

# A Modified Meta-Heuristic Approach for Vehicle Routing Problem with Simultaneous Pickup and Delivery

Alfian Faiz<sup>1</sup>, Subiyanto Subiyanto<sup>2</sup>, Ulfah Mediaty Arief<sup>3</sup>

*Electrical Departement, Engineering Faculty, Universitas Negeri Semarang, Indonesia*  
a.faiz@students.unnes.ac.id<sup>1</sup>, subiyanto@mail.unnes.ac.id<sup>2</sup>, ulfahmediatyrief@mail.unnes.ac.id<sup>3</sup>

## ARTICLE INFO

### Article history:

Received: 2018-07-02

Revised: 2018-08-09

Accepted: 2018-08-29

### Keywords:

Adaptive mechanism

Meta-heuristics

Perturbation mechanism

Variable neighborhood search

Vehicle routing problem with simultaneous pickups and deliveries

## ABSTRACT

The aim of this work is to develop an intelligent optimization software based on enhanced VNS meta-heuristic to tackle Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD). An optimization system developed based on enhanced Variable Neighborhood Search with Perturbation Mechanism and Adaptive Selection Mechanism as the simple but effective optimization approach presented in this work. The solution method composed by combining 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 each perturbation scheme success along the search. The ASM help 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 a well-known exact approach is presented to test the solution method in a generated VRPSPD benchmark instance in limited computation time. Then a test to VRPSPD scenario provided by a liquefied petroleum gas distribution company is performed. The test result confirms that solution method present superior performance against exact approach solution in giving best solution for larger sized instance and successfully obtain substantial improvements when compared to the basic VNS and original route planning technique used by a distributor company.

Copyright © 2017 International Journal of Artificial Intelligence Research.

All rights reserved.

## I. Introduction

The growing of industry sector has caused various environmental problems and drawing global attention in waste management around the world. Newly regulations force companies to take responsibility for their product's lifecycle, reducing waste production and increasing the effort in environment protection [1]–[3]. In order to reduce waste production and environmental effect, companies choose to recycle, reuse and remanufacture their product after usage. Thus, companies could get the revenue of their product for next production and protect the environment at once. Consequently, this effort to reclaim waste production from customer requires companies to make decisions for distribution plan carefully which involving the distribution of new product to customer from depot and take the packaging back to the depot. In many practical application customers also need both delivery and pick-up demand to be serviced simultaneously [3], [4]. The VRPSPD encompass such distribution scheme where customers have to be serviced simultaneously for the pickup and delivery demands.

The VRPSPD is a variant of Vehicle Routing Problem (VRP) which is a problem rising in a distribution process. The basic VRP involving the delivery process of goods to a set of customers using a set of homogenous capacitated vehicles, while the VRPSPD not only delivery process but also picking up goods from customer simultaneously. VRPSPD has many potential benefits in decreasing transportation cost from bi-directional distribution scheme since the delivery and pickup process performed simultaneously in same visit. However, this problem has a very high complexity to be

solved. From a mathematical point of view, this problem is classified by NP-Hard Combinatorial Problem which is the complexity of the problem will increase exponentially by the problem size [5]. For this reason, the usage of high-performance computation with intelligent optimization solver for solving VRPSPD effectively is needed.

Many approaches have been applied for VRPSPD, some study analyze exact methods to find an exact solution for VRPSPD such as [5], [6] with branch-and-price approach, branch-and-cut approach [7], [8]. Although exact methodologies are very useful to prove optimal solutions for well-known data instances, exact solution procedures may not be adequately efficient within a reasonable time since the solution space grows exponentially. Therefore, heuristic and metaheuristic approaches are commonly developed as solution methodologies which can produce a high-quality solution for medium and large-scale problems within relatively short computational times [9], [10]. In recent studies, algorithms that iteratively employ intensive local search strategies such as large neighborhood search, variable neighborhood search (VNS) and iterative local search solution approaches have gained attention by many studies for solving VRPSPD with better solution quality vs. time performance [11].

VNS has been proved to be one of the most successful meta-heuristic algorithms in solving various VRP variants [12]–[17]. This method incorporates a systematic intensification strategy while providing a set diversification strategy. However, this approach also has weakness as other meta-heuristic methods. In the search process VNS could be trapped in local optimum and lead to premature convergence that decreases the quality of produced solution. The aim of this work is to develop an intelligent optimization software based on enhanced VNS meta-heuristic based algorithm to solve the VRPSPD. An additional perturbation strategy is presented for escaping VNS from being trapped in local optimum condition. The additional perturbation strategy employs several destroy and repair schemes and adaptive selection mechanism to control diversification degree of the solution. The adaptive selection mechanism empirically selecting the destroy and repair scheme according to the success rate of each scheme.

