17

Logistics

AN IMPROVED CLARKE-WRIGHT ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH SIMULTANEOUS DELIVERY AND PICKUP

Tantikorn Pichpibul[1*], Ruengsak Kawtummachai[1], Anirut Kantasa-ard[2]

[1]Faculty of Logistics and Transportation Management, Panyapiwat Institute of Management, Thailand
[2]LAMIH, UMR CNRS 8201, Université Polytechnique Hauts-de-France, France
[*]**Corresponding author, E-mail:** pichpibul@gmail.com

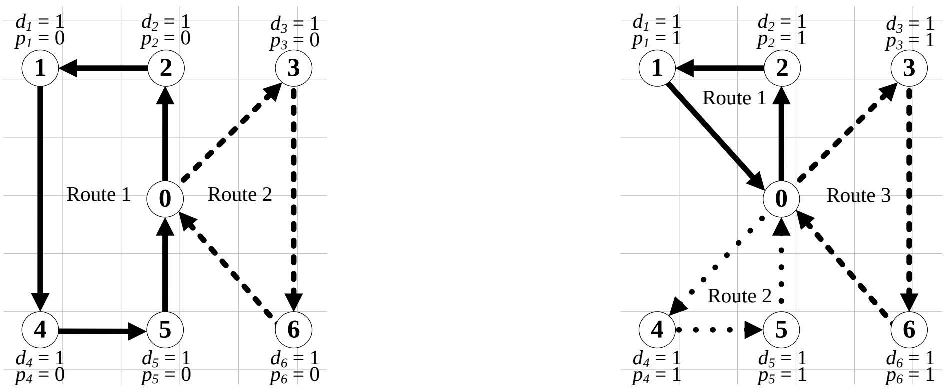
ABSTRACT

An improved Clarke-Wright (ICW) algorithm for the well-known vehicle routing problem, which simultaneously considers the customer demand from both delivery and pickup orders, is presented. In this paper, the Clarke-Wright (CW) algorithm is improved using three procedures composed of route linking, route making, and improved savings sorting. The objective is to find the solution that minimizes total cost. Computational results show that the ICW algorithm is competitive and outperforms the CW algorithm in all directions. Moreover, the best known solutions are also obtained in 90% of tested instances (36 out of 40).

Keywords: Vehicle routing problem, Clarke-Wright algorithm.

Introduction

The vehicle routing problem with simultaneous delivery and pickup (VRPSDP) is one variant of the well-known vehicle routing problem (VRP) introduced by Min (1989). The characteristics of VRPSDP are based on the capacitated vehicle routing problem (CVRP) introduced by Dantzing and Ramser (1959). It involves a single warehouse, a homogeneous fleet of vehicles, and a set of customers. The objective is to create vehicle routes that minimize the total cost while serving all customers' demands, including delivery (d) and pickup (p) orders. The different routes for CVRP and VRPSDP are shown in Figure 1. Similar to those routes, all vehicles always depart from the warehouse (0) and return to the warehouse after completing the service. In the case of CVRP, each vehicle only delivers the products to customers as shown in Figure 1(a). But in contrast to VRPSDP, each vehicle not only delivers the products to customers but also simultaneously picks up the defective products from the same customers as shown in Figure 1(b).



(a) CVRP (b) VRPSDP

Figure 1: Different routes for CVRP and VRPSDP

Literature Review

The VRPSDP can be stated as follows: A set of customers (*N*) with known demands (d_n, p_n) must be visited by a homogeneous fleet of vehicles where all vehicles have the same loading capacity (*Q*). The constraints of VRPSDP are as follows.

- Each customer must be visited once by one vehicle.
- Each customer is visited and left with the same vehicle.
- All vehicles start and end at the warehouse.
- The total delivery and pickup demands in each vehicle cannot exceed *Q*.

Because CVRP is known as a complex NP-hard problem, Jun and Kim (2012) mentioned that when the pickup demand of all customers is equal to zero, VRPSDP becomes CVRP. Therefore, VRPSDP is also an NP-hard. This has received much attention from researchers to develop algorithms designed to solve many variants of the well-known VRP. According to those algorithms, the Clarke-Wright (CW) algorithm (Clarke and Wright, 1964) is one of the most widely applied algorithms to solve those problems. But Cordeau et al. (2002) mentioned that the CW algorithm rates high in speed and simplicity, but is not the most accurate. Therefore, development of a better solution for the CW algorithm by many researchers (Altinel and Öncan, 2005; Juan et al., 2011; Doyuran and Çatay, 2011) is required. There is no previously published research that presents an improved CW algorithm to solve the VRPSDP addressed in this paper. This paper contributes to improving the CW algorithm with simple but powerful procedures that can efficiently solve VRPSDP in terms of the solution quality.

