

# Simulated Annealing for a Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Windows

Chao Wang, Fu Zhao, Dong Mu, John Sutherland

► **To cite this version:**

Chao Wang, Fu Zhao, Dong Mu, John Sutherland. Simulated Annealing for a Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Windows. Vittal Prabhu; Marco Taisch; Dimitris Kiritsis. 20th Advances in Production Management Systems (APMS), Sep 2013, State College, PA, United States. Springer, IFIP Advances in Information and Communication Technology, AICT-415 (Part II), pp.170-177, 2013, Advances in Production Management Systems. Sustainable Production and Service Supply Chains. <10.1007/978-3-642-41263-9\_21>. <hal-01451752>

**HAL Id: hal-01451752**

**<https://hal.inria.fr/hal-01451752>**

Submitted on 1 Feb 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Simulated annealing for a vehicle routing problem with simultaneous pickup-delivery and time windows

Chao Wang<sup>1,2</sup>, Fu Zhao<sup>2</sup>, Dong Mu<sup>1</sup>, John W. Sutherland<sup>2</sup>

1 School of Economics and Management, Beijing Jiaotong University, China  
mudong.bjtu@gmail.com

2 College of Engineering, Purdue University, United State  
{chaowang, fzhaoy, jwsuther}@purdue.edu

**Abstract.** A variant of vehicle routing problem in which customers require simultaneous pickup and delivery goods during specific individual time windows (VRPSPDTW) is addressed. A general mixed integer programming model focused on minimizing the number of vehicles is proposed. A simulated annealing (SA) algorithm is developed and applied to solve this NP-hard optimization problem. Computational results are reported for a set of 15 test problems from Wang and Chen's benchmark and compared with the results from a genetic algorithm (GA). It is shown that SA is an effective metaheuristic to solve the VRPSPDTW.

**Keywords:** simulated annealing, simultaneous delivery and pickup, vehicle routing problem, time windows

## 1 Introduction

The Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Windows (VRPSPDTW) is the important subclass of Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) where a fleet of homogenous or non-homogenous vehicles stationed at a depot are not only required to deliver from the depot to customers but also simultaneously to pickup goods at the customer location for return to the depot without violating vehicle capacity constraints and time windows constraints. Time windows define the earliest and latest service time allowed for each customer.

From a practical point of view, VRPSPDTW falls within the field of logistics resource integration and supply chain integration which incorporates reverse logistics into the forward channel and has attracted more and more attention recently because of the increasing focus on environmental protection and sustainable supply chains. For example, Hewlett-Packard (HP) was reported to distribute the products in logistics network and pick up the used IT hardware such as used original toner cartridges and supplies from the retail location [1]. From a mathematical point of view, VRPSPDTW is a NP-hard combinatorial optimization problem which means that exact algorithms can only be used to find solutions for small-and-medium scale in-

stances. The commercial linear programming software CPLEX has been reported to solve the VRPSPDTW problem executed on an Intel Core2 Quad 2.4G computer with 1G memory, and was only able to solve a 10 customer instance, and a portion of 25 and 50 customer scenarios [2]. For the solved instance RCdp 1004 (10 customers instance) and RCdp5001 (50 customers instance), the computation time was 1503 seconds and 327,404 seconds respectively. To solve the computational challenge associated with applied problem that involve more customers, researchers and practitioners have to develop heuristic or metaheuristic approaches to produce high-quality solutions (but not necessarily optimal solutions) with reasonable computational times.

The technical literature on VRPSPDTW problem is somewhat limited. Angelelli and Mansini [3] was the first researchers to solve VRPSPDTW with a comprehensive exact algorithm which combined a branch-and-price and branch-and-bound approach. Lai and Cao [4] proposed an Improved Differential Evolution (IDE) algorithm for solving this problem and did numerical experiments with their own instances. Boubahri et al. [5] constructed a multi-agent colonies algorithm for this problem, but did not test with the instances. Wang and Chen [2] proposed a co-evolution genetic algorithm with variants of the cheapest insertion method for VRPSPDTW. They also developed 65 instances revised from the well-known Solomon benchmark [6] for VRPTW.