The structure of this paper is arranged as follows. In the second section, brief explanation about VRPSPD is explained. The notations used are presented in section 2. The solution method for solving the VRPSPD and its components are presented in Section 3. The developed solution method discussed in section 4. The Section 5 shows several computational results for the solution method. Finally, the main conclusions of this research are given in Section 6.

## II. Notation

Below are the notations used in this work:

Notation:

$V$  set of customers to be visited

$V_0$  set of customers with depot (*customer 0*) included:  $V_0 = V \cup \{0\}$

$n$  number of customers.  $n = |V|$

$c_{ij}$  distance between customer  $i$  and  $j$

$p_i$  number of pickup demand of customer  $i$ , where  $i = 1, \dots, n$

$d_i$  the delivery demand of customer  $i$ , where  $i = 1, \dots, n$

$Q$  the capacity of each vehicle

$K$  number of available vehicles

$y_{ij}$  pickup goods transported on  $arc(i, j)$

$z_{ij}$  delivery goods transported on  $arc(i, j)$

$x_{ij}^k$  a decision variable is set to 1 if the vehicle  $k$  visit customer  $i$  before the customer  $j$ , 0 otherwise

$s_{ij}$  saving value between customer  $i$  and  $j$

- $c_{i0}$  distance between customer  $i$  and depot  
 $c_{0i}$  distance between customer depot and  $i$   
 $\lambda$  a positive parameter used to redesign routes  
 $c_{ij}$  distance between customer  $i$  and  $j$   
 $u$  a positive parameter exploits information between customer  $i$  and  $j$  regarding to their distance to depot  
 $v$  a positive parameter used to create priority for larger customer with demands  
 $d'_i$  maximum delivery and pickup demand of customer  $i$   
 $s_{ij}$  saving value between customer  $i$  and  $j$   
 $c_{i0}$  distance between customer  $i$  and depot  
 $c_{0i}$  distance between customer depot and  $i$   
 $\lambda$  a positive parameter used to redesign routes  
 $c_{ij}$  distance between customer  $i$  and  $j$   
 $prob_j$  probability of perturbation scheme  $j$   
 $w_j$  weight of perturbation scheme  $j$   
 $K$  number of perturbation scheme used

### III. VRPSPD

The problem discussed in this study deals with constructing the trip sequences of a set of vehicles in servicing pickup and delivery demands of customers simultaneously. A set of customers is located in distributed places around a main depot. The pickup and delivery demand of each client must be serviced exactly once by a given capacitated vehicles. All vehicles start from depot or distribution center carrying total amount of goods that must be delivered to customers, and picking goods from each customer along the way return to depot as collection center. See [10] for further information about the VRPSPD as well recent survey in methods used to solve this problem and its variants. The following is VRPSPD formulation as mixed-integer linear programming (MILP) taken from Alfredo Tang Montané et al., [18]:

Objective

$$\min \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_0} c_{ij} x_{ij}^k \quad (1)$$

Subject to

$$\sum_{i \in V} \sum_{k \in K} x_{ij}^k = 1 \quad \forall j \in V_0 \quad (2)$$

$$\sum_{i \in V_0} x_{ij}^k - \sum_{i \in V_0} x_{ji}^k = 0 \quad \forall j \in V_0, k \in K \quad (3)$$

$$\sum_{j \in V} x_{0j}^k \leq 1 \quad \forall k \in K \quad (4)$$

$$\sum_{i \in V} x_{i0}^k \leq 1 \quad \forall k \in K \quad (5)$$

$$\sum_{i \in V_0} y_{ji} - \sum_{i \in V_0} y_{ij} = p_j \quad \forall j \in V \quad (6)$$

$$\sum_{i \in V_0} z_{ij} - \sum_{i \in V_0} z_{ji} = d_j \quad \forall j \in V \quad (7)$$

$$y_{ij} + z_{ij} \leq Q \sum_{k \in K} x_{ij}^k \quad \forall i, j \in V \quad (8)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i, j \in V, k \in K \quad (9)$$

$$y_{ij} \geq 0, z_{ij} \geq 0, \pi_i \geq 0 \quad \forall i, j \in V \quad (10)$$

#### IV. Solution Method

In recent years, VNS received more attention in the VRP literature with some enhancement. In this study, we adopt Perturbation-based Variable Neighborhood Search (PVNS) Framework by Polat [16] and adding control to the perturbation strategy to solve VRPSPD.

