European Journal of Scientific Research
ISSN 1450-216X Vol.78 No.1 (2012), pp.119-134

http://www.europeanjournalofscientificresearch.com

New Enhancement for Clarke-Wright Savings Algorithm to Optimize the Capacitated Vehicle Routing Problem

Tantikorn Pichpibul
School of Manufacturing Systems and Mechanical Engineering
Sirindhorn International Institute of Technology, Thammasat University
P O Box 22 Thammasat-Rangsit, Klong Luang, Pathumthani 12121, Thailand
E-mail: pichpibul@gmail.com

Ruengsak Kawtummachai
Corresponding Author, Associate Professor, Faculty of Business Administration
Panyapiwat Institute of Management, 85/1 M.2 Chaengwattana Rd
Bangtalad, Pakkred, Nonthaburi 11120, Thailand
E-mail: ruengsak@yahoo.com
Tel: +66-2-832-0285; Fax: +66-2-832-0394

Abstract

The Clarke-Wright savings algorithm has been widely applied as a basis algorithm in many commercial routing packages and also has been one of the most widely known heuristics for solving the capacitated vehicle routing problem since it was published in 1964. The savings concept is to link the routes determined by the savings list in which values are sorted from largest to smallest. In this paper, we concentrate on the findings of determining a savings list in the Clarke-Wright savings algorithm and propose an enhancement for the algorithm. The two-phase probabilistic mechanism and the route post-improvement are proposed. The performance of this enhancement is reported by computational results on well-known benchmark problems in different situations composed of 90 instances involving from 16 to 135 customers.

Keywords: Capacitated Vehicle Routing Problem, Heuristics, Optimization, Clarke-Wright Savings Algorithm, Hybrid Clarke-Wright Savings Algorithm

1. Introduction

The capacitated vehicle routing problem (CVRP) was introduced by Dantzig and Ramser (1959) and has become one of the most important and widely studied problems in the area of combinatorial optimization. It is also known to be an NP-hard problem (Toth and Vigo, 2002). The basic concept of CVRP is to find a feasible set of vehicle routes that minimizes the total traveling distance and the total number of vehicles used. In each route, the vehicle departs from a given depot and returns to the same depot after completing service. CVRP involves a single depot, homogeneous fleets of vehicles with limited capacity, and a set of customers who require delivery of goods from the depot. In a mathematical model, the formulation of CVRP has only one limit on the capacity of the vehicles, (more details are available in Toth and Vigo, 2002; Kanthavel et al., 2012a). The problem, which may contain other real-world constraints, is also discussed in the other variants of VRP such as time windows (Solomon, 1987; Xiaodong, 2012) and simultaneous delivery and pickup (Dethloff, 2001; Kanthavel et al., 2012b). In most of these formulations, it has a large number of variables, and is very complex to find optimal solutions by using an exact algorithm composed of branch-and-bound algorithm (Christofides et al., 1981), branch-and-cut algorithm (Fisher, 1994; Augerat et al., 1995; Lysgaard et al., 2004), and branch-and-cut-and-price algorithm (Fukasawa et al., 2006) involving more than 100 customers due to a huge amount of computation time. This motivates the scientific community from computer science and engineering fields to develop different algorithms to tackle the problem. Most early algorithms for CVRP concentrated on route construction heuristics such as savings algorithm (Clarke and Wright, 1964), sweep algorithm (Wren and Holliday, 1972; Na et al., 2011), and cluster-first and route-second algorithm (Fisher and Jaikumar, 1981). These heuristics quickly produced the feasible solutions which could be improved by local search procedures, such as 2-opt, 3-opt (Lin, 1965), and Or-opt (Or, 1976). However, these heuristic solutions are far from the optimal solutions. Later heuristics that employed intelligent exploring and exploiting techniques to solve CVRP consisting of simulated annealing (Kirkpatrick et al., 1983), tabu search (Gendreau et al., 1994; Rochat and Taillard, 1995), genetic algorithm (Goldberg and Holland, 1988; Gen and Cheng, 1997), particle swarm optimization (Kennedy and Eberhart, 1995; Ai and Kachitvichyanukul, 2009; Kim and Son, 2010; Moghaddam et al., 2012), and ant colony optimization (Dorigo et al., 1996; Bullnheimer et al., 1999), usually produced better solutions than the early heuristics (or even optimal solutions).

The major contribution of our work is to develop a simple but powerful hybrid Clarke-Wright savings algorithm that can efficiently compete with the previous enhancements of Clarke-Wright savings algorithms and other algorithms to solve CVRP in terms of solution quality. For that goal, we test our algorithm on well-known benchmark problems in different situations composed of 90 instances involving from 16 to 135 customers in which the details are presented in section 2. The rest of this paper is organized as follows. In section 3, we review the enhancements of Clarke-Wright savings algorithm to solve CVRP. The proposed Clarke-Wright savings algorithm is described in section 4. Section 5 discusses the computational results. Finally, section 6 presents the conclusions.

2. CVRP Benchmark Problems in Different Situations

The benchmark problem sizes that we highlighted in this paper are classified as small-scale (less than 50 customers) and medium-scale (between 51 to 100 customers). All considered problems are symmetric with different features, e.g., uniformly and not uniformly dispersed customers, clustered and not clustered, with a centered or not centered depot. For all instances, we use the same nomenclature, consisting of a data set identifier, followed by n which represents the number of customers (including depot), and k which represents the maximum number of vehicles used. The problems also include vehicle capacity constraints. The details of each problem are explained in the following subsections.

2.1. Benchmark of Augerat et al. (1995)

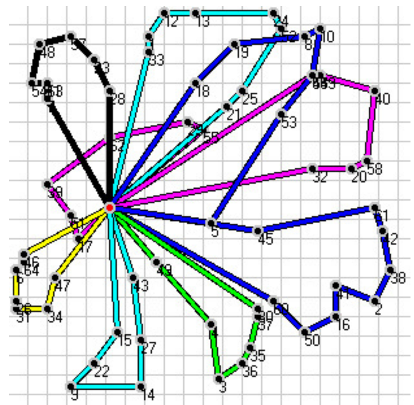
This benchmark proposed by Augerat et al. (1995) is composed of three data sets (A, B, and P). For the instances in data set A, both customer locations and demands are randomly generated by a uniform distribution. The customer locations in data set B are clustered instances. The modified version of other instances is data set P. The example of optimal solutions in data sets A, B, and P is shown in Figures 1 (a) – 1 (c). In this benchmark, the problem ranges in size from 16 to 101 customers, including the depot.

2.2. Benchmark of Christofides and Eilon (1969)

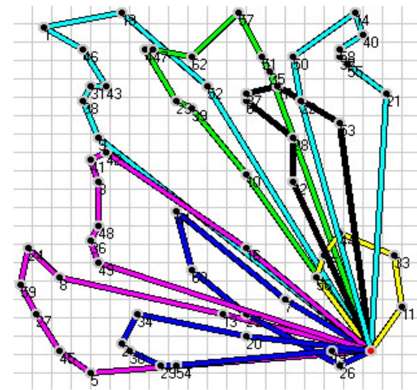
This benchmark proposed by Christofides and Eilon (1969) is composed of one data set (E). For the instances in data set E, the customers are randomly distributed in the plane and the depot is either in the center or near to it. The example of optimal solutions in data set E is shown in Figure 1 (d). In this benchmark, the problem ranges in size from 22 to 101 customers, including the depot.