Methods

In this paper, an improved Clarke-Wright (ICW) algorithm is developed to solve VRPSDP. This improved solution uses three procedures on the CW algorithm composed of route linking, route making, and improved savings sorting. The ICW is described by an example case of four customers and two vehicles in which each vehicle capacity is equal to three. We consider that the customer location and demand are shown in Figure 2. The flowchart of ICW is also given in Figure 3.

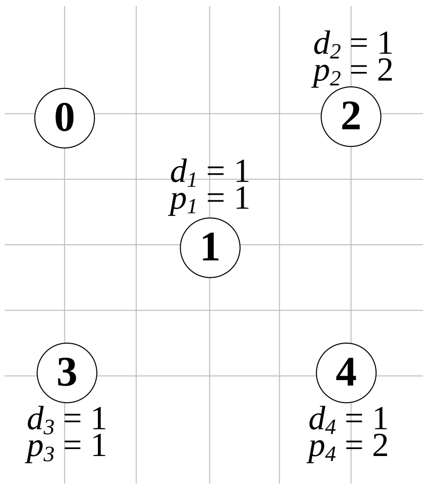


Figure 2: Customer location and demand of example case

1. Cost Loading

The cost ($c_{i,j}$) is loaded from the cost matrix as shown in Table 1 without calculation. Note that the warehouse is assigned to be customer 0 and $c_{i,j} = c_{j,i}$ due to the symmetric cost.

Table 1: Cost matrix**.**

Customer	**0**	**1**	**2**	**3**	**4**
0	-	14.14	20.00	20.00	28.28
1	14.14	-	14.14	14.14	14.14
2	20.00	14.14	-	28.28	20.00
3	20.00	14.14	28.28	-	20.00
4	28.28	14.14	20.00	20.00	-

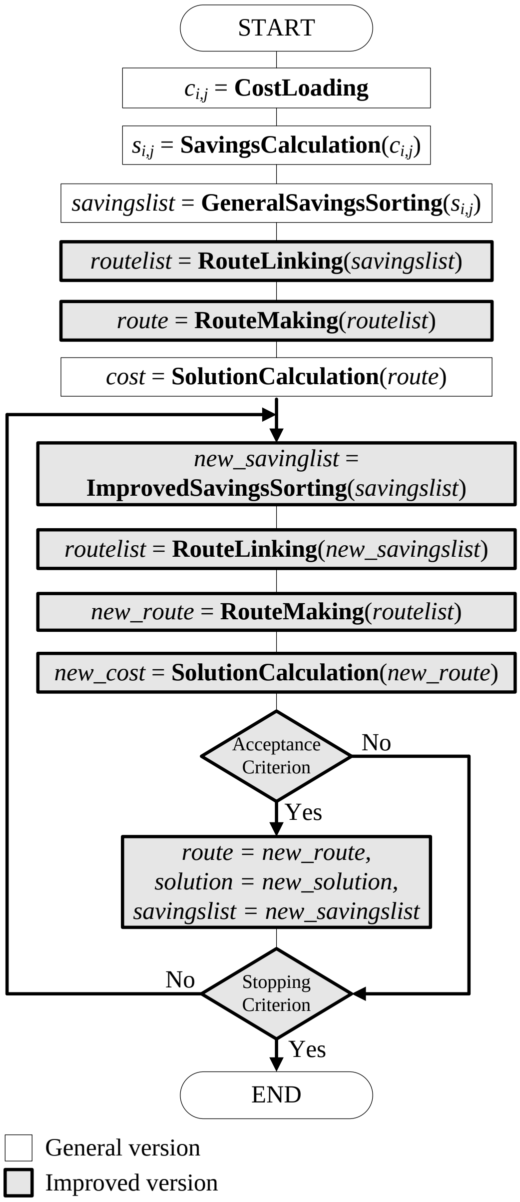


Figure 3: Flowchart of ICW

2. Savings Calculation

The savings ($s_{i,j}$) between customers *i* and *j* is calculated by using the following equation.

$$s_{i,j} = c_{i,0} + c_{0,j} - c_{i,j} \quad (1)$$

where $c_{i,0}$ is the cost from customer *i* to the warehouse, $c_{0,j}$ from the warehouse to customer *j*, and $c_{i,j}$ from customers *i* to *j*. The savings calculation is shown below.

