MULTI-TRIP VEHICLE ROUTING PROBLEM WITH BACKHAULS AND TIME WINDOWS

Suprayogi
Faculty of Industrial Technology, Institut Teknologi Bandung,
Bandung, Indonesia, E-mail: yogi@mail.ti.itb.ac.id

Rully Tri Cahyono
Faculty of Industrial Technology, Institut Teknologi Bandung,
Bandung, Indonesia, E-mail: rully.mesgapati@gmail.com

Alek Topani Lubis
Faculty of Industrial Technology, Institut Teknologi Bandung,
Bandung, Indonesia, E-mail: topanalex@gmail.com

ABSTRACT

This paper presents a variant of the basic vehicle routing problem (VRP) called the multi-trip vehicle routing problem with backhauls and time windows (MTVRPBTW). Two objectives are considered: minimizing the number of vehicles and minimizing the total tour duration time. These two objectives are handled into a single objective by defining a wighted sum. A mixed integer linear programming (MILP) model is formulated to represent the MTVRPBTW. Variable neighborhood descent (VND) is proposed to solve the MTVRPBTW. In this paper, two schemes of VND is proposed depending on the order of neighborhood structures applied.

Keywords: vehicle routing problem; backhauls; time windows; multiple trips; variable neighborhood descent

1. INTRODUCTION

The basic vehicle routing problem (VRP) first studied by Dantzig & Ramser (1959) deals with a system consisting delivery or pickup services only, not both. In the delivery service, every customer requires a demand of a particular commodity to be delivered from the depot. In other hand, the pickup service addresses a situation in which every customer in the system has a demand to be picked up and brought to the depot. A customer requiring a commodity delivery called linehaul customer, while a customer requesting a commodity pickup is called backhaul customer. There are three common schemes to handle the VRP with linehaul and backhaul customers as mentioned by Reimann & Ulrich (2006).

The first scheme is to create routes separately for linehaul and backhaul customers. Every route consists of entirely linehaul or backhaul customers. Some vehicles are designated to linehaul customers and others to backhaul customers. Therefore, the first scheme actually has two independent basic VRP as mentioned by Ropke & Pisinger (2006). An illustration for the first way is shown in Figure 1.

The second scheme is to construct routes in which a vehicle route may consist of a mixture of linehaul and backhaul customers. There is no precedence constraint imposed. The linehaul customers can be visited after backhaul customers or vice versa. In other words, linehaul and backhaul customers can take any order on a certain route. Figure 2 shows an illustration for the second scheme.

The third scheme is to construct routes in which there is a restriction that the linehaul customers must be visited after the backhauls customers on any route. Therefore, the precedence constraints exist in the third scheme. The illustration for the third scheme is depicted in Figure 3. The third scheme is motivated by some reasons. The first reason is that in practical applications, by the fact, linehaul customers have higher priority. This reason is mentioned by Toth & Vigo, (1999), Crispim & Brandão (2001), Yazgitutuncu et al. (2009), and Irnich et al. (2014). The second reason is related to avoid the difficulty if the second scheme is implemented. By the fact, often vehicles are rear-loaded. Rearranging the loads on the vehicles at the customer locations is not deemed economical or feasible. This reason is stated by Goetschalckx & Jacobs-Blecha (1989), Crispim & Brandão (2001), Yazgitutuncu et al. (2009), Irnich et al. (2014), and (Koç & Laporte, 2018). The last reason is to reduce cost and ecological impacts if the first scheme is applied as mentioned by Brandão (2006) and Koch et al. (2017).

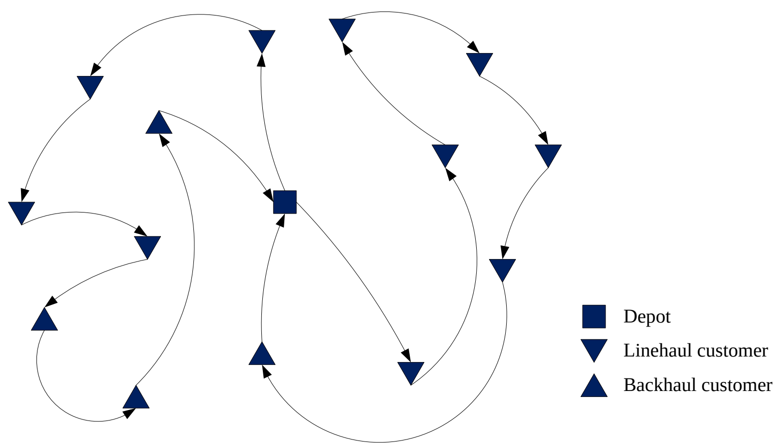

Figure 1. Illustration of independent routes for linehaul and backhaul customers

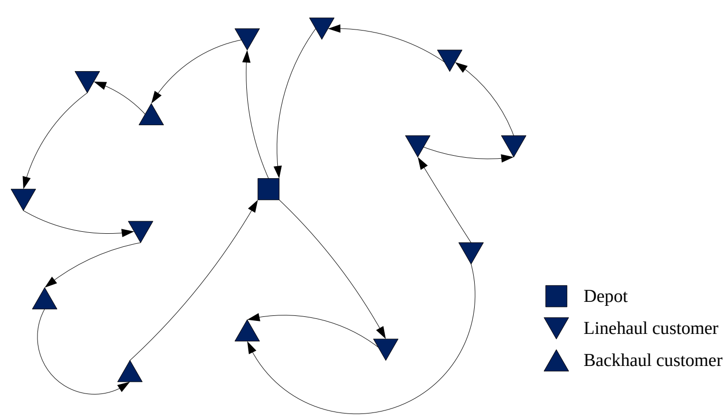

Figure 2. Illustration of routes containing linehaul and backhaul customers without precedence constraint

