Technical Report

Vehicle Routing Problem with Simultaneous Pickup and Delivery: Mixed Integer Programming Formulations and Comparative Analyses

Ismail Karaoglan\textsuperscript{a}, Fulya Altiparmak\textsuperscript{b}, Imdat Kara\textsuperscript{c}, Berna Dengiz\textsuperscript{c}

\textsuperscript{a}Department of Industrial Engineering, Selcuk University, Turkey
\textsuperscript{b}Department of Industrial Engineering, Gazi University, Turkey
\textsuperscript{c}Department of Industrial Engineering, Baskent University, Turkey

Abstract

This paper addresses the vehicle routing problem with simultaneous pickup and delivery (VRPSPD) which is the problem of optimally integrating goods deliver and pickup when there are no precedence restrictions on the order in which the operations must be performed. We review mixed integer programming formulations in the literature for VRPSPD and propose two new formulations, PM1 and PM2, for the problem. While PM1 is a two-index node-based formulation, PM2 is a two-index flow-based formulation and both formulations have $O(n^2)$ binary variables and $O(n^2)$ constraints. Additionally, we prove that PM2 produce better lower bounds than PM1. To identify the efficient formulation, we conduct an experimental study and compare existing and proposed formulations in terms of linear programming relaxations gap, optimality gap, the number of optimal solutions obtained within two hours of CPU and average CPU time on two sets of test problems adapted from the literature. The computational results indicate that PM2 outperforms the other formulations in terms of the above performance measures.

Keywords: Vehicle routing problem, Simultaneous pickup and delivery, Integer programming.
1. Introduction

Vehicle routing problems (VRP), introduced by Dantzig and Ramser (1959), are important and well-known combinatorial optimization problems occurring in many transport logistics and distribution systems of considerable economic significance. VRP can be defined as the problem of designing optimal delivery or pickup routes from one depot to a number of geographically dispersed customers under the side conditions. The objective is to find a set of routes to service all customers while minimizing the total travel distance/time or the distribution cost. There are several variants of VRP depending on tasks performed and under some restrictions in the literature. The comprehensive reviews on VRP, its variants, formulations, and solution methods can be found in Laporte and Osman (1995), Laporte (1997) and Toth and Vigo (2002).

One of the variants of VRP is the vehicle routing problem with simultaneous pickup and delivery (VRPSPD). In VRPSPD, the vehicles are not only required to deliver goods to customer, but also pickup goods at customer locations. A general assumption in VRPSPD is that all delivered goods must be originated from the depot, all pickup goods must be transported back to the depot. Delivery and pickup goods must be met simultaneously when each customer is visited only once by a vehicle and unloading is carried out before loading at the customers (Chen and Wu, 2006).

The application of VRPSPD is frequently encountered in the distribution system of grocery store chains. Each grocery store may have both a delivery (e.g., fresh food or soft drink) and pickup (e.g., outdated items or empty bottles) demand and is serviced with a single stop (Chen and Wu, 2006). Reverse logistics is also another application area for VRPSPD as companies become interested in gaining control over the whole lifecycle of their products. For example, in some countries legislation forces companies to take responsibility for their products during lifetime, especially when environmental issues are involved (Dethloff, 2001, Montane and Galvao, 2006).

VRPSPD is an NP-hard problem (Salhi and Nagy, 1999), because it can be considered as VRP, which is well-known NP-hard problem, when only delivery goods or pickup goods are considered. Thus, after VRPSPD has been introduced by Min (1989), the research on this problem has been mainly focused on heuristic and meta-heuristic approaches (see, for example, Salhi and Nagy, 1999; Dethloff, 2001; Tang and Galvao, 2002; Nagy and Salhi, 2005; Chen and Wu, 2006; Montane ve Galvao, 2006; Bianchessi and Righini, 2007; Ganesh
and Narendran, 2007; Jin Ai and Kachitvicyanukul, 2009; Gajpal and Abad, 2009). To the best of our knowledge, in the literature, there is only one study in which an exact algorithm for the problem has been developed based on branch and price approach by Dell’Amico et al., (2006). Meanwhile, mixed integer programming (MIP) formulations for VRPSPD have been presented by Min (1989), Dethloff (2001), Nagy and Salhi (2005), Montane and Galvao (2006), Dell’Amico et al. (2006), Jin Ai and Kachitvicyanukul (2009). We refer the interested readers to the papers of Berbeglia et al. (2007) and Parragh et al. (2008) for extensive review about this problem and its variations.

