# Tabu Search for the Vehicle Routing Problem with Multiple Trips, Time Windows and Simultaneous Delivery-Pickup

**Suprayogi[1*], Yusuf Priyandari[2]**

**Abstract**: This paper discusses a vehicle routing problem with multiple trips, time windows, and simultaneous delivery-pickup (VRPMTTWSDP). This problem is a variant of the basic vehicle routing problem (VRP) including the following characteristics: multiple trips, time windows, and simultaneous delivery-pickup. In this paper, a solution approach based on tabu search (TS) is proposed. In the proposed TS, the sequential insertion (SI) algorithm is used to construct an initial solution. A neighbor structure is generated by applying an operator order consisting of eleven operators of relocation, exchange, and crossover operators. A tabu solution code (TSC) method is applied as a tabu restriction mechanism. Computational experiments are carried out to examine the performance of the proposed TS using hypothetical instances. The performance of the proposed TS is compared to the local search (LS) and the genetic algorithm (GA). The comparison shows that the proposed TS is better in terms of the objective function value.

**Keywords**: Vehicle routing problem, multiple trips, time windows, simultaneous delivery-pickup, tabu search.

## Introduction

This paper discusses a vehicle routing problem with multiple trips, time windows, and simultaneous delivery-pickup (VRPMTTWSDP). This problem is a variant of the basic vehicle routing problem (VRP) includeing the following characteristics: multiple trips, time windows, and simultaneous delivery-pickup. The characteristic of multiple trips indicates that each vehicle may perform more than one route or trip during a planning horizon. The time window constraint imposes that the service to each customer must be done in its time window. The last characteristic indicates that each customer has both delivery and pickup demands.

The VRP is one of important decision problems in the logistics area. The VRP attempts to create an efficient delivery and pickup of goods. One of real VRPs discussed in this paper is a delivery and pickup of the liquefied petroleum gas (LPG) tubes. Each LPG retailer has a delivery demand of filled tubes and a pickup demand of empty tubes. Each retailer has a time window where unloading of filled tubes and loading of empty tubes must be carried out. Trucks carrying filled tubes depart from the depot to the retailers and return to the depot to carry empty tubes from the retailers.

A lot of studies has been done the VRP by considering these three characteristics separately. The vehicle routing problem with multiple trips (VRPMT) was discussed, for example, by Taillard *et al.* [1], Brandão and Mercer [2], Brandão and Mercer [3], Zhao *et al.* [4], Petch and Salhi [5], and Salhi and Petch [6]. Some discussion of the vehicle routing problem with time windows (VRPTW) were found in Solomon [7], Potvin and Rousseau [8], Kolen *et al.* [9], Savelsbergh [10], Potvin *et al.* [11], and Potvin and Bengio [12]. Some researches in the vehicle routing problem with simultaneous delivery-pickup (VRPSDP) were Min [13], Dethloff [14], Dethloff [15], and Tang Montané and Galvão [16].

Some studies in the VRP with multiple trips and time windows (VRPMTTW) were Suprayogi [17], Suprayogi and Imawati [18], Suprayogi *et al.* [19], Azi *et al.* [20], Suprayogi *et al.* [21], Cattaruzza *et al.* [22], Hernandez *et al.* [23], and Hernandez *et al.* [24]. Ong and Suprayogi [25] investigated the VPTMTTW in context of backhaul operation. The VRP with time windows and simultaneous delivery-pickup (VRPTWSDP) was studied, for example, by Mingyong and Erbao [26], Fan [27], Wang and Chen [28], Liu *et al.* [29], Avci and Topaloglu [30], and Wang *et al.* [31]. To our best knowledge, The VRP with multiple trips, time window constraint, and simultaneous delivery-pickup (VRPMTTWSDP) was found only in Suprayogi and Priyandari [32], Suprayogi and Priyandari [33], and Suprayogi and Mahaputra [34].

In this paper, a solution approach based on tabu search (TS) is proposed. Various hard combinatorial optimization problems including the VRP and its variants are solved using the TS successfully.

[1] Faculty of Industrial Technology, Institut Teknologi Bandung, Jl. Ganesha 10, Bandung 40132, Indonesia.
Email: yogi@mail.ti.itb.ac.id

[2] Faculty of Engineering, Industrial Engineering Department, Universitas Sebelas Maret, Jl. Ir. Sutami 36A Surakarta 57126, Indonesia. Email: priyandari@ft.uns.ac.id
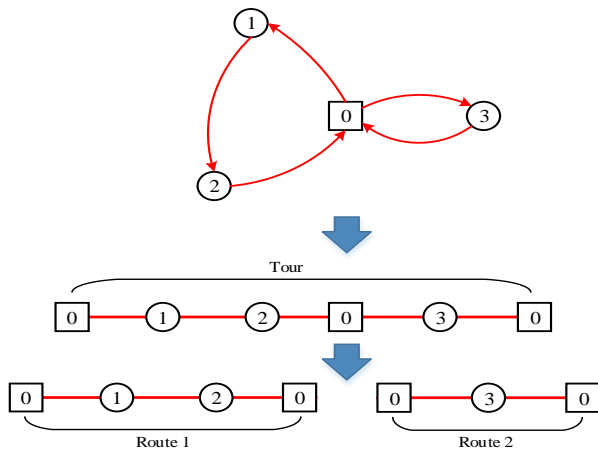
* Corresponding author