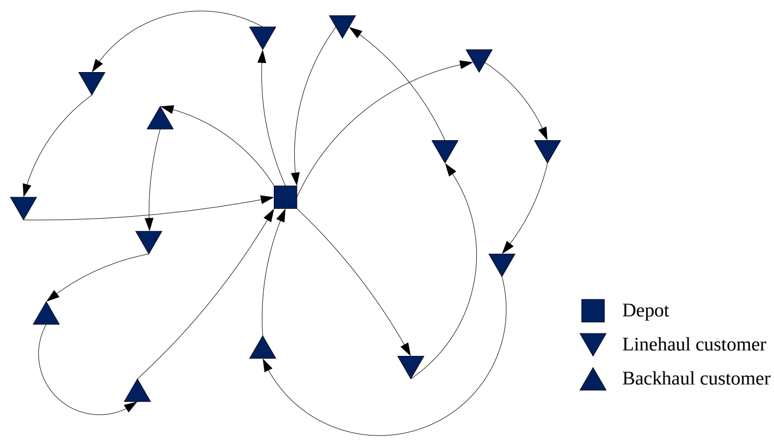

Figure 3. Illustration of routes containing linehaul and backhaul customers with precedence constraint

The VRPB has been discussed intensively. Some studies in the VRPB are Goetschalckx & Jacobs-Blecha (1989), Mingozzi et al. (1999), Toth & Vigo (1999), Crispim & Brandão (2005), Brandão (2006), Osman & Wassan (2002), Tavakkoli-Moghaddam et al. (2006), Wassan (2007), Gajpal & Abad (2009), and Zachariadis & Kiranoudis (2012). (Parragh, Doerner, & Hartl, 2008a) and (Parragh, Doerner, & Hartl, 2008b) have surveyed on the VRP including delivery and pickup services. A special survey dedicated for the VRPB is given by Koç & Laporte (2018).

As mentioned before, one of constraints imposed on the VRPB is that backhaul customers must be visited after linehaul customers on a particular route. In some discussions of the VRPB, there is another constraint included, i.e., no route consists of all backhaul customers. This constraint also states that every route consists of at least one linehaul customers. This constraint is mentioned by some author such as Brandão (2006), Ganesh & Narendran (2007), Wassan (2007), Gajpal & Abad (2009), Brandão (2016), Dominguez et al. (2016), and Wassan et al. (2017).

In this paper, the VRPB is combined with two other aspects: time window constraints and multiple trips. This variant is called multi-trip vehicle routing problem with backhauls and time windows (MTVRPBTW).

On the VRPTW, every customer has a time window characterized by a lower bound and an upper bound. The lower bound is defined the earliest time to start the service. If a vehicle arrives at a particular customer before the lower bound, then it must wait. However, there are two different definitions of the upper bound in literature. The first is defined the upper bound as the latest time to start the service (loading or unloading operations). In this case, if there is a service time (loading or unloading times) specified explicitly at a particular customer, then the time to finish the service at a particular customer may exceed the upper bound. Another definition of the upper bound appearing in Calvete et al. (2007) and Ding et al. (2012) is the latest time in which the service (loading or unloading operations) must end. In this case, a service time must be carried out between the lower bound and the upper bound. There are a lot of discussions on the VRPTW. One of reviews on the VRPTW has been provided by (El-Sherbeny, 2010).

The basic VRP assumes that a route is served exactly by one vehicle during the planning period. The VRPMT deals with the VRP where every vehicle is allowed to serve several routes along the planning period. The first study of the VRPMT is done by Fleischmann (1990). If the planning period is long and the vehicle capacity is small, compared to the basic VRP, the VRPMT has an advantage that the number of vehicles used is probably small. The VRPMT uses two different terms: route and tour as mentioned by Brandão & Mercer (1997). A route is defined as a sequence of customers visited by a vehicle starting and ending at the depot. A route is also named as a trip. A tour consists of several routed in which it is served by a vehicle. Cattaruzza et al. (2016) call a tour as a journey. Since introduced by Fleischmann (1990), the VRPMT has been investigated by widely. The first survey on the VRPMT is given by Şen & Bülbül (2008) where they focus on the pure VRPMT. A more comprehensive survey of the VRPMT was provided by Cattaruzza et al. (2016) in which some mathematical formulations and extensions of the VRPMT are described.

The discussions on the MTVRPBTW are very limited. Ong & Suprayogi (2011) discuss the MTVRPBTW and proposed a solution approach based on ant colony optimization (ACO). The MTVRPBTW addressed is a multi-objective case where the objectives involved are: minimization of the number of vehicles, minimization of the total tour duration time, and minimization of the range of tour duration time. In their problem, service times (loading and unloading times) do not depend on the quantity to be loaded or unloaded. Cahyadi et al. (2015) study the same problem to Ong & Suprayogi (2011) and proposed simulated annealing (SA). No mathematical model is given in Ong & Suprayogi (2011) and Cahyadi et al. (2015). Wassan et al. (2017) is one who discuss the MTVRPB. They develop a two-phase variable neighborhood search (VNS).

In this paper, the problem addressed is similar to the work of Ong & Suprayogi (2011) with an exception that the service time depends on the quantities loaded and unloaded both in customers and the depot. Two objectives are considered: minimization of the number of vehicles and minimization of the total tour duration time. The first objective has a higher priority than the second. The two objectives are tacked as a single objective by defining a weighted sum of the number of vehicles and the total tour duration time.

A mixed integer linier programming (MILP) model is formulated in this paper. The MILP model formulated is adapted from Azi et al. (2010). Due to the combinatorial nature of the problem, a metaheuristic approach is developed. In this paper, variable neighborhood descent (VND) is proposed. VND is a deterministic version of variable neighborhood search (VNS) introduced by Mladenović & Hansen (1997). VND is a metaheuristic approach having a way to change the neighborhood structures or local search operator once a local optimum is obtained during the search.

