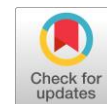# An efficient meta-heuristic algorithm for solving capacitated vehicle routing problem

Alfian Faiz [1,*], Subiyanto [2], Ulfah Mediaty Arief [3]

Electrical Engineering Department, Universitas Negeri Semarang, Indonesia
[1] a.faiz@students.unnes.ac.id; [2] subiyanto@mail.unnes.ac.id; [3] ulfahmediatyarief@mail.unnes.ac.id
* corresponding author

ARTICLE INFO

ABSTRACT

This work aims to develop an enhanced Perturbation based Variable Neighborhood Search with Adaptive Selection Mechanism (PVNS ASM) to solve the capacitated vehicle routing problem (CVRP). This approach combined Perturbation based Variable Neighborhood Search (PVNS) with Adaptive Selection Mechanism (ASM) to control perturbation scheme. Instead of stochastic approach, selection of perturbation scheme used in the algorithm employed an empirical selection based on success rate of each perturbation scheme along the search. The ASM helped algorithm to get more diversification degree and jumping from local optimum condition using most successful perturbation scheme empirically in the search process. A comparative analysis with existing heuristics in the literature has been performed on 21 CVRP benchmarks. The computational results proof that the developed method is competitive and very efficient in achieving high quality solution within reasonable computation time.

1. Introduction

The growing of industry sector has lead various problem in distribution of logistics and supply chain management. Increment of industry scale has effect in increasing the cost of the transportation aspect. To reduce the cost, CVRP has become the most necessary role in logistic and supply chain management [1], [2]. The basic version of the CVRP was firstly introduced by Dantzig & Ramser [3]. Since then, this problem has been analyzed for finding more optimal technique to provide most optimal solution using both exact optimization methods and heuristic algorithms. The CVRP or also called VRP aims to minimizing the total travel distance needed by a set of vehicles in sending an amount of goods from a central depot to a set of customers. This primary logistics distribution problem is classified by NP-Hard Combinatorial Problem which is the complexity of the problem will increase exponentially by the problem size [4]. According to this complexity, only small sized instances can be solved optimally using exact methods within a reasonable time [5], [6].

One of the VRP solvers is incorporating clustering strategy with other route construction method. This approach partitioning customers into clusters and arranging customer in each cluster to create the route. Cluster-first route-second methods are well-known in solving vehicle routing, this method try to split customers in clusters that compatible to capacity constraint, and solve a traveling salesman problem in each cluster [7]. Gillett & Miller [8] developed good approach with sweep algorithm and successfully solving large scale vehicle-dispatch problem. Fisher & Jaikumar [9] solving a generalized assignment problem to generate. Recently, Fang et. al. [10], developed Voronoi spatial neighborhood-based search heuristic and successfully reduce local search computational time and improve solution quality of large-

scale DCVRPs. Cakir et. al. [11] propose a two-stage, shape-based clustering approach to decompose the problem into smaller subproblem and employ Concorde TSP solver to generate routes for each individual cluster separately. Bouzid & Hacene [12] integrate Lagrangian Split with powerful variable neighborhood search and yield competitive result. The Lagrangian Split used to generate good quality initial solution and improve the local search performance for the VNS. Comert, et.al. [13] shows the effectiveness of presented cluster-first route-second approach in VRP problem of supermarket distribution chain. This kind of approaches also successfully tackle other combinatorial optimization problem such as TSP [14], [15], Bike sharing Rebalancing Problem [16] and Open VRP [17]. These methods are effectively solve TSP variants including VRP variants by taking the advantage of spatial information of node locations.

State-of-the-art research on large scale vehicle routing problems uses metaheuristics [11]. Metaheuristic and the predecessor heuristic approaches are commonly developed as solution methodologies which can produce a good quality solution, even though the most optimal solutions are not guaranteed in reasonable amount of computing time. Toth & Vigo [18] classified the heuristic methods into classical heuristic and meta-heuristic. Classical heuristic approaches have been studied such as saving algorithm [19], sequential and concurrent route building strategy [20], and insertion heuristic [21]. Although the classical heuristic can obtain a feasible solution quickly, this method performs relatively limited exploration of the search space which makes the resulting solution may have a large disparity compared to exact method [22]. In other side, metaheuristic approaches have gained more attention from most researches on the CVRP recently [23]. Toth & Vigo [6] present a granular tabu search strategy for solving vehicle routing problem. Golden et. al. [24] and Li et. al. [25] successfully solved many VRP instances with an approach composed by combination of record-to-record travel and variable-length neighborhood list. Tarantilis et. al. [26] present list based threshold accepting heuristic that require only one parameter to solve CVRP. Mester & Bräysy [27] provide many best-known results with the developed active guided evolutionary strategies (AGES). Reed et. al. [28] demonstrated the use of Ant Colony System in solving VRP with multi compartment. The strength of a metaheuristic algorithm based in the effectiveness of employed intensification and diversification strategies, and the efficiency of the algorithm depend on the decision between global search reinforcement and convergence search in the promising region [29]. Therefore, the focus of metaheuristic strategies studies in VRP lies on intensification and diversification aspect of the algorithm.

Among some metaheuristics approach had been used to solve VRP variants, variable neighborhood search (VNS) has been proved to be one of the most successful one [30]–[35]. However, most of metaheuristic algorithm, including the VNS algorithm, could produce low quality solution regarding to the premature convergence of the search process which lead the algorithm trapped in local optimum condition and unable to jump the solution to another promising search space [33].

One way to enhance the effectiveness of a metaheuristic method is by developing hybrid approach which incorporate combination of two or more meta-heuristic superiorities [36]. Recently, Akpinar [33] developed combination of LNS and ant colony algorithm to solve VRP, the result showed that the combined LNS-ACO outperforms the other variants of traditional LNS approach. in Akpinar [33], Sze had developed hybrid adaptive VNS and LNS approach to tackle VRP. However, this combination of more than one meta-heuristic could make more complex algorithm. The combined algorithm is relatively difficult to develop and need more computation cost during the searching process.

In this study, an enhanced PVNS with ASM is developed for solving VRP. Instead of using complex meta-heuristic approach to improve VNS, this study used a simple perturbation strategy to extend the search horizon and produce more robust solution. This approach adopts PVNS by Polat et. al. [37] and adding ASM by Li et. al. [38] to control the perturbation scheme to effectively escaping the solution out from local optimum condition and continue the search process to another promising region.