**Figure 1.** Illustration of a tour

The TS is one of metaheuristic approach capable to escape from the local optimal trap. Basic principles of the TS and its extensions is found in Glover [35] and Glover [36]. In the proposed TS, the sequential insertion (SI) algorithm is used to construct an initial solution. A neighbor structure is generated by applying an operator order consisting of eleven operators of relocation, exchange, and crossover operators. A tabu solution code (TSC) method is applied as a tabu restriction mechanism.

## Methods

### Problem Definition

In the context of the VRPMT, there are two different terms as mentioned in Brandão and Mercer [2], i.e., route (or trips) and tour. A route or trip is a sequence of visiting customers by a vehicle starting and ending at the depot. A tour is a collection of routes or trips served by the same vehicle. The number of tours is identical with the number of vehicles used. Figure 1 illustrates a tour consisting of two routes (number 0 is depot and numbers 1, 2, and 3 are customers).

The VRPMTTWSDP discussed in this paper is similar to the definition in Suprayogi and Mahaputra [34]. There is a depot and a set of customers. There is a set of vehicles with known capacity stationed at the depot and it is assumed that the number of vehicles is unlimited. Each customer has both delivery and pickup demands. A service time including unloading and loading time for each customer is given. Each customer has a time window indicating by its opening and closing times where it is assumed that the length of time window is constant and greater than or equal to the service time. If a vehicle arrives at a particular customer before its opening time, then the vehicle must wait and the service (unloading and loading) starts at the opening time. The departure time of the vehicle at a customer must be less than or equal to the closing

time. Figure 2 shows an illustration when a vehicle arrives at a customer before its opening time.

There are two main differences in the VRPMTTWS-DP defined in this paper compared to the definition in Suprayogi and Mahaputra [34]. In the VRPMTT-WSDP defined by Suprayogi and Mahaputra [34], there is no service time at the depot when a vehicle starts to serve a route. But, when a vehicle returns to the depot, a service time occurs. In the VRPMTTWSDP defined in this paper, there is a loading time at the depot when a vehicle starts to serve a route and an unloading time when a vehicle returns to the depot as illustrated in Figure 3. The second difference is in the objective function. In Suprayogi and Mahaputra [34], the objective function is a weighted sum of three objectives to be minimized: number of vehicles (*NV*), total tour duration time (*TDT*), and range of tour duration time (*RDT*). In this paper, the objective function is a weighted sum of three objectives to be minimized: Number of vehicles (*NV*), total tour duration time (*TDT*), and maximum of tour duration time (*MDT*). Both minimizing RDT and MDT attempts to make a balance among tours in terms of tour duration time.

Let $\theta$ be a solution. The objective function is expressed as:
$$f(\theta) = w_{NV}NV(\theta) + w_{TDT}TDT(\theta) \\ + w_{RDT}MDT(\theta) \qquad (1)$$
where $w_{NV}$, $w_{TDT}$ dan $w_{RDT}$ are weights associated with number of vehicles (*NV*), total tour duration time (*TDT*), and maximum of tour duration time (*MDT*). To ensure the priority order, weights are chosen such that $w_{NT}NT(\theta) > w_{TDT}TDT(\theta) > w_{RDT}RDT(\theta)$.

The VRPMTTWSDP deals with determining a solution (tour plan) minimizing the objective function expressed in equation (1) by satisfying the following constraints: (1) each route must start and end at the depot, (2) total load of any route cannot be greater than the vehicle capacity, (3) all customer must be visited, (4) each customer is visited once, (5) each tour is served only by a vehicle, (6) time to start the service at each customer cannot be done before its opening time, and (7) time to finish the service or departure time at each customer cannot be done after its closing time. Constraint (5) implies that the number of tours are equal to the number of vehicles. Constraints (5) and (6) are also applied for the depot. It implies that a tour duration time may not violate the length of planning horizon.