##### A. Generating initial Solution

The initial solution is generated by using Savings Algorithm [19] with additional consideration to vehicle capacity according to the enhanced formula in [20]. The saving formula is given in equation 11 and 12.

$$S_{ij} = c_{i0} + c_{0j} - \lambda c_{ij} + u|c_{0i} - c_{j0}| + v \frac{d'_i + d'_j}{\bar{d}} \quad (11)$$

$$d'_i = \max(d_i, p_i) \quad (12)$$

The first parameter  $\lambda$  help finding better solutions by redesign the routes. The parameter  $u$  give the result consideration of discrepancy the information between customers  $i$  and  $j$  regarding their distances to the depot. The parameter  $v$  gives priority information of customer with larger demands to assigned to the solution [21].

##### B. Variable Neighborhood Search

After the initial solution is obtained of saving algorithm, then this solution is refined by the improvement algorithm in several iterations. For this purpose, a VNS approach presented in [22] and [23] is used. This algorithm systematically changing neighborhoods to explore increasingly distant neighborhoods and improve current incumbent solution by exploring the solution space. Reference [16], [21], [23]–[25] presented the effectiveness of VNS algorithm for various VRP applications.

##### C. Shaking Step

VNS algorithm is composed by two main phases, namely shaking and local search. The shaking phase responsible to lead the search direction of the algorithm escape from the local optimum condition by restructuring 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 [16]. 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. Below are 3 three intra route neighborhood structure used in this paper:

- 1 *Exchange*: exchanging  $a$  sequential customer between two different tours.
- 2 *Cross*: This operator divides two different routes become two section for each route and exchange the sections between the routes.
- 3 *Shift (m,0)*: transfer  $m$  sequential customer from randomly selected route to another randomly selected route.

##### D. VND as Local Search Operator

After algorithm performs 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  $N_m$ ,  $m = 1, \dots, m_{max}$ , applying a set of neighborhood structures in a deterministic order. The sequence of intra-route movements considered in this work are:

- 1 *Insertion*: move a customer from its position and inserts to another position in the same route.
- 2 *Swap*: The best permutation movement that swaps the locations of two clients in the same route
- 3 *3-Opt*: moves some sub route to different position in the same route [26].
- 4 *2-Opt*: swaps pairs of links in the same route [27].

The following inter-route operator also used in the VND algorithm to control movement of nodes between two routes:

- 1 Replace (1,1): best exchange a customer from the selected route with a customer from another route which gaining less objective value.
- 2 Shift: transfer  $m$  sequential customer from one route to another.
- 3 Crossover: basic crossover structure between routes. In this structure find best possible crossover from two selected routes.

#### E. Feasibility Checking

The feasibility of VRPSPD solutions is different with the classical VRP solutions. In the VRPSPD solution, when the supply intensive clients serviced at the beginning of the trip, and the demand intensive clients are latter, the solution could become infeasible due to the fluctuation of loads in along the search even if the departure and the arrival loads are not exceeding the maximum capacity. However, reversing the sequence of an infeasible route could produce a feasible one. Therefore, the load feasibility of each route is checked in two steps. Firstly, the departure and arrival load are checked to the maximum capacity constraint, if feasible the load fluctuation also checked for each customer in the solution. In order to give more diversification degree, the fluctuation infeasible solution may be generated by shaking step allowed to enter local search step, with meaning the local search will refine the solution to better feasible solution.

#### F. Perturbation Strategy

The enchantment of VNS by adding perturbation mechanism is recently presented by [16] 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 Large Neighborhood Search algorithm is applied. This approach adopted from [28]. There are two main algorithms composed LNS, they are: the removal of customer and reinsertion of the removed customer. The removal phase using three removal methods, random removal method introduced by Shaw [29], relatedness removal method by [30] and long-arc-broken removal [28]. 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 [31] move customers from  $M$  one by one to the best position in solution while the greedy tournament algorithm selects 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. When only one customer is in the  $M'$ , then the algorithm denoted by greedy tournament 1. With the employment of tournament-based insertion, the combination of the result is increasing compared to the basic insertion method.