2. Method

2.1. Notation

The notations used in this paper are listed in Table 1.

Table 1. Notation used in this study

Notation	Meaning
V	set of customers
V_0	set of customers plus one depot (customer 0): V0 = V ∪ {0}
n	total number of customers: n = |V	
c_{ij}	distance traveled between customer i and j
c_{i0}	distance traveled between customer i and depot
d_i	delivery demands of customer $j, j \ = \ 1,\dots,n$
Q	vehicle maximum capacity
K	vehicle numbers
θ_1	first score value in perturbation selection mechanism
θ_2	second score value in perturbation selection mechanism
w_i	weight of perturbation scheme $i, i \ = \ 1,\dots,10$
λ	a positive parameter in initial construction called route shape parameter
u	a positive parameter to exploit asymmetry of information between depot to customer i and j
v	a positive parameter as assignment priority of customer with larger demands

2.2. Capacitated Vehicle Routing Problem

The CVRP is the basic version of the route planning problem. This problem made up of a single distribution center (depot) and a set of customers with some amount of demands which must be delivered from the depot via a set of K homogeneous fleet. The objective of this problem is to establish the distribution routes which is involving assignment customers including their demands to corresponding vehicles and determining trip sequences to all customers which need minimum total travelled distance without violating maximum capacity constraint of each vehicle. In graph theory, VRP can be defined as $G \ = \ (N, E)$, where N means a set of customers and a single depot ($N \ = \ \{0,\dots,n\}$), and E is the edge set ($E \ = \ \{(i, j) : \ i, j \in N\}$). Node 0 mention the depot as the beginning and ending point of the trips. The rest nodes represent the customers whose known delivery demand d_i. Each customer must be visited in single time by exactly one vehicle. Every vehicle has an exact and identic capacity. The vehicle will travel node by node where the distance from node i to node j is defined by $d_{ij} > 0$. The total delivery demand of the customers belong to a route must less than the maximum vehicle's capacity of Q_k. The mathematical model of VRP has presented in Fisher & Jaikumar [9]. For further information about the VRP variants and some solution approaches, the readers suggested to gather more information in [39]–[42].

2.3. Enhanced Perturbation-based Variable Neighborhood Search with Adaptive Selection Mechanism for VRP

In recent years, VNS received more attention in the VRP literature with some enhancement. perturbation-based variable neighborhood search with adaptive selection mechanism (PVNS ASM) adopt PVNS Framework by Polat et. al. [37] to solve VRP. The structure of the approach is given as pseudo-code in Fig 1. The main framework of this algorithm is VNS that systematically changing the solution over several iterations. Saving algorithm (SA) and variable neighborhood descent (VND) are employed for constructing initial solution and local search respectively. Finally, ASM is used to control selection of perturbation scheme according to empirical record of success for escaping from being trapped in local optimal condition.

1. $N_{mk}, k = 1, \dots, k_{max}$ as *neighborhood structures of shaking phase*
2. $N_m, m = 1, \dots, m_{max}$ as *neighborhood structures of shaking phase*
3. set 1 score to all perturbation scheme
4. construct x^0 generate initial solution with saving heuristic
5. while $p < p_{max}$
6. $s = 1$
7.   while $s < s_{max}$
8. $k = 1$
9.   while k$< k_{max}$
10. $x^2 = x_k^1 \in N_k(x^1)$ Shaking : x_k^1 *is random solution in* k^{th} *neighborhood of* x^1
11.   If $f(x^2) < f(x^1)$
12. $x^1 = x^2, k = 1$
13.   else
14. $x^3 = x^2, m = 1$
15.   while m$< m_{max}$
16. $x^3 = VND(N_m(X^2))$; *local search with VND in* m^{th} *neighbourhood of* x^2
17.   if $f(x^3) < f(x^1)$
18. $x^1 = x^3, k = 1$
19.   else
20. $k = k + 1$
21.   end if
22.   end while
23.   end if
24.   end while
25.   If $f(x^1) < f(x^0)$
26. $x^0 = x^1$; move or not: x^0 is a current best solution
27.   changeperturb = false
28.   Add θ_1 to cth perturbation scheme
29.   else
30. $s = s + 1$
31.   end if
32.   end while
33.   If (changeperturb = true)
34. c = select perturbation scheme by roulade wheel using equation 2
35.   endif
36.   changeperturb = true
37. $x^1 = x_r^0 \in P_c(X_0)$; *Perturbation:* x_r^0 *is a random solution in the* c^{th} *of* x^0
38.   add θ_2 to c^{th} perturbation scheme
39. end while
40. return $x = x^0$

Fig. 1. Structure of PVNS ASM Algorithm

2.3.1. Initial Solution Construction

A heuristic construction method is used to generate initial solution, that is, a well-known saving algorithm by Clarke & Wright [19] with additional consideration to vehicle capacity according to enhanced formula in Altınel & Öncan [43]. This construction heuristic builds a solution considering a saving value. The construction begins with a vehicle dispatched to each customer, then routes are merged repeatedly by the saving value in generated routing cost. In every iteration, two routes are merged into

single route which can result in feasible combination and has largest saving value. This construction stops when feasible combination of routes is no longer available. The saving value [43] is defined in (1).

$$S_{ij} = c_{i0} + c_{0j} - \lambda c_{ij} + u|c_{0i} - c_{j0}| + v\frac{d'_i + d'_j}{\bar{d}} \tag{1}$$

2.3.2. Solution Improvement with VNS

After initial solution is generated by saving algorithm, the solution is evaluated with an improvement algorithm. For this purpose, a VNS approach proposed by Hansen & Mladenovic [44] and [45] is used. This algorithm employs a set of neighborhoods in a deterministic order to explore increasingly distant neighborhood and improve current incumbent solution by investigating the promising region. Reference [37], [45]–[48] revealed the effectiveness of VNS in the implementation for VRP scenarios.

VNS algorithm is consisted of two main phases, namely shaking and local search. The shaking phase responsible to lead the search direction of the algorithm escape from local optimum condition by restructure the solution using curtain neighborhood change. In this paper, a set of neighborhood operators is employed since the probability to escape from local optimum is increasing when the shaking employs more than one operator [37]. 3 intra-route structures are employed which is executed in deterministic order to explore potential another search space after algorithm found local optimum in local search step (Table 2).