### Tabu Search Approach

#### *Initial Solution*

The proposed TS approach requires an initial solution. In this paper, the initial solution is constructed
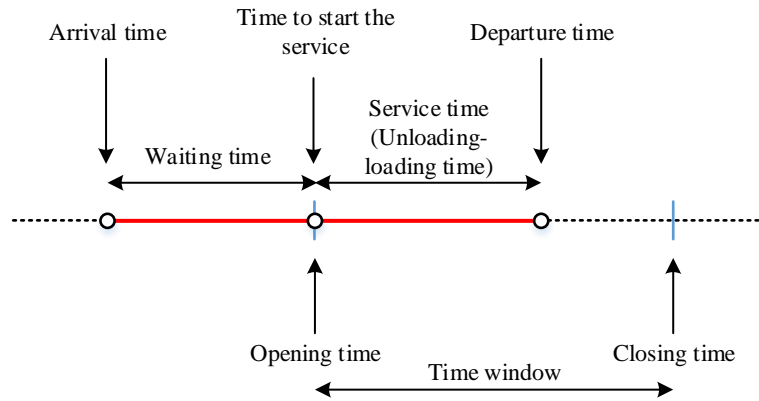
**Figure 2.** Illustration when a vehicle arrives at a customer before its opening time
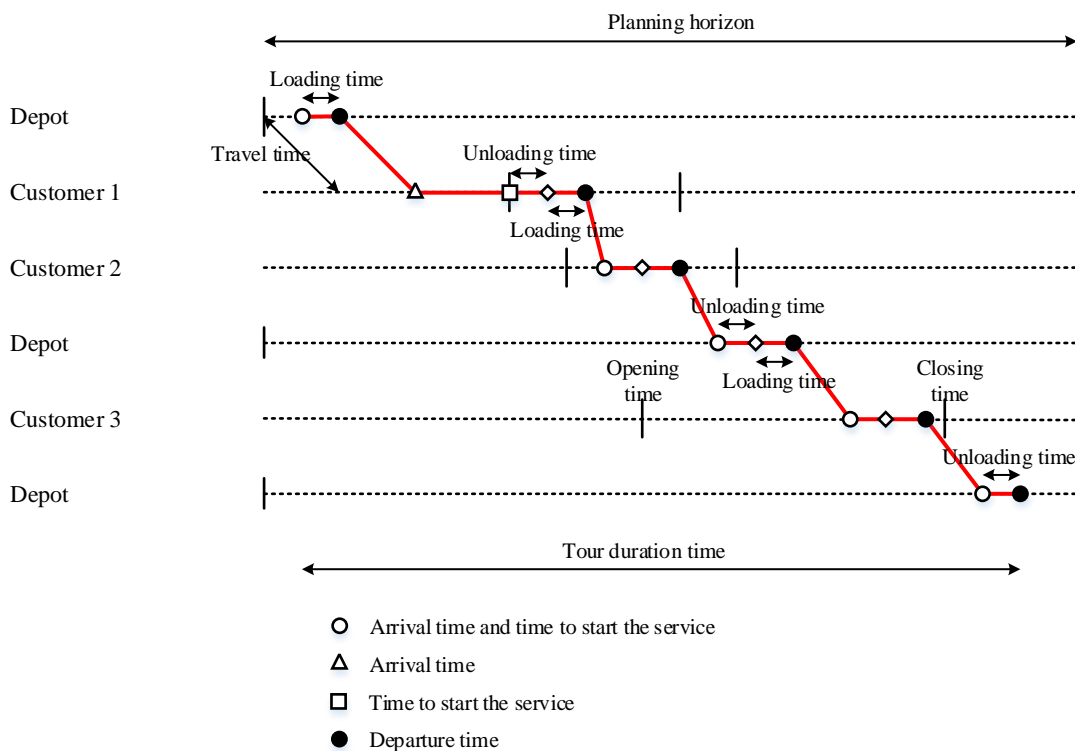


**Figure 3.** Illustration of schedule

using a sequential insertion (SI) algorithm proposed by Suprayogi and Priyandari [32]. The SI algorithm is a construction heuristic method. Basically, it attempts to insert an unrouted customer to an insertion position of the current route.

In the SI algorithm for the VRPMTTWSDP, there are two decisions must be made. The first decision is to select an unrouted customer as a seed customer for the first route in each tour. The second decision is to select an unrouted customer as a seed customer for the subsequent routes for a tour. For the first decision, there are several criteria that can be applied. Solomon [7] has proposed two criteria in the application of the SI algorithm for the VRPTW: the unrouted customer with the farthest distance from

the depot and the unrouted customer with the earliest closing time. Suprayogi [17] and Suprayogi and Imawati [18] used four criteria in the SI algorithm for the VRPMTTW: the unrouted customer with the earliest opening time, the unrouted customer with the earliest closing time, the unrouted customer with the earliest closing time, the unrouted customer with the shortest length of time window, and the unrouted customer with the farthest distance from the depot. Suprayogi and Priyandari [32] apply the same criteria with Suprayogi [17] and Suprayogi and Imawati [18] in the context of the VRPMTTWSDP. For the second decision, in this paper, the criterion of selecting an unrouted customer giving the minimum tour completion time is applied.