Figure 1: The example of optimal solutions in data sets A, B, P, E, F, and M

(a) Data set A

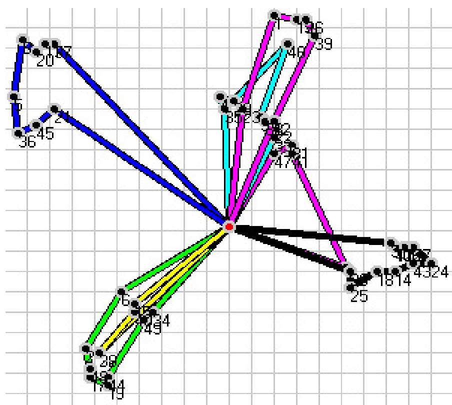


A-n65-k9 (Depot located at left position)

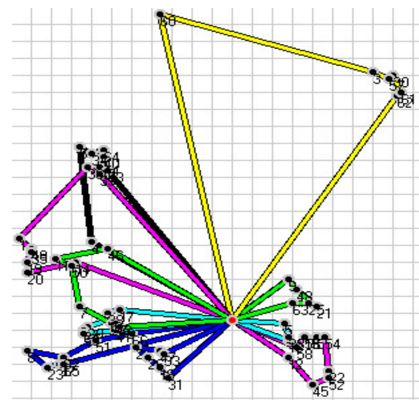


A-n63-k9 (Depot located at bottom-right position)

(b) Data set B

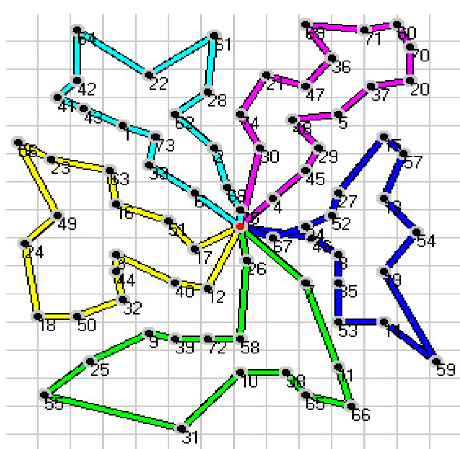


B-n50-k7 (Depot located at center position)

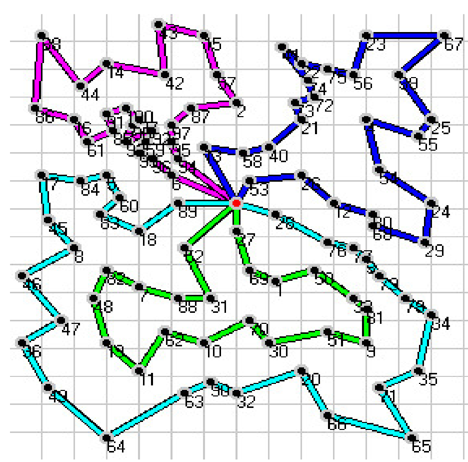


B-n67-k10 (Depot located at bottom-center position)

(c) Data set P



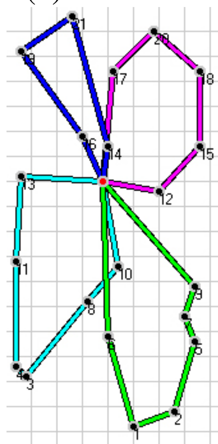
P-n76-k5 (Depot located at center position)

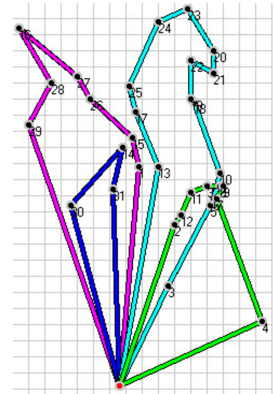


P-n101-k4 (Depot located at near-center position)

Figure 1: The example of optimal solutions in data sets A, B, P, E, F, and M - continued

(d) Data set E

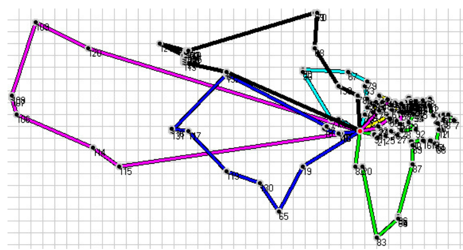




E-n33-k4 (Depot located at bottom-center position)

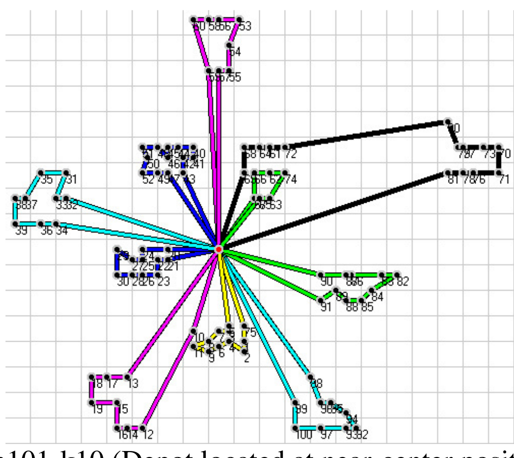
E-n22-k4 (Depot located at near-center position)

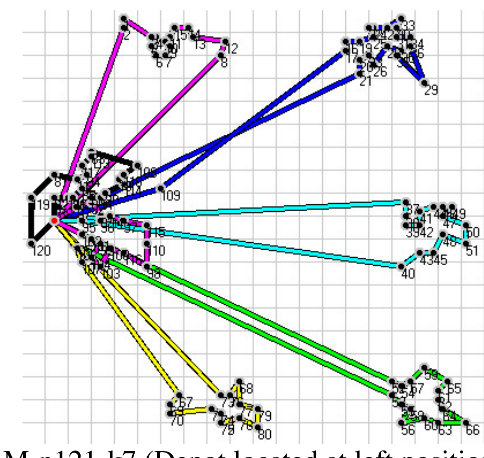
(e) Data set F



F-n135-k7 (Depot located at right position)

(f) Data set M





M-n101-k10 (Depot located at near-center position) M-n121-k7 (Depot located at left position)

2.3. Benchmark of Fisher (1994)

This benchmark proposed by Fisher (1994) is composed of one data set (F). Instances F-n45-k4 and F-n135-k7 represent a day of grocery deliveries from the Peterboro and Bramalea, Ontario terminals, respectively, of National Grocers Limited. The other instance (F-n72-k4) represents the delivery of tires, batteries, and accessories to gasoline service stations. The depot is not centered in all instances.

The example of an optimal solution in data set F is shown in Figure 1 (e). In this benchmark, the problem ranges in size from 45 to 135 customers, including the depot.

2.4. Benchmark of Christofides et al. (1979)

This benchmark proposed by Christofides et al. (1979) is composed of one data set (M). For the instances in data set M, the customers are located randomly in the plane, and are clustered around a non centered depot. The example of optimal solutions in data set M is shown in Figure 1 (f). In this benchmark, the problem ranges in size from 101 to 121 customers, including the depot.

3. The Previous Enhancements of Clarke-Wright Savings Algorithm for CVRP

The savings algorithm (CW), developed by Clarke and Wright (1964), is the most widely known heuristic for solving CVRP. Its concept is to link the routes determined by the savings list in which values are sorted from largest to smallest. The CW was also widely used as a basis in many commercial routing packages. There are two versions of CW: sequential and parallel. In the sequential version, a single route is expanded until no more routes can be linked to it. In the parallel version, several routes can be constructed in parallel. According to Laporte et al. (2000) and Toth and Vigo (2002), the parallel version usually offers better results than the sequential version. Cordeau et al. (2002) mentioned that four attributes of good heuristics are accuracy, speed, simplicity, and flexibility. Extensive testing of heuristic algorithms shows that CW rates high in speed and simplicity but is not the most accurate due to the large deviations between its solutions and the optimal solutions. Therefore, CW was modified by several researchers who proposed the enhancements of CW for CVRP.