- $s_{1,2} = c_{1,0} + c_{0,2} - c_{1,2}$ = 14.14+20.00–14.14=20.00
- $s_{1,3} = c_{1,0} + c_{0,3} - c_{1,3}$ = 14.14+20.00–14.14=20.00
- $s_{1,4} = c_{1,0} + c_{0,4} - c_{1,4}$ = 14.14+28.28–14.14=28.30
- $s_{2,3} = c_{2,0} + c_{0,3} - c_{2,3}$ = 20.00+20.00–28.28=11.70
- $s_{2,4} = c_{2,0} + c_{0,4} - c_{2,4}$ = 20.00+28.28–20.00=28.30
- $s_{3,4} = c_{3,0} + c_{0,4} - c_{3,4}$ = 20.00+28.28–20.00=28.30

3. General Savings Sorting

In the general procedure, all savings then are sorted in decreasing order (from the largest to the smallest), and are collected into the savings list consisting of six savings ($s_{1,4}$, $s_{2,4}$, $s_{3,4}$, $s_{1,2}$, $s_{1,3}$, $s_{2,3}$).

4. Route Linking

The linking of routes starts from choosing the largest savings in the savings list. Any two customers *i* and *j* will be linked and collected into the route list if the total delivery and pickup demands do not exceed the vehicle capacity. This procedure is repeated to process the next savings in the savings list until it reaches the last savings. The example of route linking procedure is shown in Table 2. Finally, the route list is composed of two savings ($s_{1,4}$, $s_{2,3}$).

Table 2: Route linking procedure**.**

$s_{i,j}$	**Route**	**Calculation**
$s_{1,4}$	1	$(d_1 + d_4) \leq 3$ = (1+1) = 2 and ($p_1 + p_4) \leq 3$ = (1+2) = 3
$s_{2,4}$	-	$(d_1 + d_4 + d_2) \leq 3$ = (1+1+1) = 3 and ($p_1 + p_4 + p_2) \leq 3$ = (1+2+2) = 5
$s_{3,4}$	-	$(d_1 + d_4 + d_3) \leq 3$ = (1+1+1) = 3 and ($p_1 + p_4 + p_3) \leq 3$ = (1+2+1) = 4
$s_{1,2}$	-	$(d_1 + d_4 + d_2) \leq 3$ = (1+1+1) = 3 and ($p_1 + p_4 + p_2) \leq 3$ = (1+2+2) = 5
$s_{1,3}$	-	$(d_1 + d_4 + d_3) \leq 3$ = (1+1+1) = 3 and ($p_1 + p_4 + p_3) \leq 3$ = (1+2+1) = 4
$s_{2,3}$	2	$(d_2 + d_3) \leq 3$ = (1+1) = 2 and ($p_2 + p_3) \leq 3$ = (2+1) = 3

5. Route Making

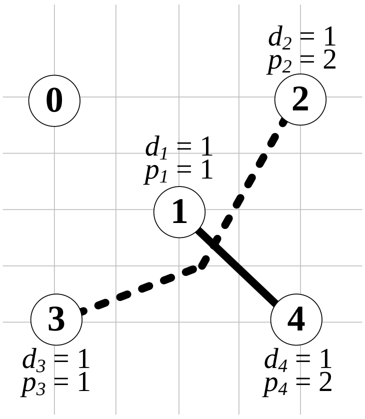
The route is made through the inclusion of customers i and j in each savings ($s_{i,j}$) from the route list. Figure 4(a) shows the final inclusion of all savings, which represents two initial routes (1-4 and 2-3). In each route, the first customer is assigned to start from the warehouse, and the last customer is assigned to return to the warehouse. Therefore, all possible solutions are made as illustrated in Figure 4(b).