According to LNS structure, one perturbation set is composed by removal method and insertion method, thus from 3 removal method and 5 insertion method this paper composed 10 different perturbation scheme (Table 1).

Table 1. Removal & insertion method as perturbation schemes

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

#### G. Adaptive Selection Mechanism

In this section, the paper describes modification to basic Perturbation based Variable Neighborhood Search by [16] with additional adaptive capability to select better perturbation scheme

of the algorithm. Using different neighborhood methods may vary significantly change the performance of the algorithm and quality of result.

The performance of different neighborhood methods may vary significantly among different problems and instances [32]. Thus, instead stochastic approach, this study employed an adaptive mechanism to select new perturbation scheme that will move current solution to the new solution space. This approach embedded based on previous work by [28] that used adaptive mechanism to the perturbation step of iterated local search. According to 10 different perturbation set in table 5, adaptive mechanism assigns equal weight to each perturbation scheme at the beginning of the search process. During the search process, the weight of the schemes is updated depending on the success of each method. The profitability of each scheme is calculated by a scoring system below:

- If new overall best solution is obtained by a perturbation scheme, the scheme is directly selected as the next perturbation method and the score  $\theta_1$  is added to its weight, otherwise,
- The score  $\theta_2$  always is assigned after applying a perturbation scheme.

In the condition where a perturbation scheme is failed to make the algorithm find new best solution, the adaptive mechanism employ roulette wheel selection to find new perturbation set to be applied. This selection method has ever been successfully applied to simulated annealing by [33] and to iterated local search by [28].

Each perturbation scheme is chosen by roulette wheel procedure. This procedure uses a probability that is proportional to an empirical quality of obtained solutions. If there are  $p$  perturbation set with weight  $w_i, i \in \{1, 2, \dots, p\}$ , then the perturbation scheme  $j$  is selected using probability formula in equation 2.

$$Prob_j = \frac{w_j}{\sum_{k=1}^p w_k} \quad (13)$$

#### H. Perturbation-Based Variable Neighborhood Search with Adaptive Selection Mechanism

The main structure of this solution method is VNS algorithm, before the algorithms start searching the solution, a fast and effective construction method namely Saving Algorithm (SA) is performed to initialize the solution. Then the search process is handled by the VNS algorithm to intensively improve the initial solution over several iterations. In order to find local optimum on the iterations, we perform Variable Neighborhood Descent (VND) algorithm and corporate several determined neighborhood operators. Finally, to escape the search process from premature convergence, the local optimum obtained after the intensification process of VNS is perturbed by a perturbation strategy with a adaptive selection mechanism that records the successes search empirically and choose better perturbation scheme used. The structure of the approach is presented in Figure. 1 in pseudo-code.

### V. Computational results

In this section, the parameter settings of the PVNS ASM is provided, benchmark instances, and some comparative testing explained. The comparative result obtained by implementing the solution method to several scenarios and compared to exact method for analyzing the capability of the PVNS ASM method in finding optimal solution in VRPSD, basic VNS method in order to reveal the effect of the perturbation strategy, moreover this method also tested on a real-world VRPSD problem for analyzing the PVNS ASM performance in real-world scenario.

```

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 using equation 11 and 12
5.  $p = 1$ 
6. while  $p < p_{max}$ 
7.    $s = 1$ 
8.   while  $s < s_{max}$ 
9.      $k = 1$ 
10.    while  $k < k_{max}$ 
11.       $x^2 = x_k^1 \in N_k(x^1)$  Shaking:  $x_k^1$  is random solution in  $k^{th}$  neighborhood of  $x^1$ 
12.      If  $f(x^2) < f(x^1)$ 
13.         $x^1 = x^2, k = 1$ 
14.      else
15.         $x^3 = x^2, m = 1$ 
16.        while  $m < m_{max}$ 
17.           $x^3 = VND(N_m(X^2));$  local search with VND in  $m^{th}$  neighborhood of  $x^2$ 
18.          if  $f(x^3) < f(x^1)$ 
19.             $x^1 = x^3, k = 1$ 
20.          else
21.             $k = k + 1$ 
22.          end if
23.        end while
24.      end if
25.    end while
26.    If  $f(x^1) < f(x^0)$ 
27.       $x^0 = x^1$ ; move or not:  $x^0$  is a current best solution
28.      changeperturb = false
29.      Add  $\theta_1$  to  $c^{th}$  perturbation scheme
30.    else
31.       $s = s + 1$ 
32.    end if
33.  end while
34.  If (changeperturb = true)
35.     $c =$  select perturbation scheme by adaptive selection mechanism using equation 13
36.  endif
37.  changeperturb = true
38.   $x^1 = x_r^0 \in P_c(X_0)$ ; Perturbation:  $x_r^0$  is a random solution in the  $c^{th}$  of  $x^0$ 
39.  add  $\theta_2$  to  $c^{th}$  perturbation scheme
40. end while
41. return  $x = x^0$ 