Table 2. Intra-route operators employed in shaking step of the algorithm

Operator	Meaning
Exchange (m,n)	This operator transfers m sequential customer from a random route (first route) to another random route (second route) and then n sequential customer transferred from second route to the first route. Where m and n are randomly generated number between 1 to customer number in each route.
Cross	This operator divides two randomly selected routes (first route) become two parts namely part A and part B. Then part A from the first route is connected to part B of the second route while the part B of the second route is connected to part A of second route [49].
Shift (m,0)	This operator transfers m sequential customer from randomly selected route to another randomly selected route at random position where m is randomly generated number between 1 and customer number in the first route.

After algorithm perform shaking phase, the solution is processed to local search phase to find local optimum. In this paper the VND algorithm is used as local search algorithm by applying a set of neighborhood structures N_m, m=1...,m$_{max}$, in a deterministic order. The sequence of movements considered in this work consist of four inter-route structures and three intra-route structures is summarized in Table 3 and Table 4. For further information about the VND algorithm, reader is referred to Avci & Topaloglu [50] and Crispim & Brandão [51].

Table 3. Inter-route structures employed in local search step of the algorithm

Operator	Meaning
Insertion	The best insertion movement that move a client from its position and inserts to another position in the same route.
Swap	This operator finds the best permutation movement that swaps the locations of two client within a route
3-Opt	3-Opt moves some sub route to different position in the same route. For the example, three links (pair of nodes) are removed and replaced by three other links.
2-opt	This operator tries to find the best two-opt movement that swaps pairs of links in the same route.

Table 4. Intra-route structures employed in local search step of the algorithm

Operator	Meaning
Replace (1,1)	This operator finds best exchange 1 customer from selected route with 1 customer from another selected route which gaining less objective value of the solution.
Shift	This Structure finds best transfer of m sequential customer from a selected route to another selected route where m is number between 1 to number of all customer in first route.
Crossover	This structure finds best possible crossover from two selected routes.

These six intra-route neighborhood structures and four inter-route neighborhood structures are employed in the VNS algorithm based on study by which is borrowed from [29], [37], [49], [52].

2.3.3. Perturbation Mechanism

The enhancement of VNS by adding perturbation mechanism in is recently presented by Polat et. al. [37] which has main objective is to extend the search space and diversification degree of the search. In this paper, instead of using perturbation mechanism which is given before, the simple destroy and repair algorithm is employed. This approach adopted from Li et. al. [38], in the study the LNS and an adaptive mechanism was applied to iterated local search (ILS) algorithm as additional perturbation strategy and successfully improved ILS for solving Multi Depot VRPSPD. This destroy and repair method consist of the removal of customer and reinsertion of the removed customer. The removal phase using three removal methods, random removal method introduced by Shaw [53], relatedness removal method by Schrimpf et. al. [54] and long-arc-broken removal [38] which is described in Table 5.

Table 5. Removal method employed in Perturbation phase.

Method	Meaning
Random removal	This method evaluates each customer of the current solution, a customer removed when a generated random number (r) smaller than the threshold level (r_0) where $r, r_0 \in [0,1]$
Relatedness removal	In this method, a customer i is choosen randomly, then select all customers that has distance from customer i between 0 and $r * \text{Average}(i)$, where $\text{Average}(i)$ indicating the average distance between all other customer in the solution to i and r is a random generated number between 0 and 1.
Long arch broken	Firstly, this algorithm computes the sum of longest arc and second longest arc of each route and rank the routes in descending way based on the computation result. Then a random route is selected randomly from first ω routes and remove the customers between the two first longest arcs.

The removed customers denoted by M which is removed from removal step then reinserted using 5 different insertion method which are greedy tournament 1, basic greedy tournament, regret tournament insertion, greedy basic and basic regret-2 insertion. The basic greedy insertion which introduced by Campbell & Savelsbergh [55] move customers from M one by one to the best position in solution while the greedy tournament algorithm select a set M' customer from M namely M' $\subseteq M$ and apply basic greedy insertion to reinsert all M' to solution. This process is repeated until all removed customer is transferred back to the solution. Different with the basic greedy tournament version, greedy tournament 1 chooses only one random customer from removed then applied greedy insertion.

2.3.4. Adaptive selection mechanism

This section describes modification to basic PVNS with additional adaptive capability to select better perturbation scheme of the algorithm. The performance of different neighborhood structures may be significantly different depend on characteristics of every problems and instances [37]. Thus, instead stochastic approach, this study employs an adaptive mechanism to select new perturbation scheme that

will move current solution to the new solution space. This approach based on previous work [38] that used adaptive mechanism to the perturbation step of iterated local search.

According to 10 different perturbation set in Table 6, adaptive mechanism assigns equal weight to each perturbation scheme in the beginning of the search process. During the search process, the weight of the schemes is altered depending on the successful operations of each method. The scoring system of ASM control possibility and the weight of each neighborhood scheme along the search process. When a new overall best solution is founded, the score θ_1 is added to the weight and the scheme is directly selected for the next perturbation call (line 27-28 Fig. 1), Otherwise, the score θ_2 is assigned after applying a perturbation scheme (line 38 Fig. 1).

Table 6. Destroy and repair methods as perturbation scheme

No	Removal Method	Insertion Method
1	Random Removal	Greedy Tournament 1
2	Random Removal	Greedy Tournament 1
3	Relatedness Removal	Greedy Tournament
4	Random Removal	Greedy Tournament
5	Relatedness Removal	Regret Tournament
6	Relatedness Removal	Regret Tournament
7	Relatedness Removal	Greedy Basic
8	Relatedness Removal	Regret 2 basic
9	Long Broken Arc Removal	Greedy Tournament
10	Long Broken Arc Removal	Regret Tournament

In the condition where a perturbation scheme is failed to make the algorithm obtain new best solution, the adaptive mechanism employ roulette wheel selection to find new perturbation set to be applied, can be seen at line 34 of PVNS ASM algorithm in Fig 1. By default, each perturbation scheme is chosen by using roulette wheel method with a probability following its empirical condition. If there are k perturbation set with weight w_i, i ∈ {1,2,...,k}, then the perturbation scheme j is selected using probability formula in (2).

$$Prob = \frac{w_j}{\sum_{k=1}^{k} w_i} \tag{2}$$

2.4. Implementation

The PVNS ASM method given in the section 2 has been developed using Visual Studio Code. All computational experiments were performed on a virtual machine by Google cloud Platform with Intel Xeon E5 Skylake 2.0 GHz processor with 3.75 GB RAM and Debian 9.0 with Kernel Linux 4.14.8-1 LTS. The test problems, parameter settings and computational results for each test problem are presented in the following subsections.