6. Solution Calculation

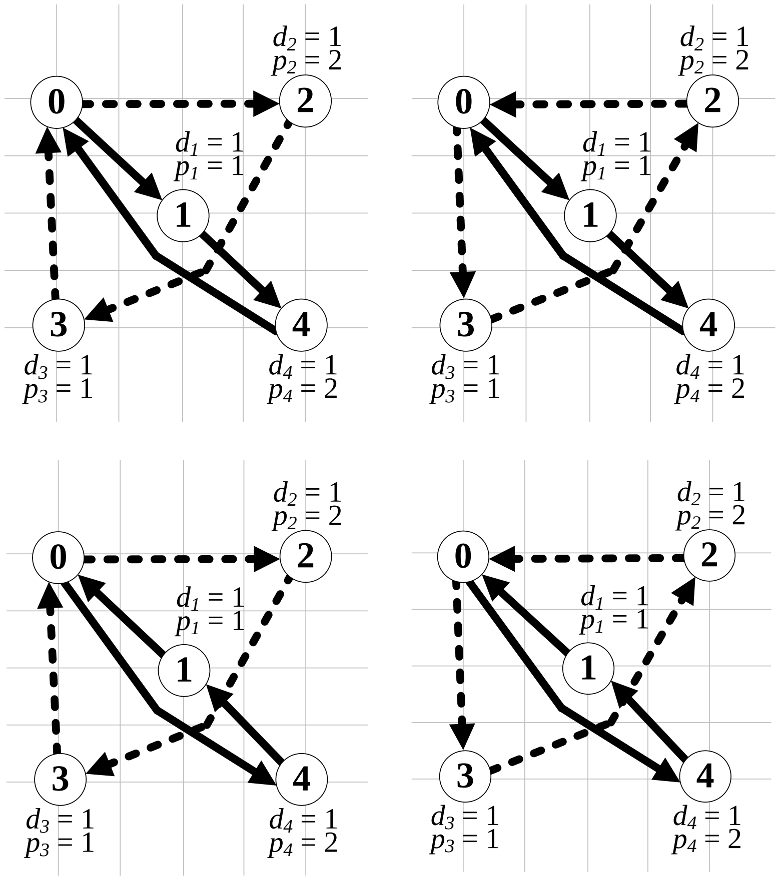
The total cost in each solution from Figure 4(b) is calculated, and the solution calculation is shown in Table 3. Note that the total cost of all solutions are similar due to the symmetric cost (0-2-3-0 = 0-3-2-0 and 0-1-4-0 = 0-4-1-0). Therefore, solutions 1, 2, 3, and 4 from Table 3 are the best solutions (lowest cost) and one of them will be chosen to be the VRPSDP solution.

Table 3: Solution calculation procedure**.**

Solution	Route	Cost	Total Cost
1	0-2-3-0	20.00 + 28.28 + 20.00 = 68.28	68.28 + 56.56 = 124.84
	0-1-4-0	14.14 + 14.14 + 28.28 = 56.56	
2	0-3-2-0	20.00 + 28.28 + 20.00 = 68.28	68.28 + 56.56 = 124.84
	0-1-4-0	14.14 + 14.14 + 28.28 = 56.56	
3	0-2-3-0	20.00 + 28.28 + 20.00 = 68.28	68.28 + 56.56 = 124.84
	0-4-1-0	28.28 + 14.14 + 14.14 = 56.56	
4	0-3-2-0	20.00 + 28.28 + 20.00 = 68.28	68.28 + 56.56 = 124.84
	0-4-1-0	28.28 + 14.14 + 14.14 = 56.56	



(a) Two initial routes



(b) All possible solutions

Figure 4: Route-making procedure

7. Improved Savings Sorting

In this procedure, all savings will be sorted differently by using the two-phase selection from the work of Pichpibul and Kawtummachai (2012), and are collected into the new savings list. The application of the two-phase selection for VRPSDP is illustrated in Figure 5 and can be described as follows.

In the first iteration, the top three savings in the savings list are chosen in order to move one savings to the new savings list. The same sorting process in the next iteration is repeated until all savings from the old savings list are moved to the new savings list. Finally, the new savings list consists of six savings ($s_{2,4}$, $s_{1,4}$, $s_{1,3}$, $s_{3,4}$, $s_{2,3}$, $s_{1,2}$).