However the literature on the VRPSPD problem is better studied than the more specific class of problem noted above, i.e., research that does not consider time windows. For good reviews on the VRPSPDTW, the reader is referred to Subramanian et al. [7]. Min [8] was the first to study this problem. He solved the book distribution problem in a public library system which involved 22 customers and two vehicles. Some exact methods have been developed for the VRPSPD, including the branch-and-price technique [9] [10] and the branch-and-cut technique [11]. However, due to the problem difficulty, the design of efficient heuristics or metaheuristics which are capable of solving this problem in large scale instances has attracted many research efforts in the past few years. Some example heuristics are: Residual Capacity and Radical Surcharge (RCRS) algorithm [12], Large Neighborhood Search (LNS) [13], and GENEVNS-TS-CL-PR algorithm [14]. Some example metaheuristics are: pure or hybrid Tabu Search (TS) [15] [16], Ant Colony Optimization (ACO) [17] [18], Genetic Algorithm (GA) [19], Particle Swarm Optimization (PSO) [20], and Iterated Local Search (ILS) algorithm [7] [21].

The purpose of the present study is to develop an efficient metaheuristic to solve the VRPSPDTW problem by using simulated annealing algorithm. The paper is organized as follows. Section 2 gives a detailed description of how the proposed simulated annealing algorithm is implemented on the VRPSPDTW problem. Section 3 provides comparisons to the VRPSPDTW benchmark. Finally, conclusions are drawn in Section 4.

## 2 Simulated Annealing (SA) Applied to VRPSPDTW

The problem-solving methodology based on the SA algorithm is presented in this section. SA implements a Metropolis sampling strategy with probability mutability, randomizes the local search procedure, and obtains the global optimum with a slow cooling schedule. The most important feature of SA compared with other local search algorithms is that SA always accepts a better or unchanged solution as a new current solution, and accepts a worse solution with a certain probability, which allows the search to escape from local minima and become a global optimum algorithm in theory.

The text below describes the steps in the simulated annealing algorithm applied to solve the VRPSPDTW problem.

### Step 1 Initialization of the solution and temperature

(1) Generate an initial heuristic solution. In order to improve the convergence speed of SA, it is better to introduce a small-size heuristic algorithm and find a high quality feasible solution as the initial solution. VRPSPDTW considers a simultaneous pickup and delivery, therefore the Push-Forward Insertion Heuristic (PFIH) and Genetic Sectoring Heuristic (GSH), which are used for solving VRPTW, are no longer applicable because the vehicles are assumed to be at full load when leaving the central depot in their algorithms. In this paper, the initial solution is constructed using the RSRC algorithm [12] proposed by Dethloff for the VRPSPD problem, where it is noted that the load of vehicle when leaving is depot is not equal to vehicle capacity, but rather, calculated in the algorithm.

(2) Set initial annealing temperature. The setting of an initial temperature determines whether the initial stage of the annealing process can accept poorer solutions with high probability. Based on the characteristics of VRPSPDTW, and combined with the relevant literature [22] about simulated annealing theory, the following method can be used to choose the initial temperature.

The cost and cooling temperature of the VRPSPDTW solution is computed as follows:

$$\text{cost}(s_k) = d + \sigma(c \times n + e_{\min}) \quad (1)$$

$$T_k = \gamma \times \text{cost}(s_k) \quad (2)$$

where  $s_k$  is the feasible solution,  $d$  is the total travel distance of the routes,  $\sigma$  is a constant,  $c$  is the number of routes in solution  $s_k$  (equal to the number of vehicles needed),  $n$  is the number of customers,  $e_{\min}$  is the number of customers in the shortest route, and  $\gamma < 1$  is a constant.

Van Laarhoven and Aarts [23] proposed a method of decreasing temperature which appears to be the most widely used method reported in the literature:

$$T_{k+1} = \beta * T_k \quad (3)$$

where  $\beta$  is the coefficient controlling the cooling schedule, and a constant less than 1.

When  $k = 0$ , the initial temperature is:

$$T_0 = \gamma \times \text{cost}(s_0) \quad (4)$$

**Step 2** Repeat the following operation under the current temperature  $T_k$  until it reaches the equilibrium state at  $T_k$ .