2. PROBLEM DEFITION AND MATHEMATICAL MODEL

2.1 Problem Definition

The MTVRPBTW studied in this paper consists of a set of customer nodes $\mathcal{C} = \mathcal{L} \cup \mathcal{B}$ where $\mathcal{L}$ denotes a set of l linehaul customer nodes and $\mathcal{B}$ is a set of b backhaul customer nodes. The VRP has a single depot and it is labeled as number 0. Let $\mathcal{N}$ be a set of nodes. Thus, $\mathcal{N} = \mathcal{C} \cup \{0\} = \mathcal{L} \cup \mathcal{B} \cup \{0\}$. Every travel time between between node $i \in \mathcal{N}$ and node $j \in \mathcal{N}$ is denoted by τ_{ij}. An unlimited number of homogeneous vehicles with capacity ϕ is available at the depot. Every linehaul customer $i \in \mathcal{L}$ is associated with a delivery demand indicated by δ_i. For every backhaul customers $i \in \mathcal{B}$ there is a pickup demand labeled by ρ_i. There is a time window associated at every customer $i \in \mathcal{C}$ where every time window for customer $i \in \mathcal{C}$ is characterized by an opening time ε_i and a closing time λ_i. A time window is also defined for the depot where ε_0 and λ_0 indicate the starting time of the planning period. The difference $\lambda_0 - \varepsilon_0$ actually represents the length of planning period. The unloading time at every customer depends on the quantities of delivery demand to be unloaded. Loading and unloading times per unit are given by γ and φ, respectively.

The objective is to minimize a weighted sum of the number of vehicles and the total tour duration time. The definition of tour duration time is taken from Savelsbergh (1992). In this case, every vehicle can depart from the depot on the first route later than time zero. A feasible solution of the VRP has to satisfy the following conditions: (1) all customers are serviced, (2) a tour may consist of one or several routes, (3) a route of every tour begins and ends at the depot, (4) a tour is served exactly by one vehicle, (5) a customer must be visited exactly once, (6) the time to start the service at every customer and the depot cannot be started prior the opening time, (7) a vehicle must wait at particular customer it arrives before the opening time, (8) the time to complete the service at customers and the depot cannot fall after the closing time, (9) a route may consists of all linehaul customers, all backhaul customers, or a mixture between linehaul and backhaul customers, and (10) the backhaul customers are visited after the linehaul customers if a particular route consists of a mixture between linehaul and backhaul customers.

2.2 Mathematical Model

In this section, a mixed integer linear programming (MILP) model for the MTVRPBTW is formulated. Define $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$ be a complete graph where $\mathcal{V}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. Set of nodes $\mathcal{V}$ consists of a depot numbered as 0. The linehaul customers and the backhaul customers are labeled by numbers $\{1, \cdots, l\}$ and numbers $\{l+1, \cdots, l+b\}$. A dummy depot denoted by $l+b+1$ is defined. Hence, set of nodes $\mathcal{V}$ is $\mathcal{V} = \{0,1, \cdots, l, l+1, \cdots, l+b, l+b+1\}$. Define $\varepsilon_{l+b+1} = \varepsilon_0$, $l_{l+b+1} = l_0$, $d_{l+b+1} = d_0 = 0$, $p_i = 0$ $(i = 1, \cdots, l)$, $d_i = 0$ $(i = l+1, \cdots, l+b+1)$, $\tau_{0j} = \tau_{l+b+1,j}$ $(j = 1, \cdots, l, l+1, \cdots, l+b)$, $\tau_{i0} = \tau_{i,l+b+1}$ $(i = 1, \cdots, l, l+1, \cdots, l+b)$, and $\tau_{0,l+b+1} = \tau_{l+b+1,0} = 0$. Based on Azi et al. (2010), a set of routes $\mathcal{R} = \{1, 2, \cdots, r'\}$ where r' is the number of routes is defined. The routes served by any vehicle are numbered in an increasing order, that is, a vehicle serves route $s \in \mathcal{R}$ after route $r \in \mathcal{R}$ if and only if $r < s$.

Notations used for the MILP model are as follows:

Sets:

- $\mathcal{V}$ set of nodes, $\mathcal{V} = \{0,1, \cdots, l, l+1, \cdots, l+b, l+b+1\}$
- $\mathcal{L}$ set of linehaul nodes, $\mathcal{L} = \{1, \cdots, l\}$
- $\mathcal{B}$ set of backhauls nodes, $\mathcal{B} = \{l+1, \cdots, l+b\}$
- $\mathcal{A}$ set of arcs
- $\mathcal{R}$ set of routes, $\mathcal{R} = \{1, 2, \cdots, r'\}$

Parameters:

- w_1 weight of the objective minimizing the number of vehicles
- w_2 weight of the objective minimizing the total tour duration time
- ϕ vehicle capacity
- τ_{ij} travel time on arc $(i, j) \in \mathcal{A}$
- d_i delivery quantity at node $i \in \mathcal{V}$
- p_i pickup quantity at node $i \in \mathcal{V}$
- e_i opening time for service at node $i \in \mathcal{V}$
- l_i closing time for service at node $i \in \mathcal{V}$
- γ loading time per unit
- φ unloading time per unit
- M big positive number

Decision variables:

- X_{ijr} binary variable where $X_{ijr} = 0$ if $(i, j) \in \mathcal{A}$ is served by route $r \in \mathcal{R}$ and $X_{ijr} = 0$ if otherwise (if $X_{0,n+1,r} = 1$, then route r is empty)
- Y_{ir} binary variable where $Y_{ir} = 1$ if node $i \in \mathcal{V}$ is served by route $r \in \mathcal{R}$ and $Y_{ir} = 0$ if otherwise
- Z_{rs} binary variable where $Z_{rs} = 1$ if route $r \in \mathcal{R}$ is followed immediately by route $s \in \mathcal{R}$ and $Z_{rs} = 0$ if otherwise
- T_{ir} arrival time at node $i \in \mathcal{V}$ on route $r \in \mathcal{R}$
- L_r loading time at the depot for route $r \in \mathcal{R}$.