Despite the increasing interest in heuristic approaches for VRPSPD, developing efficient mathematical formulations for VRPSPD has received much less attention from researchers. However, efficient mathematical formulations for the problem may allow us to solve small or moderate-size problem instance by using any commercial package or to develop new exact and/or mathematical model-based heuristic algorithms for the problem. This is the main motivation of our paper. In this paper, firstly we review existing formulations for VRPSPD in the literature and propose two new formulations, PM1 and PM2, for the problem. These new formulations have been derived from the formulations developed by Kara (2008) for VRP. While PM1 is a two-index node-based formulation, PM2 is a two-index flow-based formulation and both formulations have $O(n^2)$ binary variables and $O(n^2)$ constraints. In PM1, capacity and subtour elimination constraints are based on Miller-Tucker-Zemlin (MTZ) constraints (Miller et al. 1960; Kara et al., 2004; Kara, 2008) and PM2 is obtained by introducing new bounding constraints to Dell’Amico et al.’s (2006) formulation. Additionally, we theoretically compare the lower bounds obtained by PM2 and PM1, and it is proven that PM2 produce better lower bounds than PM1. To identify an efficient one among the existing and proposed formulations, we conduct an experimental study by solving test problems adapted from the literature. In the experimental study, we compare the formulations in terms of linear programming relaxations gap, optimality gap, the number of optimal solutions obtained within two hours of CPU and average CPU time. Computational results over the test instances show that the proposed second formulation, PM2, outperforms other formulations in terms of all performance measures. The contribution of this paper is threefold; 1) reviewing the current formulations for VRPSPD, 2) introducing two new formulations and 3) evaluating the performance of the proposed formulations on several test problems in literature.

The rest of this paper is organized as follows. Problem description is given in the next section. The existing formulations are reviewed in the Section 3 and the proposed
formulations are described in the Section 4. Computational results are reported in the Section 5 and finally, conclusion and suggestion for future researches are given in the last section.

2. Problem Description and Notations

VRPSPD can be defined formally as follows: Let $G = (N, A)$ be a complete directed network where $N = N_0 \cup \{0\}$ is the set of vertices in which $N_0$ represents the customers and “0” represents the depot, respectively. $A = \{(i, j): i, j \in N\}$ is the set of arcs, and to each arc $(i, j)$ is associated a nonnegative cost (distance) $c_{ij}$ for each $i, j \in N$. In the depot there are identical vehicles with a capacity $Q$ and there is no restriction on number of vehicles. In VRPSPD, each customer $i \in N_0$ requires a given quantity to be delivered ($d_i$) and picked-up ($p_i$). VRPSPD consists of finding a set of routes such that:

- Each route starts and ends at the depot,
- Each customer is visited exactly once by exactly one vehicle,
- The total vehicle load in any arc does not exceed the capacity of the vehicle assigned to it,
- The total routing cost is minimized.

The following are the common sets and notations used in the formulations: (Specific notations are introduced as needed.)

$N$; set of all nodes $(0, \ldots, n)$,
$N_0$; set of all customers $(1, \ldots, n)$,
$K$; set of vehicles $(1, \ldots, k)$
$c_{ij}$; cost (distance) of an arc $(i, j)$ $(i, j \in N)$
$d_i$; delivery demand of customer $i$ $(i \in N_0)$,
$p_i$; pickup demand of customer $i$ $(i \in N_0)$,
$Q$; vehicle capacity,
$M$; large positive number,

3. Formulations for VRPSPD in Literature

The formulations for VRP may be classified in terms of number of constraints: a) \textit{exponential-size formulation}, the number of constraints grow exponentially based on the
number of nodes of the graph, b) polynomial-size formulation, the number of constrains grow polynomially based on the number of nodes of the graph. Since exponential-size formulations cannot be used directly solve instances of small or moderate-sizes by any commercial software, we focus on the polynomial-size formulations in this paper.

It is well known that we need to additional variables to obtain polynomial size formulations for VRP (i.e. modeling of subtour elimination and capacity constraints). Polynomial-size formulations can also be classified into two subgroups based on the definition of additional variables: i) node-based formulation, the additional variables are relative due to the nodes of the graph, ii) flow-based formulation, the additional variables are relative due to the arcs of the graph (Kara, 2008). Thus, we introduce the existing and new formulations for VRPSPD considering this classification.

As far as we are aware, formulations for VRPSPD have been proposed by Min (1989), Dethloff (2001), Nagy and Salhi (2005), Montane and Galvao (2006), Dell’Amico et al., (2006), and Jin Ai and Kachitvicyanukul, (2009). While Dell’Amico et al.’s (2006) and Montane and Galvao’s (2006) formulations are flow-based formulations, others are node-based formulations. Meantime, only Min’s (1989) formulation among others is exponentially-size formulation.

In this section, we briefly review three formulations among five polynomial-size formulations because our preliminary study on the formulations has revealed that there are some deficiencies in Nagy and Salhi’s (2005) and Jin Ai and Kachitvicyanukul’s (2009) formulations. Therefore, we do not consider these two formulations in our further evaluations.