```

Fig. 1: Structure of PVNS ASM Algorithm

The PVNS ASM approach explained in section 3 has been developed and implemented using Visual Studio Code. The VRPSD model formulation given in [18] has been implemented by use of the IBM ILOG CPLEX Optimization Studio 12.7 solver. All numerical experiments were run on a laptop with an Intel Core-i3 370M 2.4 GHz processor with 3 GB RAM and Manjaro 17.1 with Kernel Linux 4.14.8-1 LTS.

#### A. Test Problem

The algorithm is tested on two sets of benchmark instance with various size and characteristic. The first data set is generated based on [3] consisting of 4 scenarios with customer number vary between

10 and 40. Firstly a basic dataset consisting 40 customers is generated, then the remaining small instances (30,20,10) are derived from this first problem instances. All of these 4-scenario serviced by a set of homogenous vehicles whose capacity is 340 units. For this benchmark test, PVNS ASM compared to an exact algorithm using IBM CPLEX solver and basic VNS as comparative analysis. Each instance was allowed a maximum of 3600 seconds of CPU time (1 hours) using the CPLEX solver.

In the second benchmark test, PVNS ASM and basic VNS are applied to a real-world distribution instance derived provided by a Liquefied Petroleum Gas (LPG) Distributor named PT Arie Mulya Barutama which is located in West Semarang District, Semarang, Indonesia. Every distribution period, this company is responsible to deliver some amount of LPG the customers. This benchmark set involving 52 customers, 1 central depot and a set of fleets whose restricted capacity 35 units for every trip. The result then compared with the human planning that used by the company.

### B. Parameter Settings

A robust parameter setting is required for the algorithm to efficiently perform on different datasets. Thus, several tests were performed on two main processes of the algorithm involving initial solution construction, VNS algorithm and adaptive perturbation mechanism. The saving-based heuristic presented in section 3 contains three parameters  $\lambda$ ,  $\mu$ , &  $v$ .

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. In addition, the PVNS ASM algorithm uses two parameters to control number of iterations and perturbation calls. These parameters also control intensification and diversification process which affect the quality of solutions. The first parameter is VNS termination counter  $s_{max}$  and the second is the search process termination  $p_{max}$ . The last, the parameter of scoring system  $\theta_1$  is set to 5 and  $\theta_2$  is 10. Table 2 is the summary of the parameter settings used in this work.

Table 2. Parameter setting summary of the solution method

No	Parameter	Value
1	$\lambda$	3.5
2	$\mu$	1.6
3	$v$	1
4	$k_{max}$	3
5	$m_{max}$	7
6	$s_{max}$	$100 \times N$
7	$p_{max}$	$30 \times N$
8	$\theta_1$	5
10	$\theta_2$	10
11	$f(x)$	Equation 1

N = Number of Customers

### C. Result and Discussion

The following subsections present the computational results and discussion for the solution method for the VRPSPD

#### 1) Result of PVNS ASM for randomly generated instances

Table 3 compares results of the PVNS ASM approach to exact algorithm in finding optimal solution for first VRPSPD benchmark test. Table 3 presents the computational result of compared algorithm. Bold numbers given in the table indicate that the algorithms have reached the best solution. For every instance, the benchmarking test is performed 10 times for PVNS ASM, basic VNS and IBM



CPLEX. Objective value obtained by the corresponding method presented in Table 3 as Obj, the computation times needed for solving corresponding instance also given as CPU Time, from 10 search processes, we present best objective (best obj) and the average objective (Avg obj).