Let t and r be tour and route indices, respectively. Details of the SI algorithm are as follows:
1. Set $t = 1$ and $r = 1$.
2. If $r = 1$, then choose a seed customer from unrouted customers using a particular criterion, update route $r$ and go to step 3. If $r > 1$, then try to create a route $r$ by selecting an unrouted customer. If there are feasible customers, then choose a seed customer from unrouted customer giving the minimum completion time and update route $r$. Otherwise, cancel the construction of route $r$, set $t = t + 1$, $r = 1$, and repeat step 2.
3. If there is no unrouted customer, then stop. Otherwise, go to step 4.
4. Try to insert each unrouted customer to each insertion position on route $r$.
5. If there are feasible insertions with respect to capacity and time window constraints, then insert an unrouted customer to a position on route $r$ with the minimum duration time, update route $r$, and back to step 3. Otherwise, go to step 6.
6. If there is no feasible insertion with respect to the capacity constraint, then set $r = r + 1$ and back to step 2. Otherwise, go to step 7.
7. If there is no feasible insertion with respect to the time window constraint, then set $t = t + 1$, $r = 1$, and back to step 2.

### Neighborhood Structure

The TS attempts to improve the current best solution by generating its neighbors. Neighbors are generated using an operator order of eleven operators from Suprayogi and Priyandari [33] composing: (1) intertour relocation 1-0, (2) intertour relocation 2-0, (3) intratour relocation 1-0, (4) intratour relocation 2-0, (5) intertour exchange 1-1, (6) intertour exchange 2-1, (7) intertour exchange 2-2, (8) intratour exchange 1-1, (9) intratour exchange 2-1, (10) intratour exchange 2-2, and (11) crossover.

Intertour relocation (1, 0) relocates one customer from a route of a tour to a position from a route of another tour. Intertour relocation (2, 0) is similar to intertour relocation (1, 0) by relocating two consecutive customers. Intratour relocation (1, 0) and intratour (2, 0) are relocation applied to same tour. Intertour exchange (1, 1) swaps one customer in a position from a route of a tour with one customer in a position from a route of another tour. Intertour exchange (2, 1) swaps customers in two consecutive positions of a route of a tour with one customer in a position from a route of another tour. Intertour exchange (2, 2) is similar to intertour exchange (1, 1) by exchanging customers in two consecutive positions between different tours.
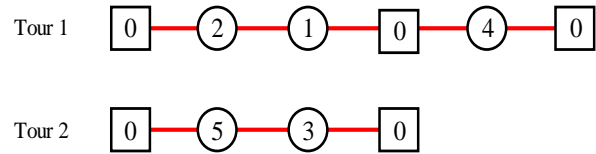


**Figure 4.** Example of a solution

**Table 1.** Illustration of the CS calculation

| $t$ | $r$ | $k$ | $L_{trk}$ | $NT$ | $NR_t$ | $NL_{tr}$ | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 2 | 2 | 3 | 0 |
| 1 | 1 | 1 | 2 | 2 | 2 | 3 | 24 |
| 1 | 1 | 2 | 1 | 2 | 2 | 3 | 24 |
| 1 | 1 | 3 | 0 | 2 | 2 | 3 | 0 |
| 1 | 2 | 0 | 0 | 2 | 2 | 2 | 0 |
| 1 | 2 | 1 | 4 | 2 | 2 | 2 | 64 |
| 1 | 2 | 2 | 0 | 2 | 2 | 2 | 0 |
| 2 | 1 | 0 | 0 | 2 | 1 | 3 | 0 |
| 2 | 1 | 1 | 5 | 2 | 1 | 3 | 60 |
| 2 | 1 | 2 | 3 | 2 | 1 | 3 | 72 |
| 2 | 1 | 3 | 0 | 2 | 1 | 3 | 0 |
| | | | $CS$ | | | | 244 |

Intratour exchange (1, 1), intratour exchange (2, 1), and intratour exchange (2, 2) are applied for same tour. Crossover divides two different tours in a two parts and exchanges their tail parts.

### Tabu Restriction

In the proposed TS, a tabu solution code (TSC) is used as a tabu restriction mechanism adapted from Wassan and Osman [37]. This TSC transforms a candidate to a value (denoted by *CS*), using the following equation:

$$CS = \sum_{t=1}^{NT} \sum_{r=1}^{NR_t} \sum_{k=0}^{NL_{tr}} t \times r \times k \times L_{trk} \times NT \times NR_t \times NL_{tr} \quad (2)$$

where $t$ is tour index, r is route index on tour $t$, k *is customer position index on* route $r$ of tour $t$, $NT$ is number of tour, $NR_t$ is number of routes of tour $t$, $NL_{tr}$ is number of customers on route $r$ of tour $t$, and $L_{trk}$ is customer number on position $k$ on route $r$ of tour $t$. Table 1 shows an example of the CS calculation for a solution giving in Figure 4.

The *CS* value for a particular iteration is stored in a tabu list called TABUSOLUTION. TABUSOLUTI-ON is an array consisting of tree columns. The first, second, and third columns in TABUSOLUTION record iteration number, *CS* value, and length of solution associated with *CS* value has been stored. The duration of the CS value stored in the list depends tabu tenure. The *CS* value is used as a tabu restriction. If the *CS* value of a candidate is equal to the *CS* values in the tabu list, then the candidate has a tabu status.