3.1. The Node-Based Formulation: Dethloff’s (2001) Formulation

The node-based formulation for VRPSPD has been proposed by Dethloff (2001) (referred to as DM in the sequel). In this model, the author has used MTZ subtour elimination constraints in his three-index formulation. The decision variables used in DM are as follows:

Decision Variables:
\[
x_{ijk} = \begin{cases} 
1 & \text{if vehicle } k \text{ travels directly from node } i \text{ to node } j \ (\forall i, j \in N; \forall k \in K) \\
0 & \text{otherwise} 
\end{cases}
\]

Additional Variables:
\[
L'_k; \text{ load of vehicle } k \text{ when leaving the depot } (\forall k \in K) \\
L_i; \text{ load of vehicle after having serviced customer } i \ (\forall i \in N_o) \\
U_i; \text{ variable used to prohibit subtours } (\forall i \in N_o),
\]
The formulation, DM, is as follows:

$$\text{min} \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} c_{ij} x_{ijk} \quad (1)$$

s.t. \[ \sum_{i \in N} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in N_0 \quad (2) \]

\[ \sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{jik} \quad \forall i \in N, \forall k \in K \quad (3) \]

\[ \sum_{j \in N_0} x_{0jk} \leq 1 \quad \forall k \in K \quad (4) \]

\[ U_i - U_j + |N| \sum_{k \in K} x_{ijk} \leq |N| - 1 \quad \forall i, j \in N_0, i \neq j \quad (5) \]

\[ 1 \leq U_i \leq |N| \quad \forall i \in N_0 \quad (6) \]

\[ L'_k = \sum_{i \in N} \sum_{j \in N_0} d_j x_{ijk} \quad \forall k \in K \quad (7) \]

\[ L_i \geq L'_k - d_i + p_i - M (1 - x_{0ik}) \quad \forall i \in N_0, \forall k \in K \quad (8) \]

\[ L_j \geq L_i - d_j + p_j - M \left(1 - \sum_{k \in K} x_{ijk}\right) \quad \forall i, j \in N_0, i \neq j \quad (9) \]

\[ 0 \leq L'_k \leq Q \quad \forall k \in K \quad (10) \]

\[ 0 \leq L_j \leq Q \quad \forall j \in N_0 \quad (11) \]

\[ x_{ijk} \in \{0,1\} \quad \forall i, j \in N, \forall k \in K \quad (12) \]

In this formulation, the objective function (1) is to minimize the total cost (distance). Constraints (2) ensure that each customer must be serviced exactly once. Constraints (3) imply that if a vehicle arrives at a customer, then the same vehicle must also leave. Constraint (4) guarantees that a vehicle can be used at most one route. While constraint (5) eliminates subtours and constraint (6) bounds additional variables, constraint (7) denotes the initial load of vehicles. Constraints (8) and (9) describe the fluctuating load on vehicle after first customer and any customers, respectively. Constraints (10) and (11) state that the vehicle loads do not exceed the vehicle capacity. Finally, constraints (12) impose binary conditions on the variables. DM has $O(n^2k)$ binary variables, $O(n)$ additional variables and $O(n^2)$ constraints.
It is worthy to note that although Dethloff (2001) has developed four heuristic procedures for the problem and compared them in terms of solution quality and solution time in his paper, any computational result for DM has not been presented in that paper.

3.2. Flow-Based Formulations

There are two different flow-based formulations for VRPSPD in the literature. These formulations have been developed by Montane and Galvao (2006) and Dell’Amico et al. (2006).

3.2.1. Montane and Galvao’s (2006) Formulation

Montane and Galvao’s formulation (2006), abbreviated as MGM, is a three-index formulation. The decision variables in MGM are as follows ($x_{ijk}$ is defined as previously):

Additional Variables:

$y_{ij}$; demand picked-up in customers routed up to node $i$ (including node $i$) and transported in arc $(i, j)$ ($\forall i, j \in N$)

$z_{ij}$; demand to be delivered to customers routed after node $i$ and transported in arc $(i, j)$ ($\forall i, j \in N$)

The formulation, MGM, is as follows:

$$\min (1) \text{ subject to (2)--(4), (12) and}$$

$$\sum_{i \in N} y_{ji} - \sum_{i \in N} y_{ij} = p_j \quad \forall j \in N_0 \quad (13)$$

$$\sum_{i \in N} z_{ij} - \sum_{i \in N} z_{ji} = d_j \quad \forall j \in N_0 \quad (14)$$

$$y_{ij} + z_{ij} \leq Q \sum_{k \in K} x_{ijk} \quad \forall i, j \in N, i \neq j \quad (15)$$

$$y_{ij}, z_{ij} \geq 0 \quad \forall i, j \in N \quad (16)$$

In this model, constraints (13) and (14) guaranty the conservation of flows for pickup and delivery demands, respectively. Constraint (15) ensures that the total load on vehicle does not exceed the vehicle capacity, and finally, constraint (16) imposes non-negativity conditions on
the variables. In MGM, there are $O(n^2k)$ binary variables, $O(n^2)$ additional variables and $O(n^2)$ constraints.

Montane and Galvao (2006) have also developed a heuristic algorithm based on tabu search algorithm for VRPSPD in their paper. They have evaluated the performance of the proposed algorithm against not only other heuristic approaches in the literature for the problem but also upper bounds obtained with formulation by solving it CPLEX over 86 test instances. It is important to note that Montane and Galvao’s (2006) study is the first one that using upper bounds obtained by the formulation to investigate the performance of the developed heuristic procedure.