Table 3. Computational result of small-sized problem instances

No	Instance	Customer number	IBM CPLEX		Basic VNS			PVNS_ASM		
			Obj.	CPU time	Obj	Avg. Obj	CPU time	Best Obj.	Avg. Obj.	CPU time
1	case1	10	<b>333.24</b>	1.436	<b>275</b>	275	<b>1.0081</b>	<b>275</b>	275	1.0578
2	case2	20	<b>511.04</b>	56.326	513	530.3	<b>7.3037</b>	513	515.8	9.3953
3	case3	30	-	3600	<b>850.42</b>	867	<b>16.9093</b>	<b>864</b>	867.2	25.0937
4	case4	40	-	3600	<b>1110</b>	1117.9	<b>33.0881</b>	<b>1079</b>	1092.5	40.7171
Average					687.105	697.55	<b>14.5773</b>	<b>682.75</b>	687.625	19.06597

According to results of Table 3, all methods able to obtain optimal solution for small-sized problem (with 10 and 20 customers). However, CPLEX takes more time to reach the optimal solution with 1.436 seconds than PVNS ASM that just takes 1.0578 seconds for the search while the basic VNS takes fewer time with 1.0081s. For the instance that contains 20 customers, two metaheuristic methods produce worse solution quality than CPLEX with 0.0038% deviation but take fewer CPU times. In the larger instances (case 3 & 4) CPLEX unable to provide feasible solutions within the predefined time limit. On the other hand, the PVNS ASM shows its efficiency by yielding optimal solutions for the 30 and 40 customers instances within the predefined time limit.

The Basic VNS method produces feasible solutions but the best objective of the solution is less than the PVNS ASM results. According to the result of PVNS ASM and basic VNS, PVNS ASM outperform 2 of 4 instances given while the other 2 instances obtain equal solution. In average of best solutions, PVNS ASM reach 4.355 unit better. Different with the solution quality, it seems that PVNS ASM takes larger CPU Time than basic VNS, this is caused by the additional perturbation strategy in PVNS ASM that add more CPU Time in the search process. Even of this computational cost, we can conclude that the PVNS ASM shows its efficiency by outperforming the exact method in producing large instance and producing better solution than the basic VNS method.

## 2) Result of PVNS ASM for real-world instances

In this subsection, the PVNS ASM is performed to a real-world VRPSPD instance to test the performance and quality of the solution against existing human planning method by the company. In addition, the basic VNS also implemented in this test to present more comparison in the algorithm performance.

Table 4 reports the computational result of PVNS ASM for real-world problem instance. This simulation performed 10 times then the best result has presented. From table 4, it is seen that PVNS ASM is able to reach better solution with 107347 meter or 36874 (35.14%) units better than the human planning solution.

As a general result, PVNS ASM approach effectively outperforms exact algorithm for small-sized problem by using smaller amount of computation time while providing as optimally solution as exact algorithm. In the larger case PVNS ASM approach shows even better performance by reaching feasible solution within the given time limit. In comparison with basic VNS, PVNS ASM also has better solution with less objective value about 4.355 in average. Moreover, for the real-world application, this approach successful on reaching optimal solution compared to human planning method.

Table 4. Computational result of real-world problem instances

Method	Best Distance (m)	No. Routes
PVNS_ASM	107347	5
Human Planning	165527	5

Even though the PVNS ASM is tested only on simulation VRPSPD cases, the result shows that PVNS ASM is highly capable to applied in various condition problem even to real-world application. Moreover, the PVNS ASM able to obtain relatively optimal solution in small amount of computation time. Referring to the complexity of this problem, VRPSPD is very hard to be solved with human planning approach, even with exact computation approach. Thus, this result shows the effectiveness of the PVNS ASM algorithm for solving VRPSPD within relatively small computation time.

Since the VRPSPD is generalized from VRPMPD, this approach can be easily applied to VRPMPD instance directly, also this approach can be extended to solve many other VRP variants with little modification. Moreover, this algorithm basically is easy to understand and implement since diversification and intensification is handled separately with two nested algorithms. Thus, this algorithm is very extendable by any software developer in other combinatorial problem.