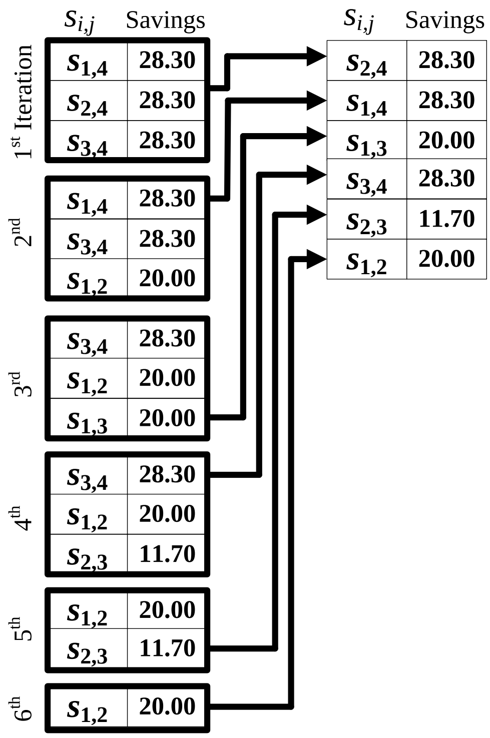
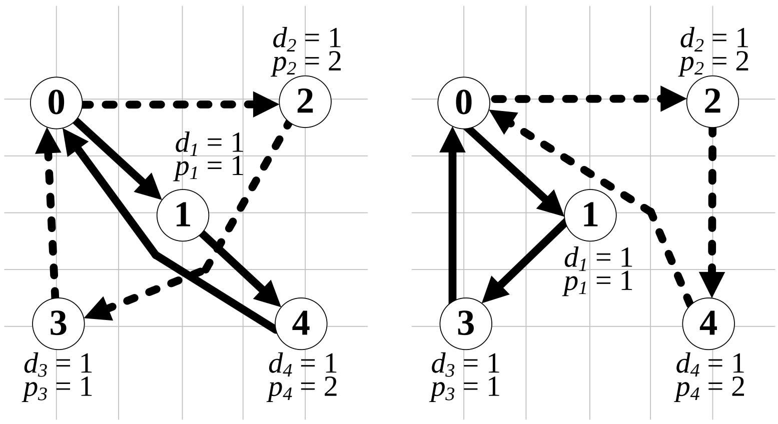


Figure 5: Improved savings sorting procedure

8. Acceptance Criterion

The new savings list is calculated by route linking and route making to create a new VRPSDP solution. After comparing both the old solution in Figure 6(a) and the new solution in Figure 6(b), the new savings list will replace the old one because the new solution is better (lower cost).



(a) Old solution (b) New solution

Figure 6: Acceptance criterion procedure

9. Stopping Criterion

The ICW is repeated until the stopping criterion, which is the number of iterations, is satisfied. In this paper, it is equal to 10,000.

Results and Discussion

The ICW was coded in Visual Basic 6 on an Intel® Core™ i7-5500U CPU 2.4 GHz with 4 GB of RAM under Windows 7 (64-bit) platform. The numerical experiment used VRPSDP benchmark problems composed of 40 instances from Dethloff (2001). The performance of ICW is evaluated by comparing it with algorithms for VRPSDP as shown in Table 4. Table 5 describes the development of ICW in detail. Each instance is discussed, in which the percentage improvement between the ICW-1 solution (icw_1) and the ICW-2 solution (icw_2) is calculated as follows:

$$\text{Improvement (\%)} = \left(\frac{icw_1 - icw_2}{icw_1}\right) \times 100 \quad (2)$$

Moreover, the percentage deviation between obtained solution (obt) and the best known solution (bes) is also calculated as follows:

$$\text{Deviation (\%)} = \left(\frac{obt - bes}{bes}\right) \times 100 \quad (3)$$

The numerical results from Table 6 indicate that ICW can find high quality solutions. Out of 40 instances, the best known solutions are obtained for 36 instances. For four instances (SCA3-0, SCA8-5, CON3-8, CON8-4), the percentage deviation between ICW solutions and the best known solutions are very low (0.069, 0.014, 0.017, 0.009).

Moreover, the results in Table 6 show that ICW-2 always performed better than ICW-1. These indicate that ICW is effective and efficient in producing high quality solutions for VRPSDP benchmark problems.

Table 4: The algorithms used to compare with ICW.