3.2.2. Dell’Amico et al.’s (2006) Formulation

Dell’Amico et al.’s (2006) formulation, abbreviated as DRSM, is a two-index formulation. The decision variables in DRSM are as follows ($y_{ij}$ and $z_{ij}$ are defined as previously):

Decision Variable:

$$x_{ij}; \begin{cases} 1 & \text{if vehicle travels directly from node } i \text{ to node } j \ (\forall i,j \in N) \\ 0 & \text{otherwise} \end{cases}$$

The formulation, DRSM, is as follows:

$$\begin{array}{cl}
\text{min} & \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \\
\text{s.t.} & \sum_{i \in N} x_{ij} = 1 \quad \forall j \in N_0 \\
& \sum_{j \in N} x_{ij} = \sum_{j \in N} x_{ji} \quad \forall i \in N \\
& y_{ij} + z_{ij} \leq Q x_{ij} \quad \forall i, j \in N, i \neq j \\
& x_{ij} \in \{0,1\} \quad \forall i, j \in N
\end{array}$$

(17)

In this model, the objective function (17) minimizes the total cost (distance). Constraints (18) ensure that each customer must be serviced exactly once. Constraints (19) are known as flow conservation equations which are enforcing the route continuity. Constraint (20) ensures
that the total load a on vehicle does not exceed the vehicle capacity. Finally, constraint (21) imposes binary conditions on the variables. DRSM has \( O(n^2) \) binary variables, \( O(n^2) \) additional variables and \( O(n^2) \) constraints in DRSM.

Dell’Amico et al. (2006) have also developed an exact branch-and-price algorithm for VRPSPD in their study. This algorithm is able to optimally solve the problems up to 40 customers. To the best of our knowledge, Dell’Amico et al. (2006) are the first researches who propose an exact algorithm for VRPSPD.

4. New Formulations

The existing formulations summarized in the Section 3 for VRPSPD have \( O(n^2) \) constraints. However, the three-index node-based and flow-based formulations proposed by Dethloff’s (2001) and Montane and Galvao’s (2006), respectively, have \( O(n^2k) \) binary variables, while Dell’Amico et al.’s (2006) formulation, which is a two-index flow-based formulation, has \( O(n^2) \) binary variables. As known, any commercial software has an increasing computational burden with the number of binary variables in a formulation because the size of enumeration tree in any software increases exponentially based on the number of binary variables. In addition, the memory requirement of any commercial software increases with size of the matrix dimensioned with the number of variables and constraints. Thus, it is expected that the formulations with three-index formulations have poor performance than that of one with two-index variables in terms of LP relaxation, CPU time, etc. Meanwhile, it is possible to obtain more strong formulation by introducing new bounding constraint to DRSM. Based on these observations, we propose two new formulations, abbreviated as PM1 and PM2, which are node-based and flow-based with two-index variables, respectively, in this paper. These formulations have \( O(n^2) \) binary variables and \( O(n^2) \) constraints, and are derived from the formulations developed by Kara (2008) for VRP. As far as we are aware, PM1 is also the first two-index node-based formulation for VRPSPD in the literature.

4.1. Proposed Node Based Formulation (PM1)

In PM1, which is a two-index node-based formulation, we adapt Miller-Tucker-Zemlin (MTZ) capacity and subtour elimination constraints for VRPSPD. These constraints have been proposed by Miller et al. (1960) for the Traveling Salesman Problem (TSP). Kulkarni and Bhave (1985) have adapted it to the capacitated VRP and Desrochers and Laporte (1991)
have lifted them and introduced new bounding constraints on additional variables. Finally, Kara et al. (2004) have corrected the lifted version of these constraints. Meanwhile, exact meanings to additional variables have been given by Kara (2008). The decision variables in PM1 are as follows ($x_{ij}$ is defined as previously):

Additional Variables:
$U_i$; delivery load on vehicle just before having serviced customer $i$ ($\forall i \in N_0$)
$V_i$; pickup load on vehicle just after having serviced customer $i$ ($\forall i \in N_0$)

The proposed first formulation, PM1, is as follows:

\[
\min (17) \text{ subject to (18), (19), (21), and }
\]
\[
U_j - U_i + Qx_{ij} + (Q-d_j -d_i)x_{ji} \leq Q - d_i \quad \forall i, j \in N_0, i \neq j \tag{22}
\]
\[
V_i - V_j + Qx_{ij} + (Q-p_j -p_i)x_{ji} \leq Q - p_j \quad \forall i, j \in N_0, i \neq j \tag{23}
\]
\[
U_i + V_i - d_i \leq Q \quad \forall i \in N_0 \tag{24}
\]
\[
U_i \geq d_i + \sum_{j \in N_0, j \neq i} d_j x_{ij} \quad \forall i \in N_0 \tag{25}
\]
\[
U_i + (Q-d_i)x_{0i} \leq Q \quad \forall i \in N_0 \tag{26}
\]
\[
V_i \geq p_i + \sum_{j \in N_0, j \neq i} p_j x_{ji} \quad \forall i \in N_0 \tag{27}
\]
\[
V_i + (Q-p_i)x_{0i} \leq Q \quad \forall i \in N_0 \tag{28}
\]