Altinel and Öncan (2005) introduced a new enhancement of CW which differs from Gaskell (1967), Yellow (1970), and Paessens (1988) in parameterized saving consisting of a route shape parameter (λ), which controls the relative significance of direct arc between two customers, and a weighted parameter (μ), which can be the use of asymmetry between two customers with respect to the distances to the depot. The new heuristic with savings criterion was considered by customer demand (ν), which includes the demand of customers on a vehicle's capacity. The proposed formulation is as follows:

$$s_{i,j} = c_{1,i} + c_{j,1} - \lambda c_{i,j} + \mu \left| c_{1,i} - c_{j,1} \right| + \nu \frac{d_i + d_j}{\bar{d}} \tag{1}$$

Here $c_{1,i}$ is the traveling distance from depot 1 to customer i, d_i is the demand of customer i, and $\bar{d}$ is the average demand of all customers. In their implementations, they also applied the works of Nelson et al. (1985) who proposed the special data structures to determine the maximum number of savings value. By considering the best value of three parameters (λ, μ, ν), Altinel and Öncan (2005) used a simple enumerative approach to produce 8820 different solutions, and chose the best improvement compared to CW. Therefore, this approach requires much computing time which can be reduced by using the genetic algorithm proposed by Battarra et al. (2008), and empirically adjusted greedy heuristics proposed by Corominas et al. (2010) to adjust the parameters which gave results of similar quality. Juan et al. (2010, 2011) presented a simulation in CVRP via the generalized CW which is a hybrid algorithm that combines the parallel version of CW with Monte Carlo simulation and state-of-the-art random number generators. The most recent enhancement of CW was a robust enhancement to savings formulation proposed by Doyuran and Çatay (2011). They adjusted the savings formulation taken from Gaskell (1967), Yellow (1970), Paessens (1988), and Altinel and Öncan (2005). Moreover, the idea of a sweep algorithm (Wren and Holliday, 1972) was also applied to savings formulation by adding the cosine value of polar coordinate angles of customers with the depot as a coefficient. The proposed formulation is as follows:

$$s_{i,j} = \left[\frac{c_{1,i} + c_{j,1} - \lambda c_{i,j}}{c^{\max}} \right] + \left[\mu \frac{\cos\theta_{i,j} \left| c^{\max} - \left(c_{1,i} - c_{j,1} \right) / 2 \right|}{c^{\max}} \right] + \left[\nu \frac{\left| \bar{d} - \left(d_i + d_j \right) / 2 \right|}{d^{\max}} \right] \tag{2}$$

where $\theta_{i,j}$ is the angle formed by the two rays originating from the depot and crossing the customer i and j, $c^{\max}$ represents the longest distance among all customer pairs, and $d^{\max}$ denotes the maximum demand among all customers.

Our summary of the previous enhancements of CW, since 1967 to 2011, shows that most of them can consecutively increase the improvements between their solutions and CW solutions on the same well-known benchmark problems. The interesting ideas of those works are related to the moves of sequence of customer pairs in the savings list by changing the savings value with different savings formulations (Gaskell, 1967; Yellow, 1970; Paessens, 1988; Altinel and Öncan, 2005; Doyuran and Çatay, 2011). Their results are better than CW results, but are not good enough when compared with the other results. In the literatures, we found that only the works of Juan et al. (2010, 2011) successfully reported 54 optimal solutions by using a probabilistic approach based on Monte Carlo simulation. Therefore, this motivates us to develop a new, competitive approach which can reach more optimal solutions.

4. The Proposed Enhancement of Clarke-Wright Savings Algorithm

The development of our algorithm, which is based on CW, is explained in this section. It looks for high-quality feasible solutions by iteratively applying a two-phase probabilistic procedure on CW and two simple local search approaches on a route post-improvement procedure. These procedures suggested by the authors are simple but effective in producing a near-optimal solution. The work of the proposed CW is given in Figure 2 and is described in the following subsections.

Figure 2: Pseudo code of the proposed CW

```
costmatrix = DoCostMatrix
costsaving = DoCostSaving(costmatrix)
Call DoSort(costsaving)
costlink = DoCostLink(costsaving)
route = DoRoute(costlink)
solution = DoSolution(route)
For iteration = 1 To 1000
   new_costsaving = TwoPhaseProbabilistic(costsaving,0,20)
   costlink = DoCostLink(new_costsaving)
   new_route = DoRoute(costlink)
   new_solution = DoSolution(new_route)
   If new_solution < solution Then
      If IsFeasible(new_solution) = True Then
         route = new_route
         solution = new_solution
         costsaving = new_costsaving
      End If
   End If
Next iteration
improveRoute = RoutePostImprovement(route)
solution = DoSolution(improveRoute)
```

4.1. Classical Savings Algorithm

The classical savings algorithm first calculates the cost savings value for every pair of customers as follows:

$$s_{i,j} = c_{1,i} + c_{j,1} - c_{i,j} \tag{3}$$

Here, $c_{i,j}$ is the traveling cost between customer i and j and the depot is assigned to be customer 1. The cost savings is thus calculated and then ranked in the descending order to the cost savings list. Next, in each value from the list, any two customers i and j are combined to the cost savings link if the total demand does not exceed vehicle capacity. The procedure is repeated to process the next value in the list until no feasible link is possible. Finally, the vehicle routes are constructed through the inclusion of customers i and j in the link. In cases of non-routed customers, each is assigned by a route that begins at the depot, visits the unassigned customer and returns to the same depot.

4.2. Two-Phase Probabilistic Procedure

To escape the greedy nature of CW, the two-phase probabilistic procedure is introduced in this subsection. Instead of ranking the cost savings value in the descending order, we randomly rank the value in the cost savings list. This procedure is an iterative improvement approach which rearranges the new cost savings value by adapting the combination of ideas between tournament selection and roulette wheel selection. Goldberg et al. (1989) gave an example of the tournament selection which randomly chooses a set of chromosomes, represented by the cost savings value in our work, and picks out the best one from the set for reproduction. The number of chromosomes in the set is called tournament size. Fitness proportionate selection or roulette wheel selection was introduced by Holland (1975). His basic idea is to determine the selection or survival probability for each chromosome proportional to the fitness value. The work of this procedure is shown in Figure 3.

Figure 3: Pseudo code of the two-phase probabilistic procedure

```
For no = 1 To UpperBound(costsaving)
  tournamentsize = RandomInteger(0 To 20)
  j = 0
  For i = 1 To UpperBound(costsaving)
    If costsaving(i).deleted = False Then
      j = j + 1: nosaving(j) = i
      If j = tournamentsize Then Exit For i
    End If
  Next i
  sumsaving = 0
  For i = 1 To tournamentsize
    sumsaving = sumsaving + costsaving(nosaving(i)).value
  Next i
  For i = 1 To tournamentsize
    If i = 1 Then
      probability(i) = costsaving(nosaving(i)).value / sumsaving
    Else
      probability(i) = probability(i – 1) + (costsaving(nosaving(i)).value / sumsaving)
    End If
  Next i
  prob = RandomDecimal(0 To 1)
  For i = 1 To tournamentsize
    If prob < probability(i) Then
      new_costsaving(no).value = costsaving(nosaving(i)).value
      costsaving(nosaving(i)).deleted = True
      Exit For i
      End If
      Next i
      Next no
      Return new_costsaving
```

The new cost savings value, replaces the previous cost savings value only if the current solution is better than the previous one. In the case that the pre-determined stopping condition is satisfied, the proposed CW will be terminated. Here, the number of iterations is employed as the stopping condition. Notice that the proposed CW may generate the infeasible solution in which the number of available vehicles is inadequate at some iteration. In this case, the cost savings value will not be selected to be a new one by this procedure.