In the following section, the effectiveness of the perturbation technique towards VNS algorithm is analyzed to obtain better solution, a comparative test against basic VNS algorithm is performed. In addition, some experiments to evaluate the performance of our algorithm when compared to existing methods from the literature also given. All of two comparative tests performed on two sets of benchmark data sets taken from literatures. These include 21 classical Euclidean VRP namely set A and 16 VRP instance namely set B described in Augerat et. al. [56]. The BKS value taken from Akpinar [33] and the distance of arcs for all the instances have been calculated as rounded Euclidean distances. The first dataset is made up by 32 to 80 customers while the second dataset are made up by 22 to 78 customers. All the benchmark instances were derived from the site <http://neo.lcc.uma.es>.

2.5. Parameter Settings

The robustness of parameter settings influences the performance of algorithm on different data sets. Thus, several tests were performed on two main process of the algorithm involving VNS algorithm and adaptive perturbation mechanism. In order to test neighborhood sequence in shaking and local search step, various combination of neighborhood structures has been used. From the test, the result shows the following sequence is the best configuration to be employed to the shaking step: N1: exchange(m,n), N2: cross, N3: shift(m,0), for the local search step: N1: insertion, N2: swap, N3: 3-opt, N4: 2-opt, N5: replace, N6: shift and N7: cross. Therefore, k_{max} is set to 3 and m_{max} is 7.

Two major parameters of the PVNS ASM is perturbation counter and VNS termination counter. In The perturbation mechanism is performed after p_{max} loops and the VNS algorithm is terminated after s_{max} iterations counted from the last accepted solution. In order to obtain optimum setting of these parameters, a test was conducted using a benchmark instance involving 20 customers. As the result the best parameters setting for p_{max} is 50 and s_{max} is 100 where N is customer number in the instance. The last, the parameter of scoring system θ_1 is set to 5 and θ_2 is 10.

3. Results and Discussion

The following subsections presents the computational results of the proposed algorithm for the VRP.

3.1. Performance evaluation of the perturbation strategy toward VNS algorithm

In this part, first computational experiment performed to analyze the effect of the perturbation strategy in search result over basic VNS algorithm. As stated above, the test performed over two set CVRP instances composed by 21 VRP instances. The best result of each instances presented in Table 7 and Table 8 for each method have been acquired by running the algorithm 10 times independently. The results are presented in Table 7 and Table 8 for the problem sets A and B and analyzed by considering the gap between BKS values and obtained results as relative deviations %Dev = ((Best–BKS)/BKS)*100.

Table 7. Computational result against basic VNS approach for dataset A

Instance	BKS	VNS			PVNS ASM		
		Best	CPU Time (s)	%Dev	Best	CPU Time (s)	%Dev
A–n32-k5	787	787	45.23	**0.0000**	787	30.67	**0.0000**
A–n33-k5	661	661	48.1	**0.0000**	661	33.49	**0.0000**
A–n33-k6	742	742	55.61	**0.0000**	742	32.39	**0.0000**
A–n34-k5	778	778	88.72	**0.0000**	778	61.17	**0.0000**
A–n36-k5	799	807	134.591	1.0013	799	90.23	**0.0000**
A–n37-k5	669	669	201.31	**0.0000**	669	59	**0.0000**
A–n37-k6	949	961	71.7	1.2645	951	23	**0.2107**
A–n38-k5	730	730	168.7	**0.0000**	730	50.04	**0.0000**
A–n39-k5	822	822	203.7	**0.0000**	822	136.29	**0.0000**
A–n39-k6	833	835	122.39	0.2401	833	98.64	**0.0000**
A–n44-k6	937	951	179.525	1.4941	942	130	**0.5336**
A–n45-k6	944	959	131.83	1.5890	948	95.74	**0.4237**
A–n45-k7	1146	1172	109.222	2.2688	1146	81	**0.0000**
A–n46-k7	914	917	125.15	0.3282	914	91.76	**0.0000**
A–n48-k7	1073	1084	133.609	1.0252	1073	47	**0.0000**
A–n55-k	1074	1385	187.172	28.9572	1074	101	**0.0000**
A–n60-k9	1355	1360	98.23	**0.3690**	1360	74	**0.3690**
A–n61-k9	1039	1050	173.199	1.0587	1042	65.7	**0.2887**
A–n63-k9	1622	1637	224.52	0.9248	1629	95.16	**0.4316**
A–n65-k9	1174	1232	215.302	4.9404	1180	84.4	**0.5111**
A–n80-k10	1763	1794	490.2707	1.7584	1784	186.2	**1.1912**
Average			152.766	2.249		**79.375**	**0.189**

Table 8. Computational result against basic VNS approach for instance B

Instances	BKS	VNS			PVNS_ASM		
		Best	**CPU.Time (s)**	**%Dev**	**Best**	**CPU.Time (s)**	**%Dev**
B-n31-k5	672	682	62.717	1.49	672	28.3	**0.00**
B-n34-k5	788	788	103.632	**0.00**	788	45.8	**0.00**
B-n38-k6	805	805	102.54	**0.00**	805	40.866	**0.00**
B-n39-k5	549	558	201.512	1.64	549	81.48	**0.00**
B-n41-k6	829	836	109.602	0.84	830	45.82	**0.12**
B-n43-k6	742	742	140.38	**0.00**	742	57.3	**0.00**
B-n44-k7	909	909	104.64	**0.00**	909	41.2	**0.00**
B-n45-k5	751	751	380.8	**0.00**	751	130	**0.00**
B-n45-k6	678	715	150.05	5.46	680	62.61	**0.29**
B-n50-k7	741	741	223.85	**0.00**	741	80.47	**0.00**
B-n50-k8	1312	1328	121.074	1.22	1313	51	**0.08**
B-n56-k7	707	714	367	0.99	707	126	**0.00**
B-n66-k9	1316	1323	277.51	0.53	1320	104.5	**0.30**
B-n67-k10	1032	1074	237.62	4.07	1042	92.15	**0.97**
B-n68-k9	1272	1252	364.94	**-1.57**	1287	149.8	1.18
B-n78-k10	1221	1249	438.25	2.29	1242	196.03	**1.72**
Average			211.63	1.06		83.33	0.29