In this formulation, $x_{ij} = 0$ whenever $h_{ij} > Q$ where

\[
h_{ij} = \max \{d_i + d_j; p_i + p_j; d_j + p_i\} \tag{29}
\]

Constraints (22) and (23) eliminate the subtours and describe the delivery and pickup loads on vehicles, separately. Constraint (24) ensures that the total load on vehicle does not exceed the vehicle capacity. Constraints (25)-(28) are called bounding constraints and restrict the upper and lower bounds of additional decision variables. In PM1, there are $O(n^2)$ binary variables, $O(n)$ additional variables and $O(n^2)$ constraints.
In this formulation, if $x_{ij} = 1$, then $U_i \geq U_j + d_i$ and $U_i \leq U_j + d_i$ because of the constraints (22) for $(i,j)$ and $(j,i)$ pairs, separately. This implies that $U_i = U_j + d_i$. These equations perform a monotonic decreasing of $U_i$ variables in accordance with delivery amount of nodes on a feasible route. Because of this behavior of $U_i$ variables, any feasible solution contains no illegal subtours. Additionally, constraints (25) and (26) ensures that if $x_{ri} = 1$, then $U_i = d_i$. In other words, if customer $i$ is the last node on a feasible route, then $U_i$ variable is equal to its delivery amount. Together with (22), these equations imply that $U_i$ variables give exact values of delivery load on vehicle just before having serviced customer $i$ on any feasible route. Similar observations can be obtained on $V_i$ variables. Finally, $U_i - d_i$ gives the delivery load on vehicle just after having serviced customer $i$, summation of this with $V_i$ gives an exact load on vehicle after having serviced customer $i$, which is the left side of constraint (24). Therefore, constraint (24) satisfies the capacity restrictions.

4.2. Proposed Flow Based Formulation

The second model (PM2) is a two-index flow-based formulation. In PM2, we introduce new bounding constraints into the Dell’Amico et al.’s (2006) formulation. These bounding constraints are related with additional variables which determine the total delivery / pickup load on the arcs.

The proposed second formulation, PM2, is as follows (decision variables are defined as previously):

$$\text{min } (17) \text{ subject to } (13), (14), (16), (18)-(21), \text{ and}$$

$$\sum_{j \in N_0} y_{0j} = Q_P \quad \text{(30)}$$

$$\sum_{j \in N_0} y_{0j} = 0 \quad \text{(31)}$$

$$\sum_{j \in N_0} z_{0j} = Q_D \quad \text{(32)}$$

$$\sum_{i \in N_0} z_{i0} = 0 \quad \text{(33)}$$

$$y_{ij} \leq (Q - p_j) x_{ij} \quad \forall i \in N, \forall j \in N_0, i \neq j \quad \text{(34)}$$
\[ z_i \leq (Q - d_i) x_{ij} \quad \forall i \in N_0, \forall j \in N, i \neq j \] (35)

\[ y_{ij} \geq p_{ij} x_{ij} \quad \forall i \in N_0, \forall j \in N, i \neq j \] (36)

\[ z_{ij} \geq d_{ij} x_{ij} \quad \forall i \in N, \forall j \in N_0, i \neq j \] (37)

where, \( Q_p = \sum_{i \in N} d_i \) and \( Q_p = \sum_{i \in N} p_i \). In this formulation, \( x_{ij} = 0 \) whenever \( h_{ij} > Q \) where \( h_{ij} \) as explained previously. Constraint (30) ensures that the total pickup load on the last arcs is equal to the total pickup demand, constraint (31) guarantees that load on the first arcs is equal to “0”. Similarly, constraint (32) and (33) determine the delivery loads on the first and the last arcs. Finally, constraints (34)-(37) are called as bounding constraints and restrict the upper and lower bounds of additional decision variables. PM2 has \( O(n^2) \) binary variables, \( O(n^2) \) additional variables and \( O(n^2) \) constraints.

4.3. Relationships Between Proposed Formulations

In most combinatorial optimization problems, the efficiency of any MIP solver or an exact algorithm based on MIP formulation depends on the strength of the linear programming (LP) relaxation of a given formulation. Valid inequalities are one of the most practical ways to strengthen the LP relaxations of the formulations. In this paper, we utilize following valid inequality for PM2.

\[ x_{ij} + x_{ji} \leq 1 \quad \forall i, j \in N_c \] (38)

Constraints (38) ensure that any feasible route does not contain a subtour with only two customers. Note that constraints (38) are useful for PM2, however it can be easily verified that it is not a facet defining inequality for PM1 as seen in Appendix A.

In the literature, it is shown that the LP bounds of flow based formulations are better than node based formulations for classical VRP (Letchford and Salazar-Gonzalez, 2006) and extension of VRP (Yaman, 2006). At this point, it is interesting to compare LP bounds of proposed formulations for VRPSPD.