(1) Local search in the neighborhood of the current solution  $s_k$  to generate a new feasible solution  $s'_k$ . We use a standard SA procedure with a random neighborhood structure which imposes three types of improvement methods, including insertion move, swap move, and 2-opt move, to solve VRPSPDTW. These moves are commonly embedded in SA heuristics, and other meta-heuristics. We define the set  $N(s_k)$  to be the set of solutions neighboring of solution  $s_k$ . At each iteration, the next solution  $s'_k$  is selected from  $N(s_k)$  either by insertion move, 2-opt move, or swap/shift move. The probabilities of choosing the insertion move, 2-opt move, or swap/shift move are set to be 1/3, 1/3 and 1/3, respectively.

Insertion move is a random improvement method, which is carried out by randomly selecting the  $i$ th number of  $s_k$  and inserting it into the position immediately before another randomly selected  $j$ th number of  $s_k$ . 2-opt move is an intra-route exchange, which tries to lower the route distance by altering the sequence of customers within a single route. It switches the route direction between two consecutive nodes. If the cost function of the route has been reduced, then the modified route is kept; otherwise, the route returns to the last condition. Swap/shift move is an inter-route exchange proposed. It either shifts a customer from one route to another, or exchanges a customer on one route with a customer on another route.

(2) Determine whether to accept the new solution

The SA process proceeds by comparison of the cost of the newly generated feasible solution  $s'_k$  and the current solution  $s_k$ . If  $s'_k \leq s_k$ , the newly generated solution  $s'_k$  becomes the current solution. If  $s'_k > s_k$ ,  $s'_k$  is accepted with the probability

$$p = \begin{cases} 1 & \text{cost}(s_k) > \text{cost}(s'_k) \\ \exp\left(\frac{\text{cost}(s_k) - \text{cost}(s'_k)}{T_k}\right) & \text{cost}(s_k) < \text{cost}(s'_k) \end{cases} \quad (5)$$

where  $T_k$  ( $k = 0, 1, \dots, k_{\max}$ ) is a parameter called temperature of annealing. This temporary deterioration of the objective function value is an uphill move process, which avoids the metaheuristic to trap into a local optimum.

(3) Determine whether to terminate the inner loop

Metropolis sampling stability criterion is used to determine the number of candidate solutions produced at the respective temperatures. Here we set L steps as the maximum number of iterations at temperature  $T$ . If the iterations counter for temperature  $T$  is not exceed L, return to **Step 2** (1) and continue the procedure of generating and accepting new feasible solution, otherwise, the inner loop is terminated, the temperature is decreased according to Equation (3) and transferred to **Step 3**.

**Step 3** End the procedure if the stopping criterion is met

The common stopping criteria are: (i) specify a determined maximum number of total iterations the search proceeds; (ii) set the final temperature  $T_F$ , below  $T_F$  the SA procedure is terminated; (iii) specify a maximum number of successive iterations  $\omega_{\max}$  which remains the non-improving best solution found so far for a consecutive temperature reduction stages. This paper uses the third stop criterion, which determines whether to stop the outer loop based on whether the gradually lowered temperature during the search procedure is able to continue to improve the current optimal solution or not.

**Step 4** Output the optimal solution.

Output the optimal solution, and terminate the algorithm.

### 3 Computation and Evaluation

In this section, we demonstrate the effectiveness of our SA metaheuristic by testing it on the benchmark datasets, namely Wang and Chen’s instances [2]. All experiments are executed on a 32 core Redhat Linux server ( $8 \times$  quad core 2.3 Mhz, Opteron 8346) with 64G of RAM.

In the proposed SA, parameters must be specified and appropriately adjusted: parameter  $\sigma$  in Equation (1), parameter  $\gamma$  represents relations between cost and temperature, the constant of decreasing temperature  $\beta$ ,  $\Delta$  which represents the probability of accepting worse solutions, parameter  $L$  which is the maximum number of iterations for temperature  $T$ , parameter  $\omega_{\max}$  which is the maximum number of iterations of non-improving the current best solution, parameters *lamda* and *alpha* in RCRS insertion criterion proposed by Dethloff [12]. The above parameter values were obtained through fine tuning and are listed in Table 1. In general the performance of the SA algorithms is sensitive to the above cooling schedule.