Comparison of percentage deviations of best result from the PVNS ASM and basic VNS approach. For better understanding, the computation result in term percentage deviation presented in Fig. 2 and computation time is presented in Fig. 3. From the observations of Fig. 2, we can conclude that the hybrid PVNS ASM algorithm outperforms the basic VNS algorithms in terms of solution quality. The PVNS ASM has obtain better solution almost for all instances except B15 (B-n68-k9). However, PVNS ASM has higher success rate on overall test performed with 22 instance achieved best-known solution and the other 15 instances obtain relatively close objective with tiny %Dev value 0.23 on average from all A and B instances, otherwise the basic VNS approach seems obtained worse solution with 13 instances achieved best-known solution, 23 worse than best-known solution and 1 instance with new best solution achieved from all A and B.
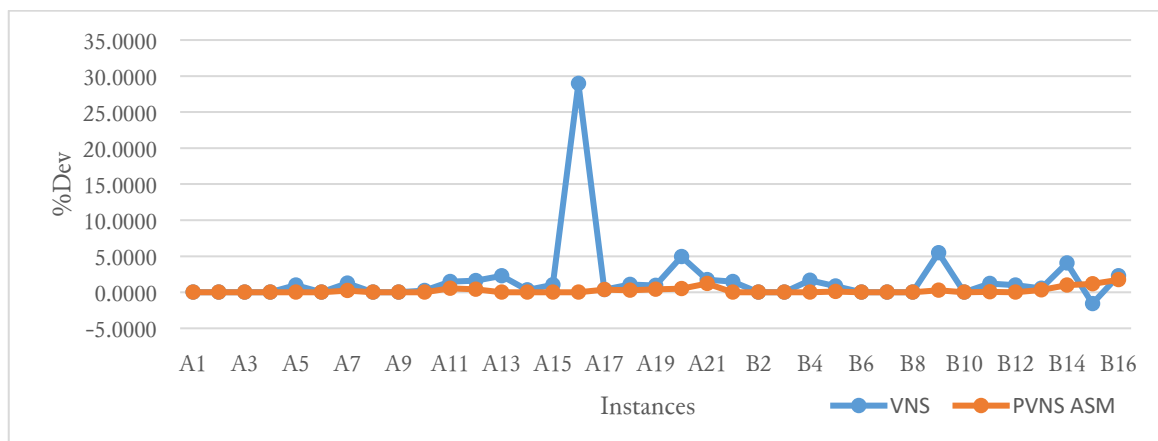

Fig. 2. Comparison of percentage deviation of objective gap by the PVNS ASM and basic VNS approach for problem set A and B

In addition, Fig. 3 shows that the PVNS ASM approach needs less computation time to find optimum solution than the basic VNS algorithm for almost all instances. Finally, the Fig. 2 and Fig. 3 proofed the effectiveness of the perturbation strategy in escaping the algorithm from local optimum.
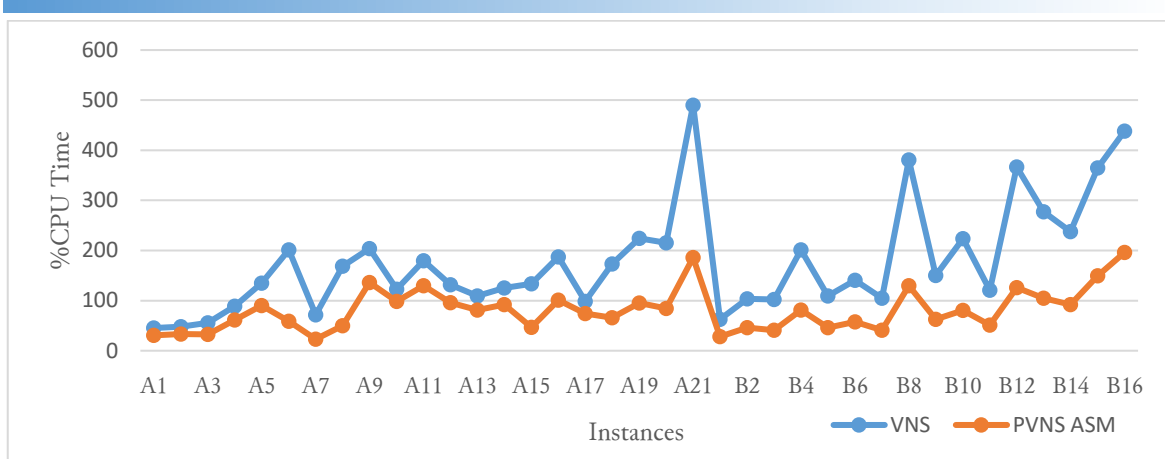
Fig. 3. Comparison of computation time needed by the PVNS ASM and basic VNS approach for problem set A and B

3.2. Comparison of PVNS ASM against some other approach

This section discusses the effectiveness of the PVNS ASM method over 4 earlier developed CVRP approaches, LNS-ACO [33]. KNN [57], Hybrid VNS-VND [58], and SC-SA [59]. For this purpose, we perform the solution method to benchmark set (9 A & 12 B instances) used by Augerat et. al. [56]. The best-known solutions (BKS) for all benchmark set were derived from [15]. The results then presented in Table 9 and Table 10, taken from 10-time executions. The best results achieved by the algorithms are marked in boldfaces which has same or better quality to best-known solution. The comparative results given in Table 9 and Table 10 showed the effectiveness of the PVNS ASM against other meta-heuristic since it could provide best result from 15 out of 21 problems set among other method, while the other 6 instance close-like solution also obtained successfully with small deviations between ranges 0.12 to 1.72.