### Tabu Tenure

*Tabu tenure is a duration (in a number of iterations) to retain a solution with tabu status in the tabu list.* In the proposed TS, a fixed tabu tenure is implemented. In the proposed TS, tabu tenure is denoted by TabuTenure.

### Aspiration Criterion

*An aspiration criterion is a mechanism to accept a candidate solution with tabu status.* In this proposed TS, a candidate with tabu status is accepted if its objective function value is better than the current best objective function value.

### Stopping Criterion

The maximum number of iterations is used as the stopping criterion for the proposed TS. Here, the maximum number of iterations is denoted by MaxIter.

### Diversificatiom

In the proposed TS, restart diversification is implemented. If there is no improvement in the current best solution during a certain number of iterations (here, denoted by MaxDivIter), then the initial solution obtained by the SI algorithm is used to generate neighbors.

### Details of TS procedure

The following notations are used in the TS procedure:

| | |
|---|---|
| ImpInOneOpr | boolean variable indicating there is an improvement in a particular operator. |
| ImpInAllOpr | boolean variable indicating there is an improvement in the operator order |
| Iter | counter for iteration |
| DivIter | counter for indicating there is no improvement |
| TrialBest | trial best objective function value |
| Best | best objective function so far |
| MaxIter | maximum number of iterations |
| MaxDivIter | maximum number of iterations if there is no improvement |

The TS procedure is described as follows:

Create an initial solution $\theta^0$ using the SI algorithm. Set $\theta' = \theta^0$, $\theta^* = \theta'$, $f(\theta') = f(\theta^0)$, and Best = $f(\theta^0)$.

Set Iter = 0.
Set DivIter = 0.
Set TrialBest = $\infty$.
Do
    Do
        Iter = Iter + 1.
        Set ImpInAllOpr = false.
        Perform the 1st operator.
        Perform the 2nd operator.
        Perform the 3rd operator.
        Perform the 4th operator.
        Perform the 5th operator.
        Perform the 6th operator.
        Perform the 7th operator.
        Perform the 8th operator.
        Perform the 9th operator.
        Perform the 10th operator.
        Perform the 11th operator.
    Until ImpInAllOpr = false.
    If TrialBest < Best then
        Set $\theta^* = \theta'$ and Best = $f(\theta')$.
        Set DivIter = 0
    Else
        Set DivIter = DivIter + 1
    End if
    If DivIter ≥ MaxDivIter
        Set $\theta' = \theta^0$
        DivIter = 0
    End If
Until Iter ≥ MaxIter
Return $\theta^*$ as the best solution.

The procedure for each operator is given as follows:

Do
    Set EachOprImp = false,
    Create a neighbor $\theta$ from the current best solution $\theta'$ lexicographically.
    If $\theta$ is feasible and $f(\theta) < $ TrialBest then
        If tabu = true then
            If $f(\theta) < $ Best then
                Set $\theta' = \theta$
                Set TrialBest = $f(\theta)$
                Set ImpInOneOpr = true
                Set ImpInAllOpr = true
                Record the solution $\theta$ to the tabu list
            End If
        Else
            Set $\theta' = \theta$
            Set TrialBest = $f(\theta)$
            Set ImpInOneOpr = true
            Set ImpInAllOpr = true
        End If
    Else
    End If
Until ImpInOneOpr = false

**Table 2.** Comparison of experiment results between TS and LS

| Instance | TS | | | | | LS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | NV | TDT | MDT | OFV | CT (seconds) | NV | TDT | MDT | OFV | CT (seconds) |
| R1 | 7.00 | 1741.55 | 420.43 | 700696.63 | 66.76 | 8.00 | 1750.22 | 478.68 | 800700.13 | 20.86 |
| R2 | 4.00 | 1348.33 | 463.17 | 400539.34 | 193.81 | 4.00 | 1368.72 | 463.42 | 400547.50 | 17.35 |
| R3 | 7.00 | 1332.02 | 434.07 | 700532.81 | 151.04 | 7.00 | 1352.36 | 393.86 | 700540.94 | 40.99 |
| C1 | 7.00 | 1701.00 | 466.32 | 700680.44 | 112.07 | 7.00 | 2008.24 | 455.48 | 700803.31 | 20.01 |
| C2 | 3.00 | 1338.96 | 463.18 | 300535.63 | 280.17 | 3.00 | 1349.02 | 453.90 | 300539.63 | 29.90 |
| C3 | 6.00 | 1342.51 | 484.53 | 600537.00 | 112.75 | 6.00 | 1353.44 | 494.42 | 600541.38 | 39.21 |
| M1 | 6.00 | 1587.97 | 413.38 | 600635.19 | 106.04 | 7.00 | 1798.27 | 434.57 | 700719.31 | 28.83 |
| M2 | 3.00 | 1328.48 | 473.40 | 300531.41 | 189.65 | 3.00 | 1338.95 | 473.65 | 300535.59 | 12.16 |
| M3 | 6.00 | 1393.43 | 373.14 | 600557.38 | 169.69 | 6.00 | 1393.62 | 383.14 | 600557.50 | 33.86 |
| Average | 5.44 | 1457.14 | 443.51 | 545027.31 | 153.55 | 5.67 | 1523.65 | 447.90 | 567276.14 | 27.02 |