**Theorem:** PM2 gives better LP bounds than PM1 (\( LP(\text{PM1}) \leq LP(\text{PM2}) \))

**Proof:** By using the exact meanings of additional variables, following relationships are obtained.

\[ U_i = \sum_{j \in N} z_{ji} \] (39)

\[ V_i = \sum_{j \in N} y_{ij} \] (40)
To show that PM2 gives better LP bounds than PM1, it is enough to prove that the capacity, subtour elimination and bounding constraints (22)-(28) are dominated by PM2. We begin with constraints (24).

Using (39) and (40), (24) can be rewritten as \( \sum_{j \in N} z_{ij} + \sum_{j \notin N} x_{ij} - d_i \leq Q \). With the help of (14), this can be simplified as \( \sum_{j \in N} z_{ij} + \sum_{j \notin N} x_{ij} \leq Q \). On the other hand, this constraint can also be obtained by summing up and simplifying constraints (20) over \( j \). It is well known that disaggregated constraints give tighter LP bounds (Appa et al., 2006). Hence, constraints (24) dominated by PM2.

Next, we show that any feasible solution for the LP relaxation of PM2 satisfies (25). Using (39), (25) can be rewritten as \( \sum_{j \in N} z_{ij} \geq d_i + \sum_{j \notin N} d_j x_{ij} \). Using (14), this can be simplified as \( \sum_{j \in N} z_{ij} \geq \sum_{j \notin N} d_j x_{ij} \), and can be rewritten as \( z_{i0} + \sum_{j \in N} z_{ij} \geq \sum_{j \notin N} d_j x_{ij} \).

With the help of (33), the first part of left hand side can be removed. This constraint can also be obtained by summing up and simplifying constraints (37) over \( j \). It means that constraints (25) is an aggregation of constraints (37). Therefore, constraints (25) dominated by PM2.

In a similar way, we can prove that any feasible solution for the LP relaxation of PM2 satisfies (26). Using (39), (26) can be rewritten as \( \sum_{j \in N} z_{ij} + (Q - d_i) x_{i0} \leq Q \). Then, using (14), this can be written as \( \sum_{j \in N} z_{ij} \leq (Q - d_i)(1 - x_{i0}) \). Since \( \sum_{j \in N} x_{ij} = 1 \), the right hand side can be changed as \( (Q - d_i) \sum_{j \in N} x_{ij} \). The left hand side can also be rewritten as \( \sum_{j \in N} z_{ij} + z_{i0} \). With the help of (33), this inequality can be decomposed as \( z_{ij} \leq (Q - d_i) x_{ij} \) for all \( i, j \in N, i \neq j \), which is an aggregation of (35). Therefore, constraints (26) dominated by PM2.

Now, we will prove that any feasible solution for the LP relaxation of PM2 satisfies constraints (22). In any feasible solution of PM2, because of (14), \( \sum_{k \in N} z_{ik} - \sum_{k \in N} z_{ik} = d_i \) and \( \sum_{k \in N} z_{kj} - \sum_{k \in N} z_{jk} = d_j \) must be satisfied, simultaneously, for any pair of customer pairs \( \forall i, j \in N, i \neq j \). By the help of (39), these equations can be combined as \( U_j - U_i + \sum_{k \in N} z_{ik} - \sum_{k \in N} z_{jk} = d_j - d_i \). Using (35) and (37), this equality can be rewritten as \( U_j - U_i + \sum_{k \in N} (Q - d_i) x_{ik} - \sum_{k \in N} d_k x_{jk} \leq d_j - d_i \). After some rearrangement and simplification this inequality is converted to \( U_j - U_i \leq d_j - Q + \sum_{k \in N} d_k x_{jk} \). On the other
hand, (22) can be rewritten as \( U_j - U_i \leq Q - d_j - Q x_{ij} - (Q - d_j - d_i) x_{ji} \). At this point, it is enough to compare the right hand side of the last two inequalities.

\[
Q - d_j - Q x_{ij} - (Q - d_j - d_i) x_{ji} \geq d_j - Q + \sum_{k \in N} d_k x_{jk} \quad \forall i, j \in N_0, i \neq j
\]  

(41)

After some rearrangement, \( Q(2-x_{ij}-x_{ji}) \geq d_j + d_j - d_j x_{ji} + \sum_{k \in N} d_k x_{jk} \) is obtained from (41). In the worst case, \( x_{ij} = 1, \ x_{ji} = 0 \) because of (38), and \( x_{jk} = 1 \) where \( d_k = \max \{d_l : l \in N_0, l \neq i \neq j\} \). After these assignments, \( Q \geq d_i + d_j + d_k \) is obtained. This situation shows that \( i-j-k \) is a part of any solution and because of capacity constraints the right hand side is always smaller or equal to \( Q \). This means that, (41) is always true. Therefore, constraints (22) dominated by PM2.

In a similar way, the domination of other constraints of PM1 ((23), (27) and (28)) can also be validated. Therefore, PM2 gives better LP bounds than PM1. This completes the proof.

5. Computational Analysis

In this section, after brief information about the test problems, we present results of an experimental study conducted to identify an efficient one among five formulations.