Table 9. Computational result avgainst CVRP methods for problem set A

Instances	No. Cust	No. Vehicle	BKS	KNN %Dev	VNS-VND %Dev	LNS_ACO %Dev	PVNS ASM %Dev
A-n32-k5	32	5	787	**0.00**	1.27	**0.00**	**0.00**
A-n33-k5	33	5	661	**0.00**	21.79	0.17	**0.00**
A-n33-k6	33	6	742	**0.00**	17.12	0.09	**0.00**
A-n37-k5	37	5	669	**0.00**	31.99	0.52	**0.00**
A-n45-k6	45	6	944	1.48	19.60	**0.09**	0.42
A-n45-k7	45	7	1146	**0.00**	0.00	0.06	**0.00**
A-n60-k9	60	9	1355	**0.00**	5.98	0.06	0.37
A-n65-k9	65	9	1174	0.94	25.30	0.66	**0.51**
A-n80-k10	80	10	1763	2.95	**0.20**	**0.20**	1.19
Average				0.60	13.15	0.21	**0.19**

The approach given in this study has relatively simple algorithm compared to hybrid approach, but the result proofed the effectiveness of this approach in solving VRP. These advantages make the algorithm easy to understand and implemented by other developer in case of real-world using or any development purpose. Although the proposed hybrid algorithm is tested only on standard CVRP cases, it can be extended to solve many other VRP variants with just little modification. More over the implementation of meta-heuristic approach in solving VRP has many direct benefits to transportations and logistics in planning the distribution system effectively while reducing the transportation cost significantly. In addition, even though the algorithm developed in this study is not configured to work in parallel computation, it has many potentials to build the more powerful algorithm with parallel configuration. Thus, the robustness and computation speed will be increased significantly since the

growing of current CPU technologies focus on making processor work with multi CPU and multi-threaded which support penalization of computation.

Table 10. Computational result against CVRP methods for problem set B

Instances	No. Cust	No. Vehicle	BKS	LNS_ACO %Dev	SC-ESA %Dev	PVNS ASM %Dev
B-n38-k6	38	6	805	**0.00**	1.24	**0.00**
B-n39-k5	39	5	549	**0.00**	0.18	**0.00**
B-n41-k6	41	6	829	**0.00**	4.46	0.12
B-n43-k6	43	6	742	**0.00**	0.54	**0.00**
B-n44-k7	44	7	909	**0.00**	1.32	**0.00**
B-n45-k5	45	5	751	**0.00**	0.00	**0.00**
B-n45-k6	45	6	678	**0.00**	1.18	0.29
B-n50-k8	50	8	1312	0.53	1.30	**0.08**
B-n66-k9	66	9	1316	1.06	1.90	**0.30**
B-n67-k10	67	10	1032	1.74	1.74	**0.97**
B-n68-k9	68	9	1272	1.42	1.57	**1.18**
B-n78-k10	78	10	1221	**0.6**	2.05	1.72
Average				0.4425	1.456666667	**0.39**

4. Conclusion

In this study, PVNS ASM algorithm is developed to solve CVRP. This method is composed through combination of savings heuristic, VNS, perturbation mechanism and the ASM to tackle the VRP. Three neighborhood structures employed as shaking operators and seven neighborhood structures as local search operators. A VND procedure is used to incorporate several neighborhood structures for local search. Ten different destroy and repair structures were presented to jump the solution from being trapped in local optimal condition. The testing results proof that the solution method efficiently reaches high-quality CVRP solutions compared to the basic VNS approach for the given benchmark instances. In addition, this the solution method has a competitive result compared to currently best-known solution in literatures. This result indicates that the combination of improvement strategy by VNS and the diversification strategy used in this paper is effective to jump from local optimum condition. Further research can be done to develop this strategy to other CVRP variants. Moreover, it is worth to try parallelization strategies by utilizing different threads of CPUs for PVNS ASM that can improve the performance of the algorithm.

References

[1]   F. Alfredo Tang Montané, R. D. Galvão, A. T. Montané, and R. D. Galvão, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service," *Comput. Oper. Res.*, vol. 33, no. 3, pp. 595–619, 2006, doi: https://doi.org/10.1016/j.cor.2004.07.009.

[2]   J. Dethloff, "Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up," *OR Spektrum*, vol. 23, no. 1, pp. 79–96, 2001, doi: https://doi.org/10.1007/PL00013346.

[3]   G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, Oct. 1959, doi: https://doi.org/10.1287/mnsc.6.1.80.

[4]   G. Nagy, N. A. Wassan, M. G. Speranza, and C. Archetti, "The vehicle routing problem with divisible deliveries and pickups," *Transp. Sci.*, vol. 49, no. 2, pp. 271–294, 2015, doi: https://doi.org/10.1287/trsc.2013.0501.

[5]   R. Baldacci, P. Toth, and D. Vigo, "Exact algorithms for routing problems under vehicle capacity constraints," *Ann. Oper. Res.*, vol. 175, no. 1, pp. 213–245, Mar. 2010, doi: https://doi.org/10.1007/s10479-009-0650-0.

[6]   P. Toth and D. Vigo, "The Granular Tabu Search and Its Application to the Vehicle-Routing Problem," *INFORMS J. Comput.*, vol. 15, no. 4, pp. 333–346, Nov. 2003, doi: https://doi.org/10.1287/ijoc.15.4.333.24890.

[7] C. Prodhon and C. Prins, "A survey of recent research on location-routing problems," *Eur. J. Oper. Res.*, vol. 238, no. 1, pp. 1–17, 2014, doi: https://doi.org/10.1016/j.ejor.2014.01.005.

[8] B. E. Gillett and L. R. Miller, "A Heuristic Algorithm for the Vehicle-Dispatch Problem," *Oper. Res.*, vol. 22, no. 2, pp. 340–349, Apr. 1974, doi: https://doi.org/10.1287/opre.22.2.340.

[9] M. L. Fisher and R. Jaikumar, "A generalized assignment heuristic for vehicle routing," *Networks*, vol. 11, no. 2, pp. 109–124, 1981, doi: https://doi.org/10.1002/net.3230110205.

[10] Z. Fang, W. Tu, Q. Li, S.L. Saw, S. Chen, and B.Y. Chen, "A Voronoi neighborhood-based search heuristic for distance / capacity constrained very large vehicle routing problems," *Int. J. Geographic. Inf. Sci.*, vol. 27, no. 4, 2013, pp. 37–41, 2012, doi: https://doi.org/10.1080/13658816.2012.707319.