**Table 3.** Comparison of experiment results between TS and GA

| Instance | TS | | | | | GA (Best solution in 5 replications) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | NV | TDT | RDT | OFV | CT (seconds) | NV | TDT | RDT | OFV | CT (seconds) |
| R1 | 7.00 | 1892.46 | 295.54 | 700757.00 | 66.20 | 8.00 | 3515.09 | 111.59 | 801406.10 | 122.00 |
| R2 | 4.00 | 1358.57 | 402.64 | 400543.44 | 158.65 | 4.00 | 1704.54 | 312.18 | 400681.80 | 59.00 |
| R3 | 7.00 | 1350.95 | 373.42 | 700540.38 | 133.15 | 6.00 | 2905.94 | 112.11 | 600195.30 | 84.00 |
| C1 | 8.00 | 2060.11 | 306.08 | 800824.06 | 55.00 | 7.00 | 3188.66 | 59.05 | 701275.70 | 115.00 |
| C2 | 3.00 | 1328.92 | 41.17 | 300531.56 | 178.68 | 4.00 | 1713.20 | 157.70 | 400685.30 | 61.00 |
| C3 | 7.00 | 1343.30 | 333.25 | 700537.31 | 102.56 | 5.00 | 2575.27 | 8.77 | 501030.10 | 78.00 |
| M1 | 6.00 | 1805.19 | 361.13 | 600722.06 | 82.17 | 7.00 | 2718.07 | 115.38 | 701148.50 | 168.00 |
| M2 | 3.00 | 1328.70 | 71.61 | 300531.47 | 179.27 | 4.00 | 1729.48 | 173.50 | 400691.80 | 60.00 |
| M3 | 5.00 | 1340.70 | 342.82 | 500536.31 | 101.17 | 5.00 | 2563.62 | 13.69 | 501025.40 | 76,0 |
| Average | 5.56 | 1534.32 | 280.85 | 556169.29 | 117.43 | 5.56 | 2512.65 | 118.22 | 556460.00 | 93.38 |

## Results and Discussions

### Hypothetical Instances

The proposed TS approach is examined using nine hypothetical instances taken from Suprayogi and Mahaputra [34]. In each instance, there are 100 customer points where their locations are generated randomly in *x-y* coordinate ranging from (0, 0) to (600, 600). The depot is located in coordinate (300, 300). A Euclidean distance for each pair of points is used. The distances are rescaled by dividing them by 60. The travel time between two points is calculated by dividing the distance to vehicle speed of 40. The length of time window of the depot is set with opening and closing times are 0 and 540, respectively. The length of time window for customers is generated randomly between 0 and 540. The unloading and loading times are 5 for each customer and the depot. The delivery and pickup demands for each customer are generated randomly from 1 to 10. The vehicle capacity is set as 20. The weights for number of vehicles (NV), total tour duration time (TDT), and maximum of tour duration time (MDT) in the objective function are set as 100000, 0.4, and 0.00005, respectively.

The proposed TS approach is implemented in Microsoft Visual Basic 6.0 and run on a laptop computer with the following specifications: Intel Core i5-3317U CPU @ 1,70 GHz processor, 4,0 GB memory, and Windows 8 64-bit operating system.

.
### Parameter Settings for the Proposed TS

For the computational experiments, the following TS parameters are set as follows: (1) criterion for selecting the first customer on the first route of each tour in the SI algorithm is the unrouted customer with the earliest closing time, (2) improvement operator order: 2, 1, 11, 7, 6, 5, 4, 3, 10, 9, and 8, (3) TabuTenure = 10, (4) MaxDivIter = 10, and (5) MaxIter = 25

### Experiment Results

Table 2 reports computational results of the proposed TS and the LS. In the LS, the operator order is same to the TS. From the experiments, it shown that the average objective function value is 3.92% lower than the LS. But, the computation time is higher 5.68% compared to the LS.

The proposed TS is compared to the genetic algorithm (GA) proposed previously by Suprayogi and Mahaputra [34]. In order to make a comparison, the hypothetical instances are modified by setting loading and loading times at the depot as 0 and 10 minutes, respectively. In addition, the third component in the objective function is also changed from the maximum of tour duration time (MDT) to the range of tour duration time (RDT). The GA from Suprayogi and Mahaputra [34] was implemented in Visual Basic 6.0 and run on a personal computer with Intel Pentium IV 1.8 GHz processor, DDRAM 256 MB memory, and operating system of Microsoft Windows XP Professional Edition. For each instance, the GA is run in 5 replications. The experiment results are reported in Table 3. The average of objective function from the TS is 0.05% lower compared to the GA. The average computation time for the GA is 93.38 seconds. Because the GA is run in 5 replications, the computation time to obtain the best solution for each instance is about 466.88 seconds and it is 2.97 seconds higher compared to the TS.

