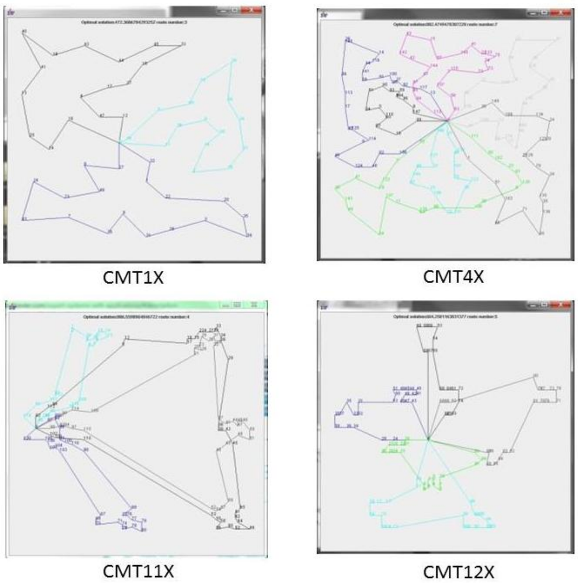

Figure 1. Vehicle routes illustration of obtained results.

In Figure 1, the different color lines represent different vehicle routes. As the distance between customers are calculated by the formulation: $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, i and j are two different customers, x and y are their coordination.

Their Euclidean distance subject to triangle inequality.therefore fewer intersections between the routes imply better routes with shorter total distance.

From Fig. 1, we can find that the proposed algorithm works better for the problem with fewer customers, such as CMT1X. There are no intersections in CMT1X. When the problem size is larger than 100 (CMT12X with 100 customers, CMT11X with 120 customers, CMT4X with 150 customers), there are more intersection within and among the routes.

The algorithms used for comparison in Table 2 are some other meta-heuristics, which are large neighborhood search (LNS), ant colony system (ACS), iterated local search (ILS), and particle swarm optimization (PSO). The deviation is obtained by comparing with the average values of other meta-heuristics. From Table 2, we can see AIS can also produce competitive results as good as other meta-heuristics. Nearly half of tests produced better results than the average of other metaheuristics.

TABLE II. COMPARISON OF AIS AND OTHER META-HEURISTICS SOLVING NAGY AND SALHI INSTANCES

instance	Other Meta-Heuristics*				dev
	min	max	average	distance	
CMT1X	466.8	467	466.9	472	0.011005
CMT1Y	466.8	467	466.9	472	0.011005
CMT2X	684.2	707	692.7	682	-0.01548
CMT2Y	684.2	709	689.5	688	-0.00213
CMT3X	721.3	742	727.4	728	0.000806
CMT3Y	721.3	739	728.2	728	-0.00029
CMT12X	662.2	685	670.9	664	-0.01027
CMT12Y	622.2	681	660.8	671	0.015454
CMT11X	835.3	898	849.9	886	0.042522
CMT11Y	833.9	920	868	877	0.010371
CMT4X	852.5	923	872.3	883	0.012287
CMT4Y	852.5	920	871.3	867	-0.00498
CMT5X	1029.3	1150	1070.5	1081	0.009776
CMT5Y	1029.3	1146	1076.4	1073	-0.00311
average deviation					0.005497

*Reference from [6-8, 15, 16]

V. CONCLUSION AND FURTHER STUDY

An AIS-based algorithm for solving VRP_SPD is presented in this article. Two commonly used instances on VRP_SPD are tested with the algorithm and competitive results compared with other two methods are obtained. The mutation operator adopted enables the algorithm to find the good solutions with the memory operator retaining these good solutions for deep exploration.

While the experimental results on problems with relatively small sized are satisfactory, the scalability and the performance on large-scale problems still required further research. Further study on the mutation rate and diversity of solution is required to improve the performance.

REFERENCES

[1] H. Min, "The multiple vehicle routing problem with simultaneous delivery and pick-up points," *Transportation Research Part A: General*, vol. 23, no. 5, pp. 377-386, 1989.

[2] M. Dell'Amico, G. Righini, and M. Salani, "A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection," *Transportation Science*, pp. 235-247, 2006.

[3] N. Bianchessi and G. Righini, "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery," *Computers & Operations Research*, vol. 34, no. 2, pp. 578-594, 2007.

[4] S. Salhi and G. Nagy, "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling," *Journal of the Operational Research Society*, pp. 1034-1042, 1999.

[5] J. Dethloff, "Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up," *OR Spectrum*, vol. 23, no. 1, pp. 79-96, 2001.

[6] A. Subramanian, *et al.*, "A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery," *Computers & Operations Research*, vol. 37, no. 11, pp. 1899-1911, 2010.

[7] Y. Gajpal and P. Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup," *Computers & Operations Research*, vol. 36, no. 12, pp. 3215-3223, 2009.

[8] V. Kachitvichyanukul, "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery," *Computers & Operations Research*, vol. 36, no. 5, pp. 1693-1702, 2009.

[9] A. Serdar Tasan and M. Gen, "A Genetic Algorithm Based Approach to Vehicle Routing Problem with Simultaneous Pick-up and Deliveries," *Computers & Industrial Engineering*, 2011.

[10] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239-251, 2002.

[11] J.Y. Potvin, "A review of bio-inspired algorithms for vehicle routing," in *Bio-inspired Algorithms for the Vehicle Routing Problem*, 2009, pp. 1-34.

[12] B.E. Gillett and L.R. Miller, "A heuristic algorithm for the vehicle-dispatch problem," *Operations Research*, pp. 340-349, 1974.

[13] G. Nagy and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries," *European Journal of Operational Research*, vol. 162, no. 1, pp. 126-141, 2005.

[14] N. Christofides, A. Mingozzi, and P. Toth, "*The vehicle routing problem*," Combinatorial optimization, vol. 11, 315-338, 1979.

[15] S. Ropke and D. Pisinger, "A unified heuristic for a large class of vehicle routing problems with backhauls," *European Journal of Operational Research*, vol. 171, no. 3, pp. 750-775, 2006.

[16] A. Subramanian, L. Cabral, and L. Ochi. (2012). "An efficient ILS heuristic for the vehicle routing problem with simultaneous pickup and delivery." *Relatório Técnico, Universidade Federal Fluminense*. [Online]. Available: http://www.ic.uff.br/~satoru/index.php

Chuhang Yu received the B.A. degree in Resources and Energy Engineering from Peking University, Beijing, China in 2011. She is curently working towards the MPhil degree at the Department of Industrial and Manufacturing Systems Engineering, the University of Hong Kong, Hong Kong.
Her research interests include artificial immune system, vehicle routing problem, transportation problem.

Henry Y. K. Lau received the B.A. degree in engineering science and the D.Phil. degree in ro-botics from the University of Oxford, Oxford, U.K in 1991.
Prior to joining The University of Hong Kong, Hong Kong, as an Associate Professor, he was a College Lecturer at Brasenose College, Oxford, and has been working in the industry as a Senior System Engineer with AEA Technology plc., U.K., working on projects involving system integration and telero-botics systems for the nuclear industry. His research interest includes intelligent automation, automated warehousing, interactive virtual reality, and design and evaluation of automated material-handling systems, such as automated warehouses using simulation and virtual-reality techniques.