U_r loading time at the depot for route $r \in \mathcal{R}$.
D_{ijr} delivery load on arc $(i,j) \in \mathcal{A}$ of route $r \in \mathcal{R}$
P_{ijr} pickup load on arc $(i,j) \in \mathcal{A}$ of route $r \in \mathcal{R}$
K number of vehicles deployed

The MILP model for the MTVRPBTW is formulated as follow:

Minimize

$$Z = w_1 K + w_2 \left(\sum_{r \in \mathcal{R}} \left(T_{n+1,r} + L_r - T_{0r} \right) \right) \tag{1}$$

subject to

$$\sum_{j \in \mathcal{V}\setminus\{0\}, j \neq i} X_{ijr} = Y_{ir}; i \in \mathcal{V}\setminus\{n+1\}, r \in \mathcal{R} \tag{2}$$
$$\sum_{r \in \mathcal{R}} Y_{ir} = 1; i \in \mathcal{V} \setminus\{0, l+b+1\} \tag{3}$$
$$\sum_{i \in \mathcal{V}\setminus\{l+b+1\}, i \neq h} X_{ihr} = \sum_{j \in V\setminus\{0\}, j \neq h} X_{hjr}\, ; h \in \mathcal{V}\setminus\{0, l+b+1\}, r \in \mathcal{R} \tag{4}$$
$$\sum_{i \in \mathcal{V}\setminus\{0\}} X_{0ir} = 1; r \in \mathcal{R} \tag{5}$$
$$\sum_{i \in \mathcal{V}\setminus\{l+b+1\}} X_{i,l+b+1,r} = 1; r \in \mathcal{R} \tag{6}$$
$$X_{ijr} = 0; i \in \mathcal{B}, j \in \mathcal{L}, r \in \mathcal{R} \tag{7}$$
$$T_{ir} + \varphi d_i + \gamma p_i + \tau_{ij} \leq T_{jr} + M\left(1 - X_{ijr}\right); i \in \mathcal{V}\setminus\{l+b+1\}, j \in \mathcal{V}\setminus\{0\}, i \neq j, r \in \mathcal{R} \tag{8}$$
$$T_{0r} + L_r + \tau_{0j} - M\left(1 - X_{0jr}\right) \leq T_{jr}; j \in \mathcal{V}\setminus\{0\}, r \in \mathcal{R} \tag{9}$$
$$T_{ir} \geq \varepsilon_i; i \in \mathcal{V}\setminus\{0, l+b+1\}, r \in \mathcal{R} \tag{10}$$
$$T_{ir} + \varphi d_i + \gamma p_i \leq \lambda_i; i \in \mathcal{V}\setminus\{0, l+b+1\}, r \in \mathcal{R} \tag{11}$$
$$T_{0r} \geq \varepsilon_0; r \in \mathcal{R} \tag{12}$$
$$T_{0r} + L_r \leq \lambda_0; r \in \mathcal{R} \tag{13}$$
$$T_{l+b+1,r} \geq \varepsilon_{l+b+1}; r \in \mathcal{R} \tag{14}$$
$$T_{l+b+1,r} \leq \lambda_{l+b+1}; r \in \mathcal{R} \tag{15}$$
$$L_r = \gamma \sum_{i \in V\setminus\{0,l+b+1\}} D_{0ir}\, ; r \in \mathcal{R} \tag{16}$$
$$U_r = \varphi \sum_{j \in V\setminus\{0,l+b+1\}} P_{n+1,j,r}\, ; r \in \mathcal{R} \tag{17}$$
$$T_{n+1,r} + U_r \leq T_{0s} + M(1 - Z_{rs}); r \in \mathcal{R}, s \in \mathcal{R}, r < s \tag{18}$$
$$T_{n+1,r} + U_r \geq T_{0s} - M(1 - Z_{rs}); r \in \mathcal{R}, s \in \mathcal{R}, r < s \tag{19}$$
$$\sum_{r \in \mathcal{R}} \sum_{s \in \mathcal{R}, r<s} Z_{rs} \geq r' - K \tag{20}$$
$$\sum_{s \in \mathcal{R}, s>r} Z_{rs} \leq 1; r \in \mathcal{R} \tag{21}$$
$$\sum_{r \in \mathcal{R}, s>r} Z_{rs} \leq 1; s \in \mathcal{R} \tag{22}$$
$$P_{0jr} = 0;\ j \in \mathcal{V}\setminus\{l+b+1\}, r \in \mathcal{R} \tag{23}$$
$$D_{i,n+1,r} = 0;\ i \in \mathcal{V}\setminus\{l+b+1\}, r \in \mathcal{R} \tag{24}$$
$$\sum_{i \in \mathcal{V}\setminus\{l+b+1\}, i \neq j} \sum_{r \in \mathcal{R}} P_{jir} - \sum_{i \in \mathcal{V}\setminus\{l+b+1\}, i \neq j} \sum_{r \in \mathcal{R}} P_{ijr} = p_j; j \in \mathcal{V}\setminus\{0, l+b+1\} \tag{25}$$
$$\sum_{i \in \mathcal{V}\setminus\{l+b+1\}, i \neq j} \sum_{r \in \mathcal{R}} D_{ijr} - \sum_{i \in \mathcal{V}\setminus\{l+b+1\}, i \neq j} \sum_{r \in \mathcal{R}} D_{jir} = d_j; j \in \mathcal{V}\setminus\{0, l+b+1\} \tag{26}$$
$$\sum_{i \in \mathcal{V}\setminus\{0,l+b+1\}} \sum_{r \in R} p_{i,l+b+1,r} = \sum_{i \in \mathcal{V}\setminus\{0,l+b+1\}} p_i \tag{27}$$
$$\sum_{i \in \mathcal{V}\setminus\{0,l+b+1\}} \sum_{r \in R} D_{0ir} = \sum_{i \in \mathcal{V}\setminus\{0,l+b+1\}} d_i \tag{28}$$
$$D_{ijr} + P_{ijr} \leq \phi X_{ijr};\ i \in \mathcal{V}\setminus\{l+b+1\}, j \in \mathcal{V}\setminus\{0\}, i \neq j, r \in \mathcal{R} \tag{29}$$
$$X_{ijr} \in \{0,1\}; i \in \mathcal{V}\setminus\{l+b+1\}, j \in \mathcal{V}\setminus\{0\}, i \neq j, r \in \mathcal{R} \tag{30}$$
$$Y_{ir} \in \{0,1\}; i \in \mathcal{V}, r \in \mathcal{R} \tag{31}$$
$$Z_{rs} \in \{0,1\}; r \in \mathcal{R}, s \in \mathcal{R}, r < s \tag{32}$$
$$T_{ir} \geq 0; i \in \mathcal{V}, r \in \mathcal{R} \tag{33}$$
$$D_{ijr} \geq 0; i \in \mathcal{V}, j \in \mathcal{V}, r \in \mathcal{R} \tag{34}$$
$$L_r \geq 0; r \in \mathcal{R} \tag{35}$$
$$U_r \geq 0; r \in \mathcal{R} \tag{36}$$