## Conclusion

This paper has discussed the tabu search (TS) for solving the vehicle routing problem with multiple trips, time windows, and simultaneous delivery-pickup (VRPMTTWSDP). In the proposed TS, the initial solution is constructed using the sequential insertion (SI) algorithm. Eleven operators consisting of relocation, exchange, and crossover operators are applied as improvement operators. A tabu solution code (TSC) is implemented as a tabu restriction mechanism.

Computational experiments have been carried out to test the performance of the proposed TS. From the computational experiments, it is shown that the proposed TS gives a better performance in terms of objective function value compared to the LS and the GA. However, if statistical tests are applied, the differences in the average objective function values do not differ significantly. Therefore, further researches can be done to improve the proposed TS so that it gives a better performance.

## References

1. Taillard, É. D., Laporte, G. and Gendreau, M., Vehicle Routeing with Multiple Use of Vehicles, *Journal of the Operational Research Society*, 47(8), 1996, pp. 1065-1070.
2. Brandão, J. and Mercer, A., A Tabu Search Algorithm for the Multi-Trip Vehicle Routing and Scheduling Problem, *European Journal of Operational Research,* 100(1), 1997, pp. 180-191.
3. Brandão, J. C. S. and Mercer, A., The Multi-Trip Vehicle Routing Problem, *Journal of the Operational Research Society*, 49(8), 1998, pp. 799-805.
4. Zhao, Q. -H., Wang, S. -Y., Lai, K. K. and Xia, G. -P., A Vehicle Routing Problem with Multiple Use of Vehicles, *Advanced Modeling and Optimization*, 4(3), 2002, pp. 31-40.
5. Petch, R. J. and Salhi, S., A Multi-Phase Constructive Heuristic for the Vehicle Routing Problem With Multiple Trips, *Discrete Applied Mathematics*, 133(1-3), 2004, pp. 69-92.
6. Salhi, S. and R. J. Petch, A GA based heuristic for the vehicle routing problem with multiple trips, *Journal of Mathematical Modelling and Algorithms,* 6(4), pp. 591-613, 2007.
7. Solomon, M. M., Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints, *Operations Research*, 35(2), 1987, pp. 254-265.
8. Potvin, J. -Y. and Rousseau, M., Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows, *European Journal of Operational Research*, 66(3), 1993, pp. 331-340.
9. Kolen, A. W. J., Rinnoy-Kan, A. H. G., and Trienekens, H. W. J. M., Vehicle Routing with TimeWindows, *Operations Research*, 35(2), 1987, pp. 266-273.
10. Savelsbergh, M. W. P., The Vehicle Routing with Time Windows: Minimizing Route Duration, *ORSA Journal on Computing*, 4(2), 1992, pp. 146-154.
11. Potvin, J. -Y., Kervahut, T., Garcia, B. -L. and Rosseau, J. -M., The Vehicle Routing Problem with Time Windows - Part I: Tabu Search, *INFORM Journal on Computing*, 8(2), 1996. pp. 158-164.
12. Potvin, J. -Y. and Bengio, S., The Vehicle Routing Problem with Time Windows Part II: Genetic Search, *INFORMS Journal on Computing*, 8(2), 1996, pp. 165-172.
13. Min, H., The Multiple Vehicle Routing Problem with Simultaneous Delivery and Pick-Up Point, *Transportation Research*, 23A(5), 1989, pp. 377-386. .
14. Dethloff, J., Vehicle Routing and Reverse Logistics: The Vehicle Routing Problem with Simultaneous Delivery And Pick-Up, *OR Spektrum*, 23, 2001, pp. 79–96.
15. Dethloff, J., Relation between Vehicle Routing Problems: An Insertion Heuristic for the Vehicle Routing Problem with Simultaneous Delivery and Pick-Up applied to the Vehicle Routing Problem With Backhauls, *Journal of the Operational Research Society,* 53, 2002, pp. 115-118.
16. Tang Montané, F. A. and Galvão, R. G., A Tabu Search Algorithm for the Vehicle Routing

Problem with Simultaneous Pick-Up and Delivery Service, *Computers & Operations Research*, 33(3), 2006, pp. 595–619.

17. Suprayogi, Algoritma Sequential Insertion untuk Memecahkan Masalah Vehicle Routing Problem dengan Multiple Trips dan Time Windows, *Jurnal Teknik dan Manajemen Industri*, 23(3), 2003, pp. 21-36.