4.3. Route Post-Improvement Procedure

When the stopping criterion of the two-phase probabilistic procedure is reached, the solution is returned and is then post-improved through the move and swap neighborhoods (Kindervater and Savelsbergh, 1997), where single customers are moved and swapped with equal probability. The move operator selects a customer and moves it to another randomly selected place. The swap operator randomly selects two customers and exchanges them. In order to exhaustively explore the solution spaces, we concentrate on both intra-route and inter-route moves. The work of Figure 4 (a) only concerns the order of customers in the same route. In contrast, the work of Figure 4 (b) considers two different routes simultaneously. Two conditions, consisting of the total number of global and local iterations, are employed as stopping conditions for this procedure.

5. Results and Discussion

The proposed CW was coded in Visual Basic 6.0 on an Intel® Core™ i7 CPU 860 clocked at 2.80 GHz with 1.99 GB of RAM under Windows XP platform. The numerical experiment used well-known benchmark problems composed of 90 instances. First, 74 instances are from Augerat et al. (1995).

Figure 4: Pseudo code of the route post-improvement procedure

(a) intra-route move

```
For iteration_global = 1 To 100
   For i = 1 To UpperBound(route)
      For iteration_local = 1 To 50
         rand = RandomInteger(1 To 2)
         If rand = 1 Then
            cal_route = DoMoveRoute(route(i))
         Else If rand = 2 Then
            cal_route = DoSwapRoute(route(i))
         End If
         cal_solution = DoSolution(cal_route)
         solution = DoSolution(route(i))
         If cal_solution < solution Then
            route(i) = cal_route
         End If
      Next iteration_local
   Next i
Next iteration_global
Return route
```

(b) inter-route move

```
For iteration_global = 1 To 100
   For i = 1 To UpperBound(route) - 1
      For j = i + 1 To UpperBound(route)
         For iteration_local = 1 To 50
            rand = RandomInteger(1 To 2)
            If rand = 1 Then
               rand = RandomInteger(1 To 2)
               If rand = 1 Then
                  cal_route = DoMoveRoute(route(i),route(j))
               Else If rand = 2 Then
                  cal_route = DoMoveRoute(route(j),route(i))
               End If
            Else If rand = 2 Then
               cal_route = DoSwapRoute(route(i),route(j))
            End If
            cal_solution = DoSolution(cal_route(i),cal_route(j))
            solution = DoSolution(route(i),route(j))
            If cal_solution < solution Then
               route(i) = cal_route(i)
               route(j) = cal_route(j)
            End If
         Next iteration_local
      Next j
   Next i
Next iteration_global
Return route
```

Table 1: The CVRP algorithms used to compare with Pichpibul and Kawtummachai (PK) algorithm

Abbreviation	Authors	Algorithm
CW	Clarke and Wright (1964)	Clarke and Wright savings algorithm
P	Paessens (1988)	Modified Clarke and Wright savings algorithm
AÖ	Altinel and Öncan (2005)	New enhancement of the Clarke and Wright savings heuristic
DÇ	Doyuran and Çatay (2011)	Robust enhancement to the Clarke-Wright savings algorithm
JFR	Juan et al. (2010, 2011)	SR-GCWS hybrid algorithm
CYW	Chen et al. (2006)	Hybrid discrete particle swarm optimization algorithm
GN	Ganesh and Narendran (2007)	Cluster-and-search heuristic
AK	Ai and Kachitvichyanukul (2009)	Particle swarm optimization and two solution representations
GPV	Geetha et al. (2010)	Hybrid particle swarm optimization with genetic operators
GGW	Groër et al. (2010)	Library of local search heuristic
KS	Kim and Son (2010)	Probability matrix based particle swarm optimization
NJK	Na et al. (2011)	Extended sweep algorithm
KP	Kanthavel and Prasad (2011)	Nested Particle Swarm Optimization
MRS	Moghaddam et al. (2012)	An advanced particle swarm algorithm

Table 2: The development of Pichpibul and Kawtummachai (PK) algorithm

Abbreviation	Details
PK – 1	Improve CW solution with two-phase probabilistic procedure
PK – 2	Improve PK – 1 solution with route post-improvement procedure

Second, 11 instances are from Christofides and Eilon (1969). Third, three instances are from Fisher (1994). The last two instances are taken from Christofides et al. (1979). For each instance, data and the optimal solution are available online at http://www.branchandcut.org. In order to identify the performance of the proposed CW, we compared it with the algorithms for CVRP as shown in Table 1. Table 2 also describes the development of the proposed CW in detail.

In this section, we notice that the distance rounding rule is represented by an integer number which is rounded distances between customers to the closest integer value, and a decimal number which is the real distance without any rounding. The percentage improvement between obtained solution (*obt*) and CW solution (*CW*) is calculated as follows:

$$\text{Percentage Improvement} = \left(\frac{CW - obt}{CW}\right) \times 100 \tag{4}$$

Moreover, the percentage deviation between obtained solution (*obt*) and the optimal solution (*opt*) is calculated as.

$$\text{Percentage Deviation} = \left(\frac{obt - opt}{opt}\right) \times 100 \tag{5}$$

The numerical results in Tables 3 and 4 show that the proposed CW can find high quality solutions in reasonable computation times. Out of 90 problems, we find the optimal solutions for 76 problems with up to 135 customers. For fourteen problems with 35-100 customers, the percentage deviations between our solutions and the optimal solutions are very low (0.155% for A-n62-k8, 0.714% for A-n64-k9, 0.284% for A-n80-k10, 0.314% for B-n35-k5, 0.147% for B-n45-k6, 0.424% for B-n56-k7, 0.228% for B-n66-k9, 0.328% for B-n78-k10, 0.144% for P-n50-k10, 0.136% for E-n76-k8, 0.602% for E-n76-k10, 0.294% for E-n76-k14, 0.122% for E-n101-k8, 0.560% for E-n101-k14). Nevertheless, in those near optimal solutions, there are three problems (E-n76-k8, E-n101-k8, E-n101-k14) for which our results are better than the compared results. Moreover, there are also four problems (A-n54-k7, A-n63-k10, A-n69-k9, P-n55-k8) where our results not only obtained the optimal solutions but also performed better than the compared results. Furthermore, when we calculate the solution in real distances instead of integer distances, there are 40 problems (A-n34-k5, A-n36-k5, A-n37-k6, A-n39-k5, A-n44-k6, A-n46-k7, A-n48-k7, A-n53-k7, A-n54-k7, A-n62-k8, A-n63-k10, A-n64-k9, An69-k9, B-n34-k5, B-n38-k6, B-n41-k6, B-n43-k6, B-n44-k7, B-n45-k6, B-n50-k8, B-n51-k7, B-n57-k7, B-n63-k10, B-n66-k9, B-n67-k10, P-n16-k8, P-n21-k2, P-n23-k8, P-n45-k5, P-n50-k7, P-n50-k8, P-n55-k7, P-n55-k8, P-n55-k10, P-n60-k15, E-n23-k3, E-n76-k8, E-n101-k8, E-n101-k14, F-n135-k7) where the proposed CW performed better than CW if compared to the previous enhancements of CW. These indicate that the proposed CW is extremely effective and efficient in producing high quality solutions for well-known benchmark problems. The important details of our improvement are as follows.