Authors	Abbreviation	Algorithm
Dethloff (2001)	S-1	Insertion-based heuristics
Tang and Galvao (2006)	S-2	A tabu search algorithm
Gajpal and Abad (2009)	S-3	An ant colony system
Souza et al. (2011)	S-4	Iterated local search and GENIUS
Subramanian et al. (2013)	S-5	Branch-cut-and-price algorithm
Goksal et al. (2013)	S-6	A hybrid discrete particle swarm optimization
Yousefikhoshbakht et al. (2014)	S-7	A combination of modified tabu search and elite ant system

Table 5: The development of ICW.

Abbreviation	Details
ICW-1	Improve CW solution with route linking and route making procedures.
ICW-2	Improve ICW-1 solution with improved savings sorting procedure.

Conclusions

In this paper, an improved Clarke-Wright (ICW) algorithm has been presented to solve the vehicle routing problem with simultaneous delivery and pickup (VRPSDP). The ICW algorithm has been improved by using route linking, route making, and an improved savings sorting. The experiments have been done using a VRPSDP benchmark problem composed of 40 instances obtained from the literature.

The computational results show that ICW is competitive and outperforms the CW algorithm in all directions in terms of solution quality. It also ties with the existing best known solutions for 36 instances out of 40 tested instances. Therefore, it can be concluded that the performance of ICW is excellent for solving VRPSDP.

Moreover, it would be an interesting and significant extension of this paper to study more powerful algorithms to improve the existing ICW solution.

Table 6: Computational results.

Instance	Solution									Improvement (%)
	S-1	S-2	S-3	S-4	S-5	S-6	S-7	ICW-1	ICW-2	
SCA3-0	689.00	640.55	635.62	635.62	635.62	635.62	635.62	684.72	636.06	7.107
SCA3-1	765.60	697.84	697.84	697.84	697.84	697.84	697.84	701.98	697.84	0.590
SCA3-2	742.80	659.34	659.34	659.34	659.34	659.34	659.34	706.83	659.34	6.719
SCA3-3	737.20	680.04	680.04	680.04	680.04	680.04	680.04	726.47	680.04	6.391
SCA3-4	747.10	690.50	690.50	690.50	690.50	690.50	690.50	796.99	690.50	13.362
SCA3-5	784.40	659.90	659.90	659.90	659.90	659.90	659.90	730.25	659.90	9.634
SCA3-6	720.40	653.81	651.09	651.09	651.09	651.09	651.09	715.42	651.09	8.992
SCA3-7	707.90	659.17	659.17	659.17	659.17	659.17	659.17	688.55	659.17	4.267
SCA3-8	807.20	719.47	719.47	719.47	719.47	719.47	719.47	746.35	719.47	3.602
SCA3-9	764.10	681.00	681.00	681.00	681.00	681.00	681.00	704.94	681.00	3.396
SCA8-0	1132.90	981.47	961.50	961.50	961.50	961.50	961.50	1015.50	961.50	5.318
SCA8-1	1150.90	1077.44	1049.65	1049.65	1049.65	1049.65	1052.40	1129.43	1049.65	7.064
SCA8-2	1100.80	1050.98	1042.69	1039.64	1039.64	1039.64	1039.64	1079.30	1039.64	3.675
SCA8-3	1115.60	983.34	983.34	983.34	983.34	983.34	983.34	1049.34	983.34	6.290
SCA8-4	1235.40	1073.46	1065.49	1065.49	1065.49	1065.49	1065.49	1112.03	1065.49	4.185
SCA8-5	1231.60	1047.24	1027.08	1027.08	1027.08	1027.08	1027.08	1085.68	1027.22	5.385
SCA8-6	1062.50	995.59	971.82	971.82	971.82	971.82	971.82	986.96	971.82	1.534
SCA8-7	1217.40	1068.56	1052.17	1051.28	1051.28	1051.28	1061.00	1083.53	1051.28	2.976
SCA8-8	1231.60	1080.58	1071.18	1071.18	1071.18	1071.18	1071.18	1116.35	1071.18	4.046
SCA8-9	1185.60	1084.80	1060.50	1060.50	1060.50	1060.50	1060.50	1132.16	1060.50	6.329
CON3-0	672.40	631.39	616.52	616.52	616.52	616.52	616.52	639.81	616.52	3.640
CON3-1	570.60	554.47	554.47	554.47	554.47	554.47	554.47	581.44	554.47	4.638
CON3-2	534.80	522.86	518.00	518.00	518.00	518.00	518.00	531.60	518.00	2.558
CON3-3	656.90	591.19	591.19	591.19	591.19	591.19	591.19	611.21	591.19	3.275
CON3-4	640.20	591.12	588.79	588.79	588.79	588.79	588.79	619.94	588.79	5.025
CON3-5	604.70	563.70	563.70	563.70	563.70	563.70	563.70	607.87	563.70	7.266