$K \geq 0$ (37)

The objective function to be minimized is represented in Eq. (1) where it is a weighted sum of the number of vehicles and the total tour duration time. Constraints (2) and (3) guarantee that each customer must be visited exactly once. Constraint (4) ensures that every customer must belong to the same route. Constraints (5) and (6) state that a route must start and end at the depot, respectively. Constraint (7) guarantees that a vehicle cannot travel from a backhaul customer to a linehaul customer. Constraint (8) and (9) define the time to start the service. Constraint (10) and (11) are time window constraints for every customer. Constraints (12)-(15) are time window constraints for the depot. The unloading and loading times at the depot are defined by Constraints (16) and (17), respectively. Constraints (18)-(22) are used to ensure the route sequence. Constraint (23) defines that the pickup load for starting a route at the depot is zero. Similarly, constraint (24) defines that the delivery load for completing a route at the depot also is zero. Constraints (25) and (26) are flow conservation constraints for pickup and delivery loads, respectively. Constraint (27) ensures that the total inflow to the depot is equal to the total pickup demand. Constraint (28) ensures that the total outflow from the depot is equal to the total delivery demand. The vehicle capacity constraint is given in constraint (29). Constraints (30)-(37) are constraints related to decision variable values.

3. SOLUTION APPROACHES

3.1. Solution Representation

This section provides a solution representation of the MTVRPBTW solved using the proposed VND. Let t $(t = 1, 2, \cdots)$, r $(r = 1, 2, \cdots)$, and k $(k = 0, 1, \cdots)$ be tour, route, and position indices, respectively. The solution θ obtained by the proposed approach includes the following variables to be determined.

NT	number of tours
$NR(t)$	number of routes of tour t
$NK(t,r)$	number of positions on routes r of tour t
$L(t,r,k)$	node (customer and depot) located at position k on routes r of tour t (In case of $k = 0$ and $k = NK(t,r)$, then $L(t,r,0) = 0$ and $L\big(t,r,NK(t,k)\big) = 0$ where they represent the depot)
$AT(t,r,k)$	arrival time of a vehicle at a node located at position k on routes r of tour t
$ST(t,r,k)$	time to start a service for a vehicle at node located at position k on routes r of tour t (if $k = 0$
$DT(t,r,k)$	departure time of a vehicle at a node located position k on routes r of tour (It also defines the time to finish a service)
$WT(t,r,k)$	waiting of vehicle at position k on routes r of tour t
$LT(t,r,k)$	loading time at a node located at position k on routes r of tour t
$UT(t,r,k)$	unloading time at a node located at position k on routes r of tour t
$DL(t,r)$	delivery load on routes r of tour t
$PL(t,r)$	pickup load on routes r of tour t
$VL(t,r,k)$	vehicle load leaving a node located at position k on routes r of tour t

As mentioned before, the MTVRPBTW addressed includes the objective of minimizing the total tour duration time. All information related to arrival times, times to start the service, and departure times are defined in the latest times. Therefore, $AT(t,r,k)$, $ST(t,r,k)$, and $DT(t,r,k)$ can be viewed as the latest arrival time, the latest time to start the service, and the latest departure time, respectively.

Because a vehicle performs exactly a tour, then the number of vehicles is identical to the number of tours expressed in the following equation:

$$NV = NT \tag{38}$$

The tour duration time of tour t is given by:

$$TDT(t) = DT\left(t, NR(t), NK\big(t, NR(t)\big)\right) - AT(t,1,0) \tag{39}$$

The total tour duration time is defined as follows:

$$TTDT = \sum_{t=1}^{NT} TDT(t) \tag{40}$$

A weighted sum of the number of vehicles and the total tour duration time is given by:

$$Z = \omega_1 \times NV + \omega_2 \times TTDT \tag{41}$$

3.2 Initial Solution

The proposed VND requires an initial solution. The initial solution is obtained by applying the insertion algorithm where it belongs to a family of construction heuristic algorithms.

Steps of the sequential insertion algorithm proposed in this paper are given in **Algorithm 1**. It begins by constructing a tour having a route with empty customer starting and ending at the depot. Then, a customer from unassigned customers is selected based on a given rule or criterion. The customer chosen is called the seed customer. There are several rules can be used to choose the seed customer. Solomon (1987) proposed two criteria for the VRPTW: (1) the unassigned customer with the farthest distance from the depot and (2) the unassigned customer with the earliest closing time. Suprayogi (2003) and Suprayogi & Imawati (2005) used four rules for the VRPMTTW, i.e., the unassigned customer with the earliest opening time, the unassigned customer with the earliest closing time, the unassigned customer with the shortest length of time window, and the unassigned customer with the farthest distance from the depot. In this proposed solution method, the rule applied is the unassigned customer with the earliest opening time.