18. Suprayogi and Imawati, D., Algoritma Sequential Insertion dengan Forward dan Backward Pass untuk Memecahkan Vehicle Routing Problem with Multiple Trips and Time Windows, *Jurnal Teknik dan Manajemen Industri*, 25(1), 2005, pp. 41-54.

19. Suprayogi, Hardianto, A., and Yamato, H., Local Search and Genetic Algorithm Techniques for Solving Vehicle Routing Problem with Multiple Trips and Time Windows, *Proceedings of International Conference on Operations and Supply Chain Management*, Bali, Indonesia, 2005.

20. Azi, N., Gendreau, M. and Potvin, J. -Y., An Exact Algorithm for a Vehicle Routing Problem with Time Windows and Multiple Use of Vehicles, *European Journal of Operational Research*, 202, 2010, pp. 756–763.

21. Suprayogi, Hidayat, Y. A. and Imawati, D., Simulated Annealing untuk Pemecahan Masalah Rute dan Jadwal Kendaraan dengan Trip Majemuk dan Jendela Waktu, *Proceedings of 6th National Industrial Engineering Conference 2011*, Surabaya, Indonesia, 2011.

22. Cattaruzza, D, Asbi, N. and Feillet, D., The Multi Trip Vehicle Routing Problem with Time Windows and Release Dates, *Transportation Science*, 50(2), 2016, pp. 676-693.

23. Hernandez, F., Feillet, D., Giroudeau, R. and Naud, O., A New Exact Algorithm to Solve the Multi-Trip Vehicle Routing Problem with Time Windows and Limited Duration, *4QR*, 12, 2014, pp. 235-259.

24. Hernandez, F., Feillet, D., Giroudeau, R. and Naud, O., Branch-and-Price Algorithms for the Solution of the Multi-Trip Vehicle Routing Problem with Time Windows, *European Journal of Operational Research*, 249(2), 2016, pp. 551–559.

25. Ong J. O. and Suprayogi, Vehicle Routing Problem with Backhaul, Multiple Trips and Time Windows, *Jurnal Teknik Industri*, 13(1), pp. 1-10, 2011.

26. Mingyong, L. and Erbao, C. An Improved Differential Evolution Algorithm for Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows, *Engineering Applications of Artificial Intelligence*, 23 (2), 2010, pp. 188–195.

27. Fan, J., The Vehicle Routing Problem with Simultaneous Pickup and Delivery based on Customer Satisfaction, *Procedia Engineering*, 15, 2011, pp. 5284-5289.

28. Wang, H. -F. and Chen, Y. Y., A Genetic Algorithm for the Simultaneous Delivery and Pickup Problems with Time Windows, *Computers & Industrial Engineering*, 62(1), 2012, pp. 84–95.

29. Liu, R., Xie, X., Augusto V. and Rodriguez, R., Heuristic Algorithms for a Vehicle Routing Problem with Simultaneous Delivery and Pickup and Time Windows in Home Health Care, *European Journal of Operational Research*, 230(3), 2013, pp. 475–486.

30. Avci, M. and Topaloglu, S., An Adaptive Local Search Algorithm for Vehicle Routing Problem with Simultaneous and Mixed Pickups and Deliveries, *Computers & Industrial Engineering*, 83, 2015, pp. 15–29.

31. Wang, C., Mu, D., Zhao, F. and Sutherland, J. W., A Parallel Simulated Annealing Method for the Vehicle Routing Problem with Simultaneous Pickup–Delivery and Time Windows, *Computers & Industrial Engineering*, 83, 2015, pp. 111–122.

32. Suprayogi and Priyandari, Y., Algoritma Sequential Insertion untuk Memecahkan Vehicle Routing Problem dengan Multiple Trips, Time Windows dan Simultaneous Pickup Delivery, *Performa*, 7(1), 2008, pp. 88-96.

33. Suprayogi and Priyandari, Y., Vehicle Routing Problem with Multiple Trips, Time Windows, and Simultaneous Delivery and Pickup Services, *Proceedings of Asia Pacific Industrial Engineering and Management Systems Conference*, Kitakyushu, Japan, 2009, pp. 1543-1552.

34. Suprayogi and Mahaputra, M. S., Pemecahan Masalah Rute Kendaraan dengan Trip Majemuk, Jendela Waktu dan Pengantaran-Penjemputan Simultan menggunakan Algoritma Genetika, *J@TI Undip: Jurnal Teknik Industri*, 12(2), 2017, pp. 95-104.

35. Glover, F., Tabu Search - Part I, *ORSA Journal on Computing*, 1(3), 1989, pp. 190-206.

36. Glover, F., Tabu Search - Part II, *ORSA Journal on Computing*, 2(1), 1990, pp. 4-32.

37. Wassan, N. A. and Osman, I. H., Tabu Search Variant for the Mix Fleet Vehicle Routing Problem, *Journal of the Operational Research Society*, 53 (7), 2002, pp. 768-782.