**Table 1.** Parameters used in the SA

SA paramater	$\sigma$	$\gamma$	$\beta$	$L$	$\omega_{\max}$	$\Delta$	<i>lamda</i>	<i>alpha</i>
Value	1	1	0.97	30	200	0.6	0.8	0.8

Wang and Chen’s instances<sup>1</sup> about VRPSPDTW were revised from Solomon’s VRPTW benchmarks<sup>2</sup>. Six different sets of problems were generated. The geographical data are randomly generated in problem sets R1 and R2, clustered in problem sets C1 and C2, and a mix of random and clustered structures in problem sets by RC1 and RC2. Problem sets R1, C1 and RC1 have a narrow scheduling horizon, and the sets R2, C2 and RC2 have a large scheduling horizon. Three 10-customer instances, three

<sup>1</sup> Download of Wang and Chen’s benchmark instances from Y. Y. Chen:  
<http://oz.nthu.edu.tw/~d933810/test.htm>

<sup>2</sup> Download of Solomon’s benchmark instances from M. M. Solomon:  
<http://w.cba.neu.edu/~msolomon/problems.htm>

25-customer instance, three 50-customer instances and six 100-customer instances are tested for proposed SA.

Table 2 compares the best results obtained by Wang and Chen [2] (GA metaheuristic) with the results obtained by the simulated annealing we developed for part of 65 Wang and Chen benchmark. The last column of this table shows the gaps between GA and proposed SA. These results are summarized at the bottom of the table. The mean and maximum percentage gaps are -0.22% and 1.61% respectively.

**Table 2.** Comparison between GA and the proposed SA to the VRPSPDTW

Problem/Instance	GA		Proposed SA		Gap	
	NV	TD	NV	TD	NV	TD
Rcdp1001/10	3	348.98	3	348.98	0	0.00%
Rcdp1004/10	2	216.69	2	216.69	0	0.00%
Rcdp1007/10	2	310.81	2	310.81	0	0.00%
Rcdp2501/25	5	551.05	5	554.33	0	0.60%
Rcdp2504/25	4	473.46	4	473.46	0	0.00%
Rcdp2507/25	5	540.87	5	540.87	0	0.00%
Rcdp5001/50	9	994.18	9	994.7	0	0.05%
Rcdp5004/50	6	725.59	6	725.9	0	0.04%
Rcdp5007/50	7	809.72	7	810.04	0	0.04%
Rdp101/100	19	1653.53	19	1660.98	0	0.45%
Cdp101/100	11	1001.97	11	992.88	0	-0.91%
RCdp101/100	15	1652.9	15	1679.46	0	1.61%
Rdp201/100	4	1280.44	4	1286.55	0	0.48%
Cdp201/100	3	591.56	3	591.56	0	0.00%
RCdp201/100	4	1587.92	4	1497.8	0	-5.68%
				Minimum		-5.68%
				Mean		-0.22%
				Maximum		1.61%

Take RCdp1001 instance as an example, a run of SA shows that the best operational plan is: vehicle 1 (route 1): 1-31-51-92-1, the delivery and pickup amount is 53 and 23 respectively, the route distance is 104.85 and need 189.04 to finish this service; vehicle 2 (route 2): 1-83-72-95-101-1, the delivery and pickup amount is 49 and 90 respectively, the route distance is 116.68 and need 209.24 to finish this service; vehicle 3 (route 3): 1-60-88-49-1, the delivery and pickup amount is 52 and 83 respectively, the route distance is 127.45 and need 206.04 to finish this service; the total distance and total travel time traveled by the three vehicles is 348.98 and 604.32 respectively.

## 4 Summary and conclusion

The vehicle routing problem with simultaneous pickup-delivery and time windows is a very application-intensive but notoriously NP-hard problem, so the complexity of medium-and-large instances in practice does not allow an exact solution of the problem, making it necessary to implement heuristic or metaheuristic approaches to provide the approximated solutions within reasonable computational time. This paper has presented a simulated annealing metaheuristic for the VRPSPDTW problem. To the

best of our knowledge, this is the first simulated annealing approach proposed to solve this problem. Comparisons from the results of the GA algorithm and the proposed SA show that the proposed method is effective for solving VRPSPDTW problem.

High-quality solutions are guaranteed by the tight cooling schedule, particularly when a sufficiently high initial temperature, slow annealing speed, a large number of iterations, and holding time at the same temperature are utilized. This is usually at the price of a high computation resources and times. Therefore, further work will concentrate on parallelization of SA to accelerate the proposed algorithm.