Table 3: Comparative results with the previous enhancements of CW by using decimal number

No	Instance	Solution						Percentage Improvement				
		CW	P	AÖ	DÇ	JFR	PK	P	AÖ	DÇ	JFR	PK
1	A-n32-k5	843.69	828.70	828.70	828.70	787.08	787.08	1.777	1.777	1.777	6.710	6.710
2	A-n33-k5	712.05	679.72	676.10	676.10	662.11	662.11	4.540	5.049	5.049	7.014	7.014
3	A-n33-k6	776.26	747.32	743.21	746.99	742.69	742.69	3.728	4.258	3.771	4.325	4.325
4	A-n34-k5	810.41	793.05	793.05	793.05	—	780.94	2.142	2.142	2.142	—	3.637
5	A-n36-k5	828.47	806.78	806.78	806.78	—	802.13	2.618	2.618	2.618	—	3.180
6	A-n37-k5	707.81	695.08	694.43	694.44	672.47	672.47	1.799	1.890	1.889	4.993	4.993
7	A-n37-k6	976.61	976.01	974.56	976.61	—	950.85	0.061	0.210	0.000	—	2.637
8	A-n38-k5	768.13	755.94	756.11	755.94	733.95	733.95	1.587	1.565	1.587	4.450	4.450
9	A-n39-k5	901.99	851.25	848.24	843.23	—	828.99	5.625	5.959	6.514	—	8.093
10	A-n39-k6	863.08	849.55	849.56	849.90	833.20	833.20	1.568	1.567	1.528	3.462	3.462
11	A-n44-k6	976.04	968.84	959.43	957.03	—	938.18	0.737	1.702	1.947	—	3.879
12	A-n45-k6	1006.45	957.05	957.06	957.06	944.88	944.88	4.908	4.907	4.907	6.118	6.118
13	A-n45-k7	1199.98	1169.00	1166.39	1168.97	1146.77	1146.77	2.581	2.799	2.584	4.434	4.434
14	A-n46-k7	939.74	933.66	933.66	929.42	—	917.72	0.647	0.647	1.099	—	2.344
15	A-n48-k7	1112.82	1104.23	1104.24	1103.99	—	1074.34	0.772	0.771	0.794	—	3.458
16	A-n53-k7	1099.45	1045.98	1045.47	1048.79	—	1012.33	4.864	4.910	4.608	—	7.924
17	A-n54-k7	1197.92	1188.64	1173.77	1172.27	—	1171.68	0.775	2.016	2.141	—	2.190
18	A-n55-k9	1099.84	1099.55	1098.51	1099.56	1074.46	1074.46	0.026	0.121	0.025	2.308	2.308
19	A-n60-k9	1421.88	1389.59	1376.20	1379.86	1355.80	1355.80	2.271	3.213	2.955	4.648	4.648
20	A-n61-k9	1102.23	1051.37	1051.10	1051.06	1039.08	1039.08	4.614	4.638	4.642	5.729	5.729
21	A-n62-k8	1352.81	1351.11	1347.87	1326.54	—	1296.58	0.126	0.365	1.942	—	4.157
22	A-n63-k9	1687.96	1648.92	1649.14	1652.42	1622.14	1622.14	2.313	2.300	2.106	3.900	3.900
23	A-n63-k10	1352.48	1349.58	1348.17	1347.30	—	1313.46	0.214	0.319	0.383	—	2.885
24	A-n64-k9	1486.92	1442.44	1439.75	1442.66	—	1410.87	2.991	3.172	2.977	—	5.114
25	A-n65-k9	1239.42	1224.71	1202.08	1197.49	1181.69	1181.69	1.187	3.013	3.383	4.658	4.658
26	A-n69-k9	1210.78	1185.08	1185.08	1181.91	—	1166.08	2.123	2.123	2.384	—	3.692
27	A-n80-k10	1860.94	1818.64	1816.78	1811.56	1766.50	1770.49	2.273	2.373	2.654	5.075	4.861
28	B-n31-k5	681.16	679.43	677.34	676.50	676.09	676.09	0.254	0.561	0.684	0.744	0.744
29	B-n34-k5	794.33	789.84	789.85	789.85	—	789.84	0.565	0.564	0.564	—	0.565
30	B-n35-k5	978.33	978.32	975.48	973.27	956.29	958.89	0.001	0.291	0.517	2.253	1.987
31	B-n38-k6	832.09	824.00	824.00	820.31	—	807.88	0.972	0.972	1.416	—	2.910
32	B-n39-k5	566.71	554.99	555.00	554.35	553.16	553.16	2.068	2.066	2.181	2.391	2.391
33	B-n41-k6	898.09	867.42	867.42	852.95	834.92	833.66	3.415	3.415	5.026	7.034	7.174
34	B-n43-k6	781.96	754.04	754.92	756.07	—	746.69	3.571	3.458	3.311	—	4.510
35	B-n44-k7	937.74	932.32	934.68	930.99	—	914.96	0.578	0.326	0.720	—	2.429
36	B-n45-k5	757.16	757.16	754.71	756.60	754.22	754.25	0.000	0.324	0.074	0.388	0.384
37	B-n45-k6	727.84	713.24	713.24	717.24	—	681.72	2.006	2.006	1.456	—	6.336
38	B-n50-k7	748.80	747.92	745.37	744.77	744.23	744.23	0.118	0.458	0.538	0.610	0.610
39	B-n50-k8	1354.03	1339.44	1338.34	1337.13	—	1315.38	1.078	1.159	1.248	—	2.855
40	B-n51-k7	1059.86	1050.00	1050.00	1043.58	—	1035.04	0.930	0.930	1.536	—	2.342
41	B-n52-k7	764.90	763.96	756.90	762.16	749.96	750.04	0.123	1.046	0.358	1.953	1.943
42	B-n56-k7	733.74	723.76	722.61	722.62	712.92	716.42	1.360	1.517	1.516	2.838	2.360
43	B-n57-k7	1239.78	1148.97	1148.98	1150.77	—	1143.33	7.325	7.324	7.179	—	7.780
44	B-n57-k9	1653.42	1619.71	1619.72	1613.27	1602.28	1602.29	2.039	2.038	2.428	3.093	3.092
45	B-n63-k10	1598.18	1562.59	1562.59	1552.36	—	1500.93	2.227	2.227	2.867	—	6.085
46	B-n64-k9	921.56	919.37	910.07	907.30	868.31	868.31	0.238	1.247	1.547	5.778	5.778
47	B-n66-k9	1416.42	1372.09	1358.32	1357.17	—	1327.51	3.130	4.102	4.183	—	6.277
48	B-n67-k10	1099.95	1090.18	1070.30	1066.79	1039.36	1039.27	0.888	2.696	3.015	5.508	5.517
49	B-n68-k9	1317.77	1317.77	1316.07	1315.76	1276.20	1277.69	0.000	0.129	0.153	3.155	3.042
50	B-n78-k10	1264.56	1263.05	1261.35	1260.50	1227.90	1229.74	0.119	0.254	0.321	2.899	2.753

Table 3: Comparative results with the previous enhancements of CW by using decimal number - continued