The findings provide a few more insightful implications. Firstly, it contributes to the vehicle routing literature by giving an implementation of a state-of-the-art approach for solving various VRPSPD and a real-world case as a reference for comparison purposes of future studies in VRPSPD literature. Moreover, the approach can be easily adapted and extended to solve other VRP variants as combinatorial optimization problems which need balance of intensification and diversification

## VI. Conclusion

In this study, an intelligent optimization software called enhanced perturbation-based neighborhood search with Adaptive Selection Mechanism (PVNS\_ASM) algorithm is developed. A saving algorithm is employed to generate the initial solution. Then the main algorithm composed based on Variable Neighborhood Search framework with additional perturbation strategy. The perturbation mechanism uses several destroy and repair scheme and we use an adaptive selection mechanism to control which perturbation scheme will be performed every time the perturbation performed. A Variable Neighborhood Descent algorithm employed as local search step in VNS, this algorithm uses three shaking operators and seven local search structures. Ten different destroy and repair structures were presented in order to avoid premature convergence by jumping the search process escape from local optima. The numerical results show that the PVNS ASM algorithm generated high-quality solutions compared to the exact approach used by IBM CPLEX and basic VNS algorithm for the given benchmark. For all the benchmark instances with various size, the PVNS ASM algorithm was able to reach to best generated exact solution with smaller amount of computation time. Furthermore, PVNS ASM also able to beat the given real-world instance with less objective value in relatively small computation time than traditional method used by the company. By looking at the results, it can be concluded that the PVNS ASM method can reduce premature convergence that raised in the basic VNS algorithm and clearly able to solve VRPSPD. With the capability to generate exact solution in small problem and optimal approach solution in larger scenario within limited computation time, it seems that PVNS ASM approach will satisfy the industrial challenge in solving VRPSPD with for various and bigger problem size effectively. As a future research direction, the implementation PVNS ASM algorithm can be analyzed for larger instance, heterogeneous fleet and with time restriction problem for closer to real-world application.

## Acknowledgment

This article was supported by Department of Electrical Engineering, Universitas Negeri Semarang. The authors are also grateful to all the member of UNNES Electrical Engineering Students Research Group (UEESRG) whose constructive comments and support have led to improving the quality of this paper.

## 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.
- [2] 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.

- [3] 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.
- [4] A. S. . Tasan and M. . c Gen, "A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries," *Comput. Ind. Eng.*, vol. 62, no. 3, pp. 755–761, 2012.
- [5] E. Angelelli and R. Mansini, "The Vehicle Routing Problem with Time Windows and Simultaneous Pick-up and Delivery," *Quant. Approaches to Distrib. Logist. Supply Chain Manag.*, pp. 249–267, 2002.
- [6] M. Dell'Amico, G. Righini, and M. Salani, "A Branch-and-Price Approach to the Vehicle Routing Problem with Simultaneous Distribution and Collection," *Transp. Sci.*, vol. 40, no. 2, pp. 235–247, May 2006.
- [7] A. Subramanian, E. Uchoa, A. A. Pessoa, and L. S. Ochi, "Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery," *Oper. Res. Lett.*, vol. 39, no. 5, pp. 338–341, 2011.
- [8] A. Subramanian, E. Uchoa, A. A. Pessoa, and L. S. Ochi, "Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery," *Optim. Lett.*, vol. 7, no. 7, pp. 1569–1581, 2013.
- [9] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, "Rejoinder on: Static pickup and delivery problems: a classification scheme and survey," *Top*, vol. 15, no. 1, pp. 45–47, 2007.
- [10] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems," pp. 21–51, 2008.
- [11] 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.
- [12] S. Akpinar, "Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem," *Expert Syst. Appl.*, vol. 61, pp. 28–38, 2016.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] F. A. T. Montané and R. D. Galvão, "Vehicle Routing Problems with Simultaneous Pick-up and Delivery Service," *Opsearch*, vol. 39, no. 1, pp. 19–32, 2002.
- [19] G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," no. August 2015, 1964.

- [20] İ. K. Altinel 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.
- [21] 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.
- [22] N. Mladenović and P. Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, 1997.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] S. Lin, "Computer Solutions of the Traveling Salesman Problem," *Bell Syst. Tech. J.*, vol. 44, no. 10, pp. 2245–2269, 1965.
- [27] G. A. Croes, "A Method for Solving Traveling-Salesman Problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] K. Chakhlevitch and P. Cowling, "Hyperheuristics: Recent developments," *Stud. Comput. Intell.*, vol. 136, pp. 3–29, 2008.
- [33] S. Ropke and D. Pisinger, "A unified heuristic for a large class of Vehicle Routing Problems with Backhauls," *Eur. J. Oper. Res.*, vol. 171, no. 3, pp. 750–775, 2006.