After the seed customer is selected, the next step is to attempt for inserting unassigned customers to the current route. If there are feasible insertions regarding to capacity and time window constraints, then an insertion giving the minimum tour completion time is selected. The procedure is repeated until there is no feasible insertion. If it happens, then a temporary route with empty customer starting and ending at the depot for the current tour is constructed. Every unassigned customer is tried to be inserted to this empty route. If there are feasible insertions, then an unassigned customer with the minimum tour completion time is chosen and the procedure proceeds to the similar steps to insert unassigned customers to the current route. If there is no feasible insertion, then the temporary route of the current tour is removed and a new tour with an empty route of the new tour is constructed. Again, the seed customer is selected. The procedure continues until all customers are assigned.

After a complete solution (tour and schedule plan) is obtained, in order to determine the latest arrival times, the latest time to start the service, and the latest departure times, a backward procedure is applied. The objective function value is determined.

Algorithm 1

1. Set $t = 1$ and $NT = 1$.
2. Set $r = 1$ and $NR(t) = 1$.
3. Select $i \in UC$ as a seed customer. Let i^* be the seed customer.
4. Set $L(t,r,0) = 0$, $L(t,r,1) = i^*$, $NK(t,r) = 2$, $L(t,r,NK(t,r)) = 0$, and $UC = UC \backslash i^*$.
5. If $UC = \emptyset$, go to step 14.
6. Try to select $i \in UC$ and insert to positions from $k = 1$ to $k = NK(t,r)$.
7. If there are feasible insertions, go to step 8. Otherwise, go to step 9.
8. Select customer i^* and insertion position k^* giving the minimum completion time. Set $L(t,r,k^*) = i^*$, and $NK(t,r) = NK(t,r) + 1$. Update $L(t,r,k)$ for other k and set $UC = UC \backslash i^*$. Back to step 5.
9. Set $r = r + 1$ and $NR(t) = NR(t) + 1$.
10. Try to set $L(t,r,1) = i$ for $i \in UC$.
11. If there are feasible customers, then go to step 12. Otherwise, go to step 13.
12. Select customer i^* giving the minimum completion time. Set $L(t,r,1) = i^*$, $NK(t,r) = NK(t,r) + 1$, and set $UC = UC \backslash i^*$. Back to step 5.
13. Set $r = r - 1$ and $NR(t) = NR(t) - 1$. Set $t = t + 1$. Back to step 2.
14. Perform a backward procedure.
15. Calculate an objective function value.

3.3 Neighborhood Structures

There are five neighborhood structures used in the proposed VND: inter-tour relocation, intra-tour relocation, inter-tour exchange, intra-tour exchange, and cross. Every neighborhood structure is briefly described as follows.

Inter-tour relocation. This neighborhood structure relocates one customer from a position tour to another tour. This neighborhood structure can be applied outside existing routes. This neighborhood structure is capable to reduce routes and tours.

Intra-tour relocation. This neighborhood structure relocates one customer from a tour to another route on the same tour.

Inter-tour exchange. This neighborhood structure exchanges one customer from a tour with one customer on another tour.

Intra-tour exchange. This neighborhood structure exchanges one customer from a tour with one customer on the same tour.

Crossover. This neighborhood structure divides two tours into two parts and exchange the tails of the respective tours. This neighborhood structure is also capable to reduce routes and tours.

3.4 Variable Neighborhood Descent

The proposed VND is given in **Algorithm 2**. VND starts by generating the initial solution using the sequential insertion algorithm given in step 1. The improvement phase is given in the rest steps. The best improvement strategy is used in every neighborhood structure. VND terminates if there is no improvement for the current best solution.

Algorithm 2

1. Generate an initial solution θ^0 using the sequential insertion algorithm.
2. $\theta' = \theta^0$ and $f(\theta') = f(\theta^0)$
3. $\theta^* = \theta^0$ and $f(\theta^*) = f(\theta^0)$
4. $BL = f(\theta')$
5. $BG = f(\theta^*)$.
6. $\theta'' = \theta'$
7. $h = 1$
8. $LocalImproved = False$
9. **Do**
10.   Set $Improved = False$
11.   **Do**
12.   Generate θ from θ'' using a move of neighborhood structure h.
13.   **If** θ is feasible **Then**
14.   **If** $f(\theta) < BL$ **Then**
15. $BL = f(\theta)$
16. $\theta' = \theta$
17. $Improved = True$
18.   **End If**
19.   **End If**
20.   **Loop Until** all moves of neighborhood structure h are explored.
21.   **Loop Until** $Improved = False$
22.   **If** $BL < BG$ **Then**
23. $\theta^* = \theta'$
24. $LocalImprove = True$
25.   **End If**
26.   **If** $LocalImprove = True$ **Then**
27. $h = 1$
28.   **Else**
29. $h \leftarrow h + 1$
30.   **End If**
31. **Loop Until** $h > h_{max}$

4. NUMERICAL EXPERIMENTS

4.1 Instances

The proposed VND is coded using Visual Basic 6 and run on a PC with Intel® Core™ i7-3770s CPU @ 3.10GHz processor, RAM 8GB memory, and Windows 8.1 64-bit operating system.

In order to assess the proposed VND, a set of instances are generated. The instances are classified based on the following characteristics: number of customers, vehicle capacity, customer location, and composition of narrow and wide time windows.