51	P-n16-k8	478.77	451.94	451.94	451.94	—	451.34	5.604	5.604	5.604	—	5.729
52	P-n19-k2	237.89	220.64	220.64	220.64	212.66	212.66	7.251	7.251	7.251	10.606	10.606
53	P-n20-k2	234.00	233.99	232.86	224.13	217.42	217.42	0.004	0.487	4.218	7.085	7.085
54	P-n21-k2	236.19	236.18	231.54	212.71	—	212.71	0.004	1.969	9.941	—	9.941
55	P-n22-k2	239.50	219.89	219.89	217.87	217.85	217.85	8.188	8.188	9.031	9.040	9.040
56	P-n22-k8	590.62	589.39	589.39	588.79	588.79	588.79	0.208	0.208	0.310	0.310	0.310
57	P-n23-k8	539.48	536.71	536.71	536.35	—	531.17	0.513	0.513	0.580	—	1.540
58	P-n40-k5	518.37	468.20	468.20	470.20	461.73	461.73	9.678	9.678	9.293	10.927	10.927
59	P-n45-k5	572.95	523.91	522.41	521.31	—	512.79	8.559	8.821	9.013	—	10.500
60	P-n50-k7	597.03	578.94	577.73	577.73	—	559.86	3.030	3.233	3.233	—	6.226
61	P-n50-k8	674.34	646.54	646.55	646.55	633.96	632.70	4.123	4.121	4.121	5.988	6.176
62	P-n50-k10	739.84	712.77	712.77	712.77	699.56	700.66	3.659	3.659	3.659	5.444	5.296
63	P-n51-k10	790.97	754.97	754.98	747.25	741.50	741.50	4.551	4.550	5.527	6.254	6.254
64	P-n55-k7	618.68	589.54	587.44	584.23	—	570.27	4.710	5.049	5.568	—	7.825
65	P-n55-k8	631.67	594.84	588.04	594.30	—	578.61	5.831	6.907	5.916	—	8.400
66	P-n55-k10	736.45	716.06	715.21	709.33	—	696.83	2.769	2.884	3.683	—	5.380
67	P-n55-k15	978.07	963.32	963.32	959.93	944.56	944.56	1.508	1.508	1.855	3.426	3.426
68	P-n60-k10	800.19	769.27	768.12	765.08	748.07	748.07	3.865	4.009	4.389	6.515	6.515
69	P-n60-k15	1016.96	1006.94	1002.77	996.87	—	971.58	0.985	1.395	1.975	—	4.462
70	P-n65-k10	844.61	829.17	825.92	815.96	795.66	795.66	1.828	2.213	3.392	5.796	5.796
71	P-n70-k10	896.86	853.94	853.94	855.10	829.93	829.93	4.786	4.786	4.656	7.463	7.463
72	P-n76-k4	688.34	643.14	641.78	616.30	598.19	598.20	6.567	6.764	10.466	13.097	13.095
73	P-n76-k5	709.38	655.03	652.93	647.31	633.32	633.32	7.662	7.958	8.750	10.722	10.722
74	P-n101-k4	765.38	722.83	711.03	702.04	691.29	691.29	5.559	7.101	8.276	9.680	9.680
75	E-n22-k4	388.77	375.28	375.28	375.28	375.28	375.28	3.470	3.470	3.470	3.470	3.470
76	E-n23-k3	621.09	573.01	573.01	573.01	—	568.56	7.741	7.741	7.741	—	8.458
77	E-n30-k3	534.45	506.67	506.67	507.51	505.01	505.01	5.198	5.198	5.041	5.508	5.508
78	E-n33-k4	843.10	843.09	843.10	842.83	837.67	837.67	0.001	0.000	0.032	0.644	0.644
79	E-n51-k5	584.64	566.10	555.55	537.29	524.61	524.61	3.171	4.976	8.099	10.268	10.268
80	E-n76-k7	738.13	718.88	718.88	716.48	687.60	687.80	2.608	2.608	2.933	6.846	6.819
81	E-n76-k8	794.74	783.12	779.42	768.05	—	742.08	1.462	1.928	3.358	—	6.626
82	E-n76-k10	907.39	866.29	860.21	864.29	835.26	838.60	4.529	5.200	4.750	7.949	7.581
83	E-n76-k14	1054.60	1052.30	1045.04	1049.31	1024.40	1030.52	0.218	0.907	0.502	2.864	2.283
84	E-n101-k8	889.00	865.60	867.35	854.49	—	830.89	2.632	2.435	3.882	—	6.536
85	E-n101-k14	1139.07	1133.99	1126.39	1127.01	—	1090.89	0.446	1.113	1.059	—	4.229
86	F-n45-k4	739.02	—	—	—	723.54	723.54	—	—	—	2.094	2.094
87	F-n72-k4	256.19	—	—	—	241.97	241.97	—	—	—	5.549	5.549
88	F-n135-k7	1219.32	—	—	—	1164.73	1164.54	—	—	—	4.477	4.493
89	M-n101-k10	833.51	826.00	824.66	825.76	819.56	819.56	0.901	1.062	0.930	1.674	1.674
90	M-n121-k7	1068.14	1066.40	1057.80	1059.87	1043.88	1043.89	0.163	0.968	0.774	2.271	2.271

The average percentage improvement between CW solutions and our solutions for benchmark of data sets A, B, P, E, F and M are 4.474%, 3.472%, 7.183%, 5.642%, 4.045% and 1.973%, respectively. We have found that CW solutions were improved by the average of 4.465%. This finding shows that data set P has the highest average deviation and data set M has the lowest average deviation. We can conclude that the problems which have the features like clustered customers can be solved by CW better than the problems which have the features like uniformly dispersed customers with a centered depot. In addition, the results show that the proposed CW can solve both above-mentioned problems to obtain optimal or near optimal solutions.

6. Conclusions and Further Work

In this paper, we present a new enhancement for Clarke-Wright savings algorithm to optimize the capacitated vehicle routing problem. We modified the Clarke-Wright savings algorithm with two procedures consisting of two-phase probabilistic and route post-improvement procedures in which the neighborhood structures composed of move and swap operators are used to improve the last best solution. We also have done experiments using well-known benchmark problems composed of 90 instances obtained from the literatures and have compared them with the optimal solutions.