5.1. Test Problems

We consider two sets of test problems. The first set is adapted from test problems given by Christofides et al. (1979) for the capacitated VRP. There are 14 test problems in Christofides et al.’s (1979) set and the number of customers change between 50 and 199. We consider 5 problems in this set (i.e. 1\textsuperscript{st}, 2\textsuperscript{nd}, 3\textsuperscript{rd}, 11\textsuperscript{th} and 12\textsuperscript{th} instances) having different coordinate and demand. In order to obtain VRPSPD test problems, first 10, 15 and 20 customers in those problems are considered. The Euclidean distances between customers are rounded to the nearest integer. In order to generate the delivery and pickup demands, separation methods proposed by Salhi and Nagy (1999) and Angelelli and Mansini (2002) are used. These methods are briefly explained as follows. In the first separation method, a ratio \( r_i = \min(x_i / y_i; y_i / x_i) \) is calculated for all customers, then the delivery and pickup demands are generated as \( d_i = r_i q_i \) and \( p_i = q_i - d_i \), where \( q_i \) is the original demand of customer \( i \).

We refer to this type of problems as X. Similarly, another type of problem, called Y, is generated by exchanging delivery and pickup demands. In the second separation method, the original demand of the test data is considered as delivery demand \( (d_i = q_i) \) and the pickup
demand is generated as \( p_i = \left\lfloor (1-\gamma)q_i \right\rfloor \) if \( i \) is even and \( p_i = \left\lfloor (1+\gamma)q_i \right\rfloor \) if \( i \) is odd. In this paper, we consider two \( \gamma \) values as 0.2 and 0.8 to generate two different types of problems called \( Z \) and \( W \), respectively. Additionally, to investigate the effect of vehicle capacity on the performance of formulations, we consider different vehicle capacities. In each test instance, three vehicle capacities are generated by \( Q = \lambda \times \max(d_i, p_i) \) where \( \lambda \) is taken as 1, 2 and 4, respectively. As a result, we generate totally 180 different test problems in the first set considering 5 different test instances, 3 levels of the number of customers, 4 different separation strategies and 3 different vehicle capacities. In rest of the paper, we refer to this set as TS1.

The second set of test problems is adapted from the test instances given in Dethloff (2001), which are originally generated for VRPSPD. Dethloff’s (2001) test set includes 40 randomly generated test instances with 50 customers who are located considering two different geographical scenarios, called SCA and CON. In the SCA, the coordinates of the customers are uniformly distributed over the interval \([0,100]\). In the CON, while one half of the customers are distributed in the same way as in the SCA, the coordinates of the other half are uniformly distributed over the interval \([100/3, 200/3]\). Euclidean distances are used in both cases. The delivery demands of the customers are uniformly distributed between \([0,100]\) and the pickup amount of each customer is generated according to delivery demand such that \( p_i = (0.5 + r_i)d_i \), where \( r_i \) is a random number between \([0,1]\). Additionally, three different vehicle capacities are considered and these are generated in a similar fashion explained for the first test set. We consider the first 10, 15 and 20 customers of the instances in order to obtain small size test problems and generate totally 180 different test problems in the second set by using 2 different geographical scenarios, 3 different capacity levels, 3 levels of the number of customers and 10 test instances in each combination of these factors. This set is denoted by TS2 in rest of the paper.

5.2. Computational Results

Our experimental study consists in solving MIP formulations and their LP relaxations with the state-of-the-art LP/MIP solver GAMS/CPLEX (V10.2) on Intel Xeon 3.16 GHz equipped with 1 GB RAM computer (the operating system is Windows Vista Business with SP1). We use the default CPLEX parameters to solve MIP formulations and their LP relaxations. Meanwhile, we limit the computation time (CPU time) of the formulations with two hours.
To analyze the computational results, we use following performance measures: the number of optimal solutions ($Z_{NUM}^*$) obtained within two hours of CPU time, average CPU time (ACT) in seconds and two different gap values. The gap values are defined as follows:

1) LP Relaxation Gap ($GapLR$): It is the gap between the objective function value of LP relaxation ($Z^{LR}$) corresponding to a particular formulation and the optimal or the best known integer feasible solution ($Z^*$) which is the best objective value obtained by solving the formulations with GAMS/CPLEX for a maximum of two hours. The $GapLR$ is calculated by equation (42):

$$GapLR = 100 \times \left( \frac{Z^* - Z^{LR}}{Z^*} \right)$$

(42)

2) Optimality gap ($GapOp$): It is the gap between the lower bound ($Z^{LB}$) and upper bound when the corresponding formulation is solved by GAMS/CPLEX within two hours of CPU time. The $GapOp$ is calculated by equation (43):

$$GapLB = 100 \times \left( \frac{Z^{UB} - Z^{LB}}{Z^{LB}} \right)$$

(43)

These gap values report how close the corresponding bounds to the optimal / best known solution. It is well-known that the quality of the bounds is one of the critical issues in reducing execution time of exact approaches. A tighter bound can help the commercial solvers to get optimal solutions in a shorter time (Kashan and Karimi, 2008).