### **Acknowledgements**

This work has been supported by the National Natural Science Foundation of China (Grant No. 71132008), China Scholarship Council (201207090034), and the Fundamental Research Funds for the Central Universities (2012YJS034). Also, this research was supported in part by the computational resources provided by the Division of Environmental and Ecological Engineering at Purdue University.

### **References**

1. HP Global citizenship, <http://www8.hp.com/us/en/hp-information/environment/product-recycling.html#UVH59pHEovw>.
2. Wang, H.F., Chen, Y.Y.: A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & Industrial Engineering* 62(1), 84-95 (2012).
3. Angelelli, E., Mansini, R.: A branch-and-price algorithm for a simultaneous pickup and delivery problem. *Quantitative Approaches to Distribution Logistics and Supply Chain Management*, 249-267 (2003).
4. Lai, M.Y., Cao, E.B.: An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. *Engineering Applications of Artificial Intelligence* 23(2), 188-195 (2010). doi:<http://dx.doi.org/10.1016/j.engappai.2009.09.001>
5. Boubahri, L., Addouche, S.A., El Mhamedi, A.: Multi-ant colonies algorithms for the VRPSPDTW. In: *Communications, Computing and Control Applications (CCCA), 2011 International Conference on*, 3-5 March 2011 2011, pp. 1-6
6. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35(2), 254-265 (1987).
7. Subramanian, A., Drummond, L.M.A., Bentes, C., Ochi, L.S., Farias, R.: A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Computers & Operations Research* 37(11), 1899-1911 (2010). doi:<http://dx.doi.org/10.1016/j.cor.2009.10.011>
8. Min, H.: The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General* 23(5), 377-386 (1989). doi:[http://dx.doi.org/10.1016/0191-2607\(89\)90085-X](http://dx.doi.org/10.1016/0191-2607(89)90085-X)



9. Dell'Amico, M., Righini, G., Salani, M.: A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science* 40(2), 235-247 (2006).
10. Subramanian, A., Uchoa, E., Pessoa, A.A., Ochi, L.S.: Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Operations Research Letters* 39(5), 338-341 (2011).
11. Rieck, J., Zimmermann, J.: A Branch-and-Cut Approach to the Vehicle Routing Problem with Simultaneous Delivery and Pick-up. *Operations Research Proceedings 2008*, 301-306 (2009).
12. Dethloff, J.: Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *Or Spectrum* 23(1), 79-96 (2001).
13. Ropke, S., Pisinger, D.: A unified heuristic for a large class of vehicle routing problems with backhauls. *Eur. J. Oper. Res.* 171(3), 750-775 (2006).
14. Cruz, R., Silva, T., Souza, M., Coelho, V., Mine, M., Martins, A.: GENVNS-TS-CL-PR: A heuristic approach for solving the vehicle routing problem with simultaneous pickup and delivery. *Electronic Notes in Discrete Mathematics* 39, 217-224 (2012).
15. Montane, A.T.F., Galvao, R.D.: A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research* 33(3), 595-619 (2006).
16. Zachariadis, E.E., Kiranoudis, C.T.: A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications* 38(3), 2717-2726 (2011).
17. Gajpal, Y., Abad, P.: An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research* 36(12), 3215-3223 (2009).
18. Çatay, B.: A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Expert Systems with Applications* 37(10), 6809-6817 (2010).
19. Tasan, A.S., Gen, M.: A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering* (2011).
20. Ai, T.J., Kachitvichyanukul, V.: A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research* 36(5), 1693-1702 (2009).
21. Subramanian, A., Cabral, L.D.A.F.: An ILS based heuristic for the vehicle routing problem with simultaneous pickup and delivery and time limit. *Evolutionary Computation in Combinatorial Optimization*, 135-146 (2008).
22. Czech, Z.J., Czarnas, P.: Parallel simulated annealing for the vehicle routing problem with time windows. In: *Parallel, Distributed and Network-based Processing, 2002. Proceedings. 10th Euromicro Workshop on 2002*, pp. 376-383. IEEE
23. van Laarhoven, P.J., Aarts, E.H.: *Simulated annealing: theory and applications*, vol. 37. Springer, (1987)