Table 4: Comparative results with the other algorithms by using integer number

No	Instance	Optimal Solution	CYW	GN	AK	GPV	GGW	KS	JFR	NJK	KP	MRS	PK	
													Solution	Time (s)
1	A-n32-k5	784	—	784	—	—	—	—	784	810	—	784	784	1.261
2	A-n33-k5	661	661	661	661	661	—	661	661	686	661	661	661	1.891
3	A-n33-k6	742	—	742	—	—	—	—	742	743	—	742	742	1.483
4	A-n34-k5	778	—	778	—	—	—	—	—	785	—	778	778	2.980
5	A-n36-k5	799	—	799	—	—	—	—	—	826	—	799	799	1.823
6	A-n37-k5	669	—	669	—	—	—	—	669	670	—	669	669	2.459
7	A-n37-k6	949	—	949	—	—	—	—	—	962	—	949	949	1.850
8	A-n38-k5	730	—	730	—	—	—	—	730	749	—	730	730	3.295
9	A-n39-k5	822	—	822	—	—	—	—	—	—	—	822	822	2.744
10	A-n39-k6	831	—	831	—	—	—	—	831	856	—	—	831	2.140
11	A-n44-k6	937	—	937	—	—	941	—	—	957	—	937	937	3.190
12	A-n45-k6	944	—	944	—	—	948	—	944	991	—	—	944	4.579
13	A-n45-k7	1146	—	1146	—	—	—	—	1146	1173	—	—	1146	3.918
14	A-n46-k7	914	914	914	914	921	914	914	—	946	914	914	914	4.793
15	A-n48-k7	1073	—	1073	—	—	1073	—	—	1113	—	—	1073	5.842
16	A-n53-k7	1010	—	1017	—	—	1010	—	—	—	—	—	1010	5.441
17	A-n54-k7	1167	—	1172	—	—	—	—	—	—	—	—	1167	6.342
18	A-n55-k9	1073	—	1073	—	—	—	—	1073	1095	—	—	1073	6.274
19	A-n60-k9	1354	1354	1358	1355	1368	—	1354	1354	1420	1354	1354	1354	8.285
20	A-n61-k9	1034	—	1038	—	—	—	—	1034	1100	—	—	1034	8.981
21	A-n62-k8	1288	—	1288	—	—	—	—	—	1359	—	—	1290	11.321
22	A-n63-k9	1616	—	1627	—	—	—	—	1616	1712	—	—	1616	9.747
23	A-n63-k10	1314	—	1322	—	—	—	—	—	1386	—	—	1314	12.574
24	A-n64-k9	1401	—	1410	—	—	—	—	—	1499	—	—	1411	10.713
25	A-n65-k9	1174	—	1177	—	—	—	—	1174	1223	—	—	1174	14.670
26	A-n69-k9	1159	—	1163	—	—	—	—	—	1207	—	—	1159	12.627
27	A-n80-k10	1763	—	1780	—	—	—	—	1763	1866	—	—	1768	31.162
28	B-n31-k5	672	—	672	—	—	—	—	672	677	—	672	672	1.267
29	B-n34-k5	788	—	788	—	—	—	—	—	802	—	788	788	1.705
30	B-n35-k5	955	955	955	955	955	—	955	955	962	955	955	958	1.570
31	B-n38-k6	805	—	805	—	—	—	—	—	817	—	805	805	2.514
32	B-n39-k5	549	—	549	—	—	—	—	549	575	—	549	549	2.477
33	B-n41-k6	829	—	829	—	—	839	—	829	843	—	829	829	2.308
34	B-n43-k6	742	—	742	—	—	742	—	—	746	—	742	742	3.158
35	B-n44-k7	909	—	909	—	—	909	—	—	942	—	909	909	3.182
36	B-n45-k5	751	751	751	751	754	—	751	751	797	751	751	751	3.329
37	B-n45-k6	678	—	678	—	—	—	—	—	732	—	678	679	4.502
38	B-n50-k7	741	—	741	—	—	741	—	741	779	—	741	741	3.605
39	B-n50-k8	1312	—	1318	—	—	—	—	—	1349	—	1312	1312	4.895
40	B-n51-k7	1032	—	1032	1032	—	1032	—	—	—	—	1032	1032	4.999
41	B-n52-k7	747	—	747	—	—	747	—	747	758	—	747	747	4.436
42	B-n56-k7	707	—	710	—	—	707	—	707	726	—	707	710	5.372
43	B-n57-k7	1155	—	1193	1155	—	—	—	—	—	—	1155	1155	6.894
44	B-n57-k9	1598	—	1599	—	—	—	—	1598	1642	—	1598	1598	8.028
45	B-n63-k10	1496	—	1510	1499	—	—	—	—	—	—	1496	1496	7.862
46	B-n64-k9	861	—	864	—	—	884	—	861	1161	—	863	861	20.228
47	B-n66-k9	1316	—	—	—	—	—	—	—	1363	—	1316	1319	15.839
48	B-n67-k10	1032	—	1037	—	—	—	—	1032	1080	—	1035	1032	10.675
49	B-n68-k9	1272	1272	1275	1274	1281	—	1275	1272	1308	1272	1272	1272	20.423
50	B-n78-k10	1221	1239	1260	1223	—	—	1223	1221	1268	1221	1221	1225	21.282
51	P-n16-k8	450	—	450	—	—	—	—	—	513	—	—	450	0.543
52	P-n19-k2	212	—	212	—	—	—	—	212	219	—	—	212	0.651
53	P-n20-k2	216	—	216	—	—	—	—	216	217	—	—	216	0.649
54	P-n21-k2	211	—	211	—	—	—	—	—	211	—	—	211	0.665
55	P-n22-k2	216	—	216	—	—	—	—	216	216	—	—	216	0.695
56	P-n22-k8	603	—	603	—	—	—	—	603	560	—	—	603	5.417
57	P-n23-k8	529	—	529	—	—	—	—	—	554	—	529	529	8.450
58	P-n40-k5	458	—	458	—	—	—	—	458	467	—	458	458	2.425
59	P-n45-k5	510	—	510	—	—	—	—	—	—	—	510	510	2.829
60	P-n50-k7	554	—	554	—	—	—	—	—	—	—	554	554	3.785
61	P-n50-k8	631	—	643	—	—	—	—	631	—	—	631	631	9.200

Table 4: Comparative results with the other algorithms by using integer number - continued

62	P-n50-k10	696	—	696	—	—	—	—	696	—	—	696	697	3.737
63	P-n51-k10	741	—	741	—	—	—	—	741	—	—	741	741	4.115
64	P-n55-k7	568	—	568	—	—	—	—	—	—	—	568	568	5.865
65	P-n55-k8	576	—	—	—	—	—	—	—	—	—	588	576	5.552
66	P-n55-k10	694	—	698	—	—	—	—	—	—	—	694	694	4.887
67	P-n55-k15	989	—	989	995	—	—	—	989	—	—	989	989	6.405
68	P-n60-k10	744	—	744	—	—	—	—	744	—	—	744	744	6.327
69	P-n60-k15	968	—	968	—	—	—	—	—	—	—	968	968	6.628
70	P-n65-k10	792	—	800	—	—	—	—	792	—	—	792	792	7.794
71	P-n70-k10	827	—	827	—	—	—	—	827	—	—	827	827	14.141
72	P-n76-k4	593	602	593	594	610	—	594	593	612	593	593	593	25.255
73	P-n76-k5	627	—	627	—	—	—	—	627	—	—	629	627	13.606
74	P-n101-k4	681	694	687	683	—	—	683	681	715	681	688	681	41.125
75	E-n22-k4	375	—	375	—	—	—	—	375	375	—	—	375	0.474
76	E-n23-k3	569	—	569	—	—	—	—	—	569	—	—	569	0.501
77	E-n30-k3	534	534	534	534	534	—	534	534	543	534	534	534	1.952
78	E-n33-k4	835	—	835	—	—	—	—	835	852	—	—	835	1.402
79	E-n51-k5	521	528	521	521	522	521	521	521	532	521	521	521	3.962
80	E-n76-k7	682	688	690	682	—	—	687	682	703	682	682	682	12.301
81	E-n76-k8	735	—	738	—	—	—	—	—	746	—	—	736	15.207
82	E-n76-k10	830	—	867	—	—	—	—	830	907	—	—	835	13.854
83	E-n76-k14	1021	—	1032	—	—	—	—	1021	1072	—	—	1024	10.096
84	E-n101-k8	817	—	830	—	—	—	—	—	850	—	—	818	33.005
85	E-n101-k14	1071	—	1099	—	—	—	—	—	1152	—	—	1077	37.768
86	F-n45-k4	721	—	—	—	—	—	—	721	750	—	—	721	3.072
87	F-n72-k4	237	244	—	237	253	—	237	237	241	237	237	237	15.856
88	F-n135-k7	1162	1215	—	1162	—	—	1170	1162	1221	1162	1167	1162	62.030
89	M-n101-k10	820	824	—	820	—	—	820	820	—	820	821	820	9.931
90	M-n121-k7	1034	1038	—	1036	—	—	1034	1034	—	1034	1035	1034	37.238

The computational results show that our algorithm is truly efficient and satisfactory in terms of solution quality. It is competitive or outperforms the previous enhancements of Clarke-Wright savings algorithm and generates the optimal solution in 84% of all tested instances (76 out of 90). Moreover, our solutions also get small average percentage deviation when compared with the optimal one (0.049%). Therefore, we can conclude that the performance of our algorithm is excellent when compared with many other algorithms in each instance.

During the development, we have mentioned the ideas related to our algorithm that deserve more attention in further studies. Consequently, it may be interesting to study a more powerful post-improvement procedure which can be adapted to improve the existing algorithm. Another further study is to extend our algorithm to deal with other variants of the studied problems such as time windows, and simultaneous delivery and pickup.

Acknowledgements

The first author expresses gratitude to Panyapiwat Institute of Management, Thailand for providing the financial support for his Ph.D. study in Sirindhorn International Institute of Technology, Thammasat University, Thailand.

References

[1] Ai, T.J. and V., Kachitvichyanukul, 2009. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers & Industrial Engineering* 56, pp. 380-387.

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

[3] Augerat, P., J., Belenguer, E., Benavent, A., Corbern, D., Naddef and G., Rinaldi, 1995. "Computational results with a branch and cut code for the capacitated vehicle routing problem", *Research Report* 949-M, Universite Joseph Fourier, Grenoble, France.