[11] F. Cakir, W. N. Street, and B. W. Thomas, "Revisiting Cluster First , Route Second for the Vehicle Routing Problem," no. August, 2015, available at: https://pdfs.semanticscholar.org/2891/fc38fe7b52142701c63bbc85f0276501d6f3.pdf.

[12] M. C. Bouzid, A.H. Hacene, and S. Salhi, "An Integration of Lagrangian Split and VNS : The case of the Capacitated Vehicle Routing Problem," *Comput. Oper. Res.*, vol. 78, pp. 513-525, 2017, doi: https://doi.org/10.1016/j.cor.2016.02.009.

[13] S. E. Comert, H. R. Yazgan, S. Kır, and F. Yener, "A cluster first-route second approach for a capacitated vehicle routing problem: a case study," *Int. J. Procure. Manag.*, vol. 11, no. 4, p. 399, 2018, doi: https://doi.org/10.1504/IJPM.2018.092766.

[14] A. A. Khan, M. U. Khan, and M. Iqbal, "Multilevel Graph Partitioning Scheme to Solve Traveling Salesman Problem," in *2012 Ninth International Conference on Information Technology - New Generations*, Las Vegas, NV, 2012, pp. 458-463. doi: https://doi.org/10.1109/ITNG.2012.106.

[15] C. Walshaw, "A Multilevel Approach to the Travelling Salesman Problem," *Oper. Res.*, vol. 50, no. 5, pp. 751-922, 2002, doi: https://doi.org/10.1287/opre.50.5.862.373.

[16] C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl, "A Cluster-First Route-Second Approach for Balancing Bicycle Sharing Systems," 2015, pp. 439–446, doi: https://doi.org/10.1007/978-3-319-27340-2_55.

[17] D. Sariklis and S. Powell, "A heuristic method for the open vehicle routing problem," *J. Oper. Res. Soc.*, vol. 51, no. 5, pp. 564–573, May 2000, doi: https://doi.org/10.1057/palgrave.jors.2600924.

[18] P. Toth and D. Vigo, *Vehicle routing: problems, methods, and applications*. Philadelphia, PA, USA, PA: Society for Industrial and Applied Mathematics, 2014, available at: https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.fm.

[19] G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Oper. Res.*, vol. 12, no. 4, pp. 519-643, 1964, doi: https://doi.org/10.1287/opre.12.4.568.

[20] R. H. Mole and S. R. Jameson, "A Sequential Route-Building Algorithm Employing a Generalised Savings Criterion," *Oper. Res. Q.*, vol. 27, no. 2, p. 503, 1976, doi: https://doi.org/10.2307/3008819.

[21] N. Christofides and S. Eilon, "An Algorithm for the Vehicle- dispatching Problem," *Oper. Res.*, vol. 20, no. 3, pp. 309–318, 1969, doi: https://doi.org/10.1057/jors.1969.75.

[22] S. W. Lin, Z. J. Lee, K. C. Ying, and C. Y. Lee, "Applying hybrid meta-heuristics for capacitated vehicle routing problem," *Expert Syst. Appl.*, vol. 36, no. 2 PART 1, pp. 1505–1512, 2009, doi: https://doi.org/10.1016/j.eswa.2007.11.060.

[23] V. F. Yu, A. A. N. P. Redi, C. L. Yang, E. Ruskartina, and B. Santosa, "Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem," *Appl. Soft Comput. J.*, vol. 52, pp. 657–672, 2017, doi: https://doi.org/10.1016/j.asoc.2016.10.006.

[24] B. L. Golden, E. A. Wasil, J. P. Kelly, and I.-M. Chao, "The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Sets, and Computational Results," in *Fleet Management and Logistics*, Boston, MA: Springer US, 1998, pp. 33–56, doi: https://doi.org/10.1007/978-1-4615-5755-5_2.

[25] F. Li, B. Golden, and E. Wasil, "Very large-scale vehicle routing: new test problems, algorithms, and results," *Comput. Oper. Res.*, vol. 32, no. 5, pp. 1165–1179, May 2005, doi: https://doi.org/10.1016/j.cor.2003.10.002.

[26] C. D. Tarantilis, C. T. Kiranoudis, and V. S. Vassiliadis, "A list based threshold accepting algorithm for the capacitated vehicle routing problem," *Int. J. Comput. Math.*, vol. 79, no. 5, pp. 537–553, 2002, doi: https://doi.org/10.1080/00207160210948.

[27] D. Mester and O. Bräysy, "Active guided evolution strategies for large-scale vehicle routing problems with time windows," *Comput. Oper. Res.*, vol. 32, no. 6, pp. 1593–1614, Jun. 2005, doi: https://doi.org/10.1016/j.cor.2003.11.017.

[28] M. Reed, A. Yiannakou, and R. Evering, "An ant colony algorithm for the multi-compartment vehicle routing problem," *Appl. Soft Comput. J.*, vol. 15, pp. 169–176, 2014, doi: https://doi.org/10.1016/j.asoc.2013.10.017.

[29] C. B. C. B. Kalayci and C. Kaya, "An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery," *Expert Syst. Appl.*, vol. 66, pp. 1–18, 2016, doi: https://doi.org/10.1016/j.eswa.2016.09.017.

[30] M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann, "A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows," *J. Heuristics*, vol. 10, no. 6, pp. 613–627, Dec. 2004, doi: https://doi.org/10.1007/s10732-005-5432-5.

[31] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau, "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems," *Comput. Oper. Res.*, vol. 34, no. 9, pp. 2743–2757, Sep. 2007, doi: https://doi.org/10.1016/j.cor.2005.10.010.

[32] K. Fleszar, I. H. Osman, and K. S. Hindi, "A variable neighbourhood search algorithm for the open vehicle routing problem," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 803–809, 2009, doi: https://doi.org/10.1016/j.ejor.2007.06.064.

[33] S. Akpinar, "Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem," *Expert Syst. Appl.*, vol. 61, pp. 28–38, 2016, doi: https://doi.org/10.1016/j.eswa.2016.05.023.

[34] J. F. Sze, S. Salhi, and N. Wassan, "A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem," *Expert Syst. Appl.*, vol. 65, pp. 383–397, 2016, doi: https://doi.org/10.1016/j.eswa.2016.08.060.