Instance	Solution									Improvement (%)
	S-1	S-2	S-3	S-4	S-5	S-6	S-7	ICW-1	ICW-2	
CON3-6	521.30	506.19	499.05	499.05	499.05	499.05	500.80	521.92	499.05	4.382
CON3-7	602.80	577.68	576.48	576.48	576.48	576.48	576.48	603.41	576.48	4.463
CON3-8	556.20	523.05	523.05	523.05	523.05	523.05	523.05	542.45	523.14	3.560
CON3-9	612.80	580.05	578.24	578.24	578.24	578.24	578.24	604.99	578.24	4.422
CON8-0	967.30	860.48	857.17	857.17	857.17	857.17	857.17	865.66	857.17	0.981
CON8-1	828.70	740.85	740.85	740.85	740.85	740.85	740.85	756.11	740.85	2.018
CON8-2	770.20	723.32	712.89	712.89	712.89	712.89	712.89	734.84	712.89	2.987
CON8-3	906.70	811.23	811.07	811.07	811.07	811.07	811.07	818.62	811.07	0.922
CON8-4	876.80	772.25	772.25	772.25	772.25	772.25	772.25	797.64	772.32	3.174
CON8-5	866.90	756.91	754.88	754.88	754.88	754.88	755.70	766.02	754.88	1.454
CON8-6	749.10	678.92	678.92	678.92	678.92	678.92	678.92	704.06	678.92	3.571
CON8-7	929.80	814.50	811.96	811.96	811.96	811.96	814.80	824.72	811.96	1.547
CON8-8	833.10	775.59	767.53	767.53	767.53	767.53	767.53	791.08	767.53	2.977
CON8-9	877.30	809.00	809.00	809.00	809.00	809.00	809.00	829.92	809.00	2.521

References

Altinel, İ.K. & Öncan, T. (2005). A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. *The Journal of the Operational Research Society, 56*, 954-961.

Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y. & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society, 53*, 512-522.

Clarke, G. & Wright, J.W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*, 568-581.

Dantzig, G.B. & Ramser, J.H. (1959). The truck dispatching problem. *Management Science, 6*, 80-91.

Jun, Y. & Kim, B.I. (2012). New best solutions to VRPSPD benchmark problems by a perturbation based algorithm. *Expert Systems with Applications, 39*, 5641-5648.

Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spektrum, 23*, 79-96.

Doyuran, T. & Çatay, B. (2011). A robust enhancement to the Clarke-Wright savings algorithm. *Journal of the Operational Research Society, 62*, 223-231.

Gajpal, Y. & Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research, 36*, 3215-3223.

Goksal, F.P., Karaoglan, I. & Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering, 65*, 39-53.

Juan, A.A., Faulin, J., Jorba, J., Riera, D., Masip, D. & Barrios, B. On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics. *Journal of the Operational Research Society, 62*, 1085-1097.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research A, 23*, 377-386.

Pichpibul, T. & Kawtummachai, R. (2012). An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem. *ScienceAsia, 38,* 307-318.

Souza, M.J.F., Mine, M.T., Silva, M.S.A., Ochi, L.S. & Subramanian, A. (2011). A hybrid heuristic, based on Iterated Local Search and GENIUS, for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Int. J. Logistics Systems and Management*, *10*, 142-157.

Subramanian, A., Uchoa, E., Pessoa, A.A. & Ochi, L.S. (2013). Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optim Lett*, *7*, 1569-1581.

Tang, F.A. & Galvao, R.D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers and Operations Research, 33*, 595-619.

Yousefikhoshbakht, M., Didehvar, F. & Rahmati, F. (2014). A combination of modified tabu search and elite ant system to solve the vehicle routing problem with simultaneous pickup and delivery. *Journal of Industrial and Production Engineering, 31*, 65-75.