[4] Battarra, M., B., Golden and D., Vigo, 2008. Tuning a parametric Clarke-Wright heuristic via a genetic algorithm. *Journal of the Operational Research Society* 59, pp. 1568-1572.

[5] Bullnheimer, B., R.F., Hartl and C., Strauss, 1999. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research* 89, pp. 319–328.

[6] Chen, A., G., Yang and Z., Wu, 2006. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *Journal of Zhejiang University Science A* 7, pp. 607-614.

[7] Christofides, N. and S., Eilon, 1969. An algorithm for the vehicle dispatching problem. *Operational Research Quarterly* 20, pp. 309–318.

[8] Christofides, N., A., Mingozzi and P., Toth, 1979. "The vehicle routing problem", *Combinatorial optimization*, Wiley, New York, pp. 315-338.

[9] Christofides, N., A., Mingozzi and P., Toth, 1981. Exact algorithm for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming* 20, pp. 255-282.

[10] Clarke, G. and J.W., Wright, 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12, pp. 568-581.

[11] Cordeau, J.F., M., Gendreau, G., Laporte, J.Y., Potvin and F., Semet, 2002. A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53, pp. 512-522.

[12] Corominas, A., A., García-Villoria and R., Pastor, 2010. Fine-tuning a parametric Clarke and Wright heuristic by means of EAGH (empirically adjusted greedy heuristics). *Journal of the Operational Research Society* 61, pp. 1309-1314.

[13] Dantzig, G.B. and J.H., Ramser, 1959. The truck dispatching problem. *Management Science* 6, pp. 80-91.

[14] Dethloff, J., 2001. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spectrum* 23, pp. 79-96.

[15] Dorigo, M., V., Maniezzo and A., Colorni, 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Mans, and Cybernetics* 1, pp. 29-41.

[16] Doyuran, T. and B., Çatay, 2011. A robust enhancement to the Clarke-Wright savings algorithm. *Journal of the Operational Research Society* 62, pp. 223-231.

[17] Fisher, M.L., 1994. Optimal solution of vehicle routing problems using minimum K-trees. *Operations Research* 42, pp. 626–642.

[18] Fisher, M.L. and R., Jaikumar, 1981. A generalized assignment heuristic for vehicle routing, *Networks* 11, pp. 109-124.

[19] Fukasawa, R., H., Longo, J., Lysgaard, M., Poggi de Aragão, M., Reis, E., Uchoa and R.F., Werneck, 2006. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming Series A* 106, pp. 491-511.

[20] Ganesh, K. and T.T., Narendran, 2007. CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up. *European Journal of Operational Research* 178, pp. 699-717.

[21] Gaskell, T.J., 1967. Bases for vehicle fleet scheduling. *Operational Research Quarterly* 18, pp. 281-295.

[22] Geetha, S., G., Poonthalir and P.T., Vanathi, 2010. A hybrid particle swarm optimization with genetic operators for vehicle routing problem. *Journal of Advances in Information Technology* 1, pp. 181-188.

[23] Gen, M. and R., Cheng, 1997. “Genetic algorithms and engineering design”, Wiley, New York.

[24] Gendreau, M., A., Hertz and G., Laporte, 1994. A tabu search heuristic for the vehicle routing problem. *Management Science* 40, pp. 1276-1290.

[25] Goldberg, D.E. and J.H., Holland, 1988. Genetic algorithms and machine learning. *Machine Learning* 3, pp. 95-99.

[26] Goldberg, D., B., Korb and K., Deb, 1989. Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems* 3, pp. 493-530.

[27] Groër, C., B., Golden and E., Wasil, 2010. A library of local search heuristics for the vehicle routing problem. Mathematical Programming *Computation* 2, pp. 79-101.

[28] Holland, J., 1975. “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor.

[29] Juan, A.A., J., Faulin, R., Ruiz, B., Barrios and S., Caballé, 2010. The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. *Applied Soft Computing* 10, pp. 215-224.

[30] Juan, A.A., J., Faulin, J., Jorba, D., Riera, D., Masip and B., Barrios, 2011. 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, pp. 1085-1097.

[31] Kanthavel, K. and P., Prasad, 2011, Optimization of Capacitated Vehicle Routing Problem by Nested Particle Swarm Optimization. *American Journal of Applied Sciences* 8, pp. 107-112.

[32] Kanthavel K., P.S.S., Prasad and K.P., Vignesh, 2012a. A Study of Sweep Algorithm Based Candidate List Performance in Capacitated Vehicle Routing Problem using Nested Particle Swarm Optimization. *European Journal of Scientific Research* 67, pp. 467-473.

[33] Kanthavel K., P.S.S., Prasad and K.P., Vignesh, 2012b. Optimization of Vehicle Routing Problem with Simultaneous Delivery and Pickup using Nested Particle Swarm Optimization. *European Journal of Scientific Research* 73, pp. 331-337.

[34] Kennedy, J. and R., Eberhart, 1995. “Particle swarm optimization”, *In Proceedings of IEEE International Conference on Neural Networks*, pp. 1942-1948.

[35] Kim, B.I. and S.J., Son, 2010. A probability matrix based particle swarm optimization for the capacitated vehicle routing problem. *Journal of Intelligent Manufacturing*, doi:10.1007/s10845-010-0455-7.

[36] Kindervater, G.A.P. and M.W.P., Savelsbergh, 1997. “Vehicle routing: Handling edges exchanges windows”, In: E., Aarts and J.K., Lenstra (Eds.), *Local Search in Combinatorial Optimization*, John Wiley & Sons Ltd., Chichester, pp. 337-360.

[37] Kirkpatrick, S., J., Gelatt and M., Vecchi, 1983. Optimization by simulated annealing. *Science* 220, pp. 671-680.

[38] Laporte, G., M., Gendreau, J., Potvin and F., Semet, 2000. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research* 7, pp. 285-300.

[39] Lin, S., 1965. Computer solutions of the travelling salesman problem. *Bell System Technical Journal* 44, pp. 2245-2269.

[40] Lysgaard, J., A.N., Letchford and R.W., Eglese, 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming Series A* 100, pp. 423-445.

[41] Moghaddam, B.F., R., Ruiz and S.J., Sadjadi, 2012. Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers & Industrial Engineering* 62, pp. 306-317.

[42] Na, B., Jun Y. and Kim B.I., 2011. Some extensions to the sweep algorithm. *International Journal of Advanced Manufacturing Technology* 56, pp. 1057-1067.

[43] Nelson, M.D., K.E., Nygard, J.H., Griffin and W.E., Shreve, 1985. Implementation techniques for the vehicle routing problem. *Computers & Operations Research* 12, pp. 273-283.

[44] Or, I., 1976. "Traveling Salesman-Type Combinatorial Optimization Problems and their Relation to the Logistics of Regional Blood Banking", Ph.D. Dissertation, Northwestern University.

[45] Paessens, H., 1988. The savings algorithm for the vehicle routing problem. *European Journal of Operational Research* 34, pp. 336-344.

[46] Rochat, Y. and É.D., Taillard, 1995. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics* 1, pp. 147-167.

[47] Solomon, M.M., 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research* 35, pp. 254-265.

[48] Toth, P. and D., Vigo, 2002. "The vehicle routing problem", *SIAM monographs on discrete mathematics and applications*, SIAM Publishing, Philadelphia, PA.

[49] Wren, A. and A., Holliday, 1972. Computer scheduling of vehicles from one or more depots to a number of delivery points, *Operational Research Quarterly* 23, pp. 333-344.

[50] Xiaodong, L., (2012). A Summarization of Algorithms for Vehicle Routing Problem with Time Windows. *European Journal of Scientific Research* 73, pp. 296-309.

[51] Yellow, P., 1970. A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly* 21, pp. 281–283.