[35] N. Mladenović and P.Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, 1997, doi: https://doi.org/10.1016/S0305-0548(97)00031-2.

[36] C. Blum, "Hybrid metaheuristics in combinatorial optimization: A tutorial," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7505 LNCS, no. 6, pp. 1–10, 2012, doi: https://doi.org/10.1007/978-3-642-33860-1_1.

[37] O. Polat, C. B. Kalayci, O. Kulak, and H.-O. Günther, "A perturbation based variable neighborhood search heuristic for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery with Time Limit," *Eur. J. Oper. Res.*, vol. 242, no. 2, pp. 369–382, 2015, doi: https://doi.org/10.1016/j.ejor.2014.10.010.

[38] J. Li, P. M. Pardalos, H. Sun, J. Pei, and Y. Zhang, "Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3551–3561, 2015, doi: https://doi.org/10.1016/j.eswa.2014.12.004.

[39] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1472–1483, Nov. 2009, doi: https://doi.org/10.1016/j.cie.2009.05.009.

[40] G. Laporte, R. Musmanno, and F. Vocaturo, "An Adaptive Large Neighbourhood Search Heuristic for the Capacitated Arc-Routing Problem with Stochastic Demands," *Transp. Sci.*, vol. 44, no. 1, pp. 125–135, Feb. 2010, doi: https://doi.org/10.1287/trsc.1090.0290.

[41] P. Toth and D. Vigo, *Vehicle Routing Problem*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014, doi: https://doi.org/10.1137/1.9781611973594.

[42] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Comput. Ind. Eng.*, vol. 99, pp. 300–313, Sep. 2016, doi: https://doi.org/10.1016/j.cie.2015.12.007.

[43] İ. K. Altınel and T. Öncan, "A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem," *J. Oper. Res. Soc.*, vol. 56, no. 8, pp. 954–961, 2005, doi: https://doi.org/10.1057/palgrave.jors.2601916.

[44] P. Hansen and N. Mladenovic, "Variable neighborhood search: Principles and applications," *Eur. J. Oper. Res.*, vol. 130, no. 3, pp. 449–467, 2001, doi: https://doi.org/10.1016/S0377-2217(00)00100-4.

[45] P. Hansen, N. Mladenović, and J. A. Moreno Pérez, "Variable neighbourhood search: Methods and applications," *Ann. Oper. Res.*, vol. 175, no. 1, pp. 367–407, 2010, doi: https://doi.org/10.1007/s10479-009-0657-6.

[46] T. Doyuran and B. Çatay, "A robust enhancement to the Clarke−Wright savings algorithm," *J. Oper. Res. Soc.*, vol. 62, no. 1, pp. 223–231, 2011, doi: https://doi.org/10.1057/jors.2009.176.

[47] V. C. Hemmelmayr, K. F. Doerner, and R. F. Hartl, "A variable neighborhood search heuristic for periodic routing problems," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 791–802, 2009, doi: https://doi.org/10.1016/j.ejor.2007.08.048.

[48] F. Pan, C. Ye, K. Wang, and J. Cao, "Research on the Vehicle Routing Problem with Time Windows Using Firefly Algorithm," *J. Comput.*, vol. 8, no. 9, pp. 2256–2261, 2013, doi: https://doi.org/10.4304/jcp.8.9.2256-2261.

[49] F. P. Goksal, I. Karaoglan, and F. Altiparmak, "A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery," *Comput. Ind. Eng.*, vol. 65, no. 1, pp. 39–53, 2013, doi: https://doi.org/10.1016/j.cie.2012.01.005.

[50] M. Avci and S. Topaloglu, "An adaptive local search algorithm for vehicle routing problem with simultaneous and mixed pickups and deliveries," *Comput. Ind. Eng.*, vol. 83, pp. 15–29, 2015, doi: https://doi.org/10.1016/j.cie.2015.02.002.

[51] J. Crispim and J. Brandão, "Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls," *J. Oper. Res. Soc.*, vol. 56, no. 11, pp. 1296–1302, 2005, doi: https://doi.org/10.1057/palgrave.jors.2601935.

[52] Y. Gajpal and P. Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup," *Comput. Oper. Res.*, vol. 36, no. 12, pp. 3215–3223, 2009, doi: https://doi.org/10.1016/j.cor.2009.02.017.

[53] P. Shaw, "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems," in *Principles and Practice of Constraint Programming --- CP98: 4th International Conference, CP98 Pisa, Italy, October 26--30, 1998 Proceedings*, M. Maher and J.-F. Puget, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 417–431, doi: https://doi.org/10.1007/3-540-49481-2_30.

[54] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck, "Record Breaking Optimization Results Using the Ruin and Recreate Principle," *J. Comput. Phys.*, vol. 159, no. 2, pp. 139–171, Apr. 2000, doi: https://doi.org/10.1006/jcph.1999.6413.

[55] A. M. Campbell and M. Savelsbergh, "Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems," *Transp. Sci.*, vol. 38, no. 3, pp. 369–378, 2004, doi: https://doi.org/10.1287/trsc.1030.0046.

[56] P. Augerat, J. Belenguer, E. Benavent, A. Coberan, D. Naddef, and G. Rinaldi, "Computational results with a branch-and-cut code for the capacitated vehicle routing problem," *Rapp. Rech. - IMAG*, vol. 95, no. 949–M, p. 30, 1998, available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.718.7812.

[57] M. A. Mohammed *et al.*, "Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution," *J. Comput. Sci.*, vol. 21, pp. 232–240, 2017, doi: https://doi.org/10.1016/j.jocs.2017.04.012.

[58] M. Amous, S. Toumi, B. Jarboui, and M. Eddaly, "A variable neighborhood search algorithm for the capacitated vehicle routing problem," *Electron. Notes Discret. Math.*, vol. 58, pp. 231–238, 2017, doi: https://doi.org/10.1016/j.endm.2017.03.030.

[59] M. Stanojević, B. Stanojević, and M. Vujošević, "Enhanced savings calculation and its applications for solving capacitated vehicle routing problem," *Appl. Math. Comput.*, vol. 219, no. 20, pp. 10302–10312, 2013, doi: https://doi.org/10.1016/j.amc.2013.04.002.