According to the number of customers, there are two groups of the instances, i.e., instances with 20 and 50 customers. Based on the customer location, instances are divided into three groups inspired from (Solomon, 1987), i.e., clustered, randomized, and mixed instances. For every clustered instance, there are five cluster locations where coordinates (x, y) of customer locations for clusters 1 to 5 are generated randomly as follows: cluster 1 (0, 0) to (0, 50), cluster 2 (550, 0) to (600, 50), cluster 3 (0, 550) to (50, 600), cluster 4 (550, 550) to (600, 600), and cluster 5 (275, 275) to (325, 325). For the randomized instances, coordinates for customer locations are generated randomly ranging from (0, 0) to (600, 600). In every mixed instance, the composition of clustered and randomized locations is 50%-50%. The instances are divided into three groups depending on the composition of customers with narrow and wide time windows, i.e., group 1 (narrow = 100%, wide = 0%), group 2 (narrow = 50%, wide = 50%), and group 3 (narrow = 0% and wide = 100%). The narrow time windows are set randomly from 1 to 30, while the wide time windows are generated randomly ranging from 100 to 600.

Table 1 shows characteristics for the instances. For every instance, the coordinate of depot location is (300, 300). Additional data required are: loading time per unit γ = 1, unloading time per unit γ = 1, cost per vehicle c_1 = 10000, and cost per time unit c_2 = 1. Every distance between two nodes is an Euclidean distance. To convert the distance data into the travel time data, a conversion factor is given where it can be interpreted as the vehicle speed. The conversion factor σ is set as 10. Vehicle capacity ϕ is set as 20.

Table 1. Instances

Number of customers	Vehicle capacity	Customer location	Composition of narrow (N) and wide (W) time windows		
			N = 100% W = 0%	N = 50% W = 50%	N = 0% W = 100%
20	20	Clustered	C-020-1	C-020-1	C-020-3
		Randomized	R-020-1	R-020-1	R-020-3
		Mixed	M-020-1	M-020-1	M-020-3
50	20	Clustered	C-050-1	C-050-2	C-050-3
		Randomized	R-050-1	R-050-2	R-050-3
		Mixed	M-050-1	M-050-2	M-050-3

4.2 Results

VND depends on the order of neighborhood structures applied. Based on the order of neighborhood structures, there are two schemes of VND developed, called VND-1 and VND-2. VND has the following order: (1) inter-tour relocation, (2) crossver, (3) inter-tour exchange, (4) intra-tour relocation, and (5) intra-tour exchange. VND-2 applies the following order of neighborhood structures: (1) intra-tour exchange, (2) intra-tour relocation, (3) inter-tour exchange, (4) crossover, and (5) inter-tour relocation. Both VND-1 and VND-2 use the earliest closing time to select the seed customer in the sequential insertion algorithm.

Table 2 show computional results. Columns OFV and CT indicate the objective function value and the computation time, respecively. Based on the paired t-test, there is no significant difference between VND-1 and VND-2 in terms of the objective function. But, in terms of the computation time, VND-1 is faster than VND-2.

Table 2. Results

Instance	VND-1		VND-2	
	OFV	CT (Sec.)	OFV	CT (Sec.)
C-020-1	41477.44	4.01	41579.77	2.26
C-020-2	41071.60	3.18	31038.55	5.84
C-020-3	30810.03	2.22	30719.67	4.38
R-020-1	51410.35	3.32	51410.35	4.17
R-020-2	40931.00	3.51	40917.94	2.92
R-020-3	20651.70	6.28	20739.19	2.01
M-020-1	41406.45	3.15	41415.05	3.37
M-020-2	40957.97	3.69	30970.46	5.20
M-020-3	30763.24	2.36	30751.34	3.40
C-050-1	92913.48	50.41	92861.50	90.06
C-050-2	72408.94	35.94	82302.94	56.82
C-050-3	41335.20	49.09	41409.87	32.62
R-050-1	72578.74	47.14	82732.60	84.07
R-050-2	72075.35	46.18	62120.14	66.24
R-050-3	41334.43	43.68	41357.09	39.83
M-050-1	82947.04	41.80	92724.71	97.48
M-050-2	72444.12	32.07	92072.20	79.15
M-050-3	41311.98	46.15	41300.32	63.09

5. CONCUSIONS

This paper has presented an extension of the basic vehicle routing problem (VRP) called the multi-trip vehicle routing problem with backhauls and time windows (MTVRPBTW). Service-dependent times are considered in the problem. The MTVRPBTW presented dealing with the problem of determining a solution minimizing a weighted sum of the number of vehicles and the total tour duration time. A mixed integer linear programming (MILP) model has been formulated to represent the problem. A solution approach based on variable neighborhood search (VND) has been proposed.

ACKNOWLEDGMENT

This work is supported by the *Program Peningkatan Penelitian, Pengabdian kepada Masyarakat, dan Inovasi* (P3MI) of Institut Teknologi Bandung.

6. REFERENCES

Azi, N., Gendreau, M., & Potvin, J.-Y. (2010). An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, *202*(3), 756–763. https://doi.org/10.1016/j.ejor.2009.06.034

Brandão, J, & Mercer, A. (1997). A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, *100*(1), 180–191. https://doi.org/10.1016/S0377-2217(97)00010-6

Brandão, José. (2006). A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, *173*(2), 540–555. https://doi.org/10.1016/j.ejor.2005.01.042

Brandão, José. (2016). A deterministic iterated local search algorithm for the vehicle routing problem with backhauls. *TOP*, *24*(2), 445–465. https://doi.org/10.1007/s11750-015-0404-x

Cahyadi, F., Ong, J. O., & Kosasih, J. S. (2015). Perancangan algoritma simulated annealing untuk rute kendaraan yang mempertimbangkan backhaul, rute majemuk, dan time window. *Jurnal Telematika*, *7*(1).

Calvete, H. I., Galé, C., Oliveros, M.-J., & Sánchez-Valverde, B. (2007). A goal programming approach to vehicle routing problems with soft time windows. *European Journal of Operational Research*, *177*(3), 1720–1733. https://doi.org/10.1016/j.ejor.2005.10.010

Cattaruzza, D., Absi, N., & Feillet, D. (2016). Vehicle routing problems with multiple trips. *4OR*, *14*(3), 223–259. https://doi.org/10.1007/s10288-016-0306-2

Crispim, J, & Brandão, J. (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. *Journal of the Operational Research Society*, *56*(11), 1296–1302. https://doi.org/10.1057/palgrave.jors.2601935

Crispim, José, & Brandão, J. (2001). Reactive tabu search and variable neighbourhood descent applied to the vehicle routing problem with backhauls. *Proceedings of the 4th Metaheuristics International Conference (MIC'2001)*, 631–636.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*(1), 80–91. https://doi.org/10.1287/mnsc.6.1.80

Ding, Q., Hu, X., Sun, L., & Wang, Y. (2012). An improved ant colony optimization and its application to vehicle routing problem with time windows. *Neurocomputing*, *98*, 101–107. https://doi.org/10.1016/j.neucom.2011.09.040

Dominguez, O., Guimarans, D., Juan, A. A., & de la Nuez, I. (2016). A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research*, *255*(2), 442–462. https://doi.org/10.1016/j.ejor.2016.05.002

El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, *22*(3), 123–131. https://doi.org/10.1016/j.jksus.2010.03.002

Fleischmann, B. (1990). *The vehicle routing problem with multiple use of the vehicles*. Working paper, Fachbereich Wirtschaftswissenschaften. Universität Hamburg.

Gajpal, Y., & Abad, P. L. (2009). Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. *European Journal of Operational Research*, *196*(1), 102–117. https://doi.org/10.1016/j.ejor.2008.02.025

Ganesh, K., & Narendran, T. T. (2007). CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up. *European Journal of Operational Research*, *178*(3), 699–717. https://doi.org/10.1016/j.ejor.2006.01.037

Goetschalckx, M., & Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls. *European Journal of Operational Research*, *42*(1), 39–51. https://doi.org/10.1016/0377-2217(89)90057-X

Irnich, S., Schneider, M., & Vigo, D. (2014). Four Variants of the Vehicle Routing Problem. In P. Toth & D. Vigo (Eds.), *Vehicle Routing: Problems, Methods, and Applications* (Second Edi, pp. 241–271).

Koç, Ç., & Laporte, G. (2018). Vehicle routing with backhauls: Review and research perspectives. *Computers & Operations Research*, *91*, 79–91. https://doi.org/10.1016/j.cor.2017.11.003

Koch, H., Schlögell, M., & Bortfeldt, A. (2017). *A hybrid algorithm for the vehicle routing problem with three-dimensional loading constraints and mixed backhauls*. Working Paper No. 11/2017. Faculty of Economics and Management, Universität Magdeburg.

Mingozzi, A., Giorgi, S., & Baldacci, R. (1999). An exact method for the vehicle routing problem with backhauls. *Transportation Science*, *33*(3), 315–329. https://doi.org/10.1287/trsc.33.3.315

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2

Ong, J. O., & Suprayogi. (2011). Vehicle routing problem with backhaul, multiple trips and time window. *Jurnal Teknik Industri*, *13*(1), 1–10. https://doi.org/10.9744/jti.13.1.1-10

Osman, I. H., & Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls. *Journal of Scheduling*, *5*(4), 263–285. https://doi.org/10.1002/jos.122

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008a). A survey on pickup and delivery problems - Part I: Transportation between customers and depot. *Journal Für Betriebswirtschaft*, *58*(1), 21–51. https://doi.org/10.1007/s11301-008-0033-7

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008b). A survey on pickup and delivery problems - Part II: Transportation between pickup and delivery locations. *Journal Für Betriebswirtschaft*, *58*(2), 81–117. https://doi.org/10.1007/s11301-008-0036-4

Reimann, M., & Ulrich, H. (2006). Comparing backhauling strategies in vehicle routing using ant colony optimization. *Central European Journal of Operations Research*, *14*(2), 105–123. https://doi.org/10.1007/s10100-006-0163-8

Ropke, S., & Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, *171*(3), 750–775. https://doi.org/10.1016/j.ejor.2004.09.004

Savelsbergh, M. (1992). The vehicle routing problem with time windows: minimizing route duration. *ORSA Journal on Computing*, *4*(2), 146–154. https://doi.org/10.1287/ijoc.4.2.146

Şen, A., & Bülbül, K. (2008). A survey on multi trip vehicle routing problem. *International Logistics & Supply Chain Congress*, 401–407.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, *35*(2), 254–265. https://doi.org/10.1287/opre.35.2.254

Suprayogi. (2003). Algoritma sequential insertion untuk memecahkan vehicle routing problem dengan multiple trips dan time windows. *Jurnal Teknik Dan Manajemen Industri*, *23*(3), 30–46.

Suprayogi, & Imawati, D. (2005). 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), 41–54.

Tavakkoli-Moghaddam, R., Saremi, A. R., & Ziaee, M. S. (2006). A memetic algorithm for a vehicle routing problem with backhauls. *Applied Mathematics and Computation*, *181*(2), 1049–1060. https://doi.org/10.1016/j.amc.2006.01.059

Toth, P., & Vigo, D. (1999). A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *European Journal of Operational Research*, *113*(3), 528–543. https://doi.org/10.1016/S0377-2217(98)00086-1

Wassan, N. (2007). Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. *Journal of the Operational Research Society*, *58*(12), 1630–1641. https://doi.org/10.1057/palgrave.jors.2602313

Wassan, Naveed, Wassan, N., Nagy, G., & Salhi, S. (2017). The multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search. *Computers & Operations Research*, *78*, 454–467. https://doi.org/10.1016/j.cor.2015.12.017

Yazgitutuncu, G., Carreto, C. A. C., & Baker, B. M. (2009). A visual interactive approach to classical and mixed vehicle routing problems with backhauls☆. *Omega*, *37*(1), 138–154. https://doi.org/10.1016/j.omega.2006.11.001

Zachariadis, E. E., & Kiranoudis, C. T. (2012). An effective local search approach for the vehicle routing problem with backhauls. *Expert Systems with Applications*, *39*(3), 3174–3184. https://doi.org/10.1016/j.eswa.2011.09.004