Tables 1 and 2 present computational results in terms of the number of customers and the vehicle capacities for TS1 and TS2 which are adapted from Christofides et al. (1979) and Dethloff (2001) test sets, respectively. Each cell in the tables reports average (total) value for the corresponding performance measure over 20 test instances. The general impression from Tables 1 and 2 is that two-index formulations perform better than three-index formulations in terms of $Z_{NUM}^*$, ACT and optimality gap. Moreover, the proposed second formulation, PM2, which is a two-index flow-based formulation, exhibits better performance than other formulations with respect to all performance measures. While maximum linear relaxation and optimality gaps of PM2 are 9.26% and 1.32%, respectively, it reaches optimum solutions in 340 out of 360 test instances in a given computation time limit. Furthermore, its ACT is between 0.07 and 2722 seconds. It is worthy to note that the main difference between PM2 and DRSM is bounding constraints. As it is seen from computational results, these constraints significantly improves the performance of PM2 over DRSM. Meanwhile, DRSM and PM1, which are two-index flow-based and node-based formulations, respectively, have similar
performance in terms of $Z_{\text{NUM}}$, ACT and optimality gap. DRSM and PM1 reach optimum solutions 325 and 319 out of 360 test instances, respectively, and maximum optimality gaps in these formulations are 3.78% and 4.97%, respectively. However, linear relaxation gaps of PM1 are slightly better than that of DRSM in most of the test instances. It is also interesting to note that DRSM is second one which is having poor quality linear relaxation gaps after DM. Finally, as seen from Tables 1 and 2, DM has poor performance in terms of all measures. Its average LP relaxation and optimality gaps are around 23% and 12%, respectively, and also it reaches optimal solutions only in 162 out 360 problems.

The results in Tables 1 and 2 indicate that the performance of all formulations deteriorate with the number of customers. Especially, DM and MGM do not reach optimal solutions as frequently as DRSM, PM1 and PM2, when the number of customers is increased. Moreover, both formulations fail to find integer solutions in some test instances with 20 customers. In a similar fashion, when the vehicle capacity is lowered, i.e. $\lambda$ is taken as 1 and 2, all formulations are compelling to solve test instances. The two-index formulations, DRMS, PM1 and PM2, have a difficulty to solve the problems especially when $\lambda$ is 2. As an example, we can consider PM2 for the problem with 20 customers in TS2. When $\lambda$ is 2, PM2 needs 2525.01 seconds to find integer or optimal solutions for those problems and its LP relaxation and optimality gaps are 9.47% and 0.30%, respectively. However, PM2 finds integer or optimal solutions in 536.07 and 156.19 seconds for the same problems when $\lambda$ is taken as 1 and 4, respectively, and also its LP relaxation gaps reduce to 6.95% and 7.40% for the corresponding $\lambda$ values, respectively. In contrary of two-index formulations, the performance of three-index formulations, i.e. DM and MGM, deteriorates when $\lambda$ is reduced. It means that these formulations exhibits worst performance when $\lambda$ is 1.

Finally, we investigate the effects of four separation methods, which are used to determine delivery and pickup demand for each customer in each test instance of TS1, and two geographical scenarios of TS2, i.e. the SCA where the customers are uniformly distributed in a grid and the CON where one half of the customers are clustered. Our preliminary study shows that the performances of all formulations are not affected from separation methods. However, geographical scenarios influence the performance of all formulations especially in terms of ACT. Table 3 reports computational results for TS2 according to geographical scenarios when $\lambda = 1$ for the vehicle capacity. As seen from the table, the two-index formulations solve test instances with the CON scenario much easily than
<table>
<thead>
<tr>
<th>N</th>
<th>λ</th>
<th>DM</th>
<th>MGM</th>
<th>DRMS</th>
<th>PM1</th>
<th>PM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>19.54</td>
<td>4.61</td>
<td>3830.49</td>
<td>13.90</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>11.71</td>
<td>1.80</td>
<td>422.83</td>
<td>8.26</td>
<td>1.67</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6.53</td>
<td>0.00</td>
<td>1.26</td>
<td>5.22</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.59</td>
<td>2.14</td>
<td>1418.19</td>
<td>9.13</td>
<td>0.83</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>25.95</td>
<td>19.05</td>
<td>7200.00</td>
<td>15.63</td>
<td>8.48</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>16.73</td>
<td>7.04</td>
<td>6460.65</td>
<td>9.57</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>13.17</td>
<td>0.00</td>
<td>374.60</td>
<td>9.25</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18.62</td>
<td>8.70</td>
<td>4678.42</td>
<td>11.48</td>
<td>3.24</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>32.14</td>
<td>33.96*</td>
<td>7200.00</td>
<td>17.34</td>
<td>14.34</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>25.22</td>
<td>30.52</td>
<td>7200.00</td>
<td>13.06</td>
<td>11.09</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>21.58</td>
<td>17.69</td>
<td>5333.13</td>
<td>13.10</td>
<td>9.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26.31</td>
<td>27.39</td>
<td>6577.71</td>
<td>14.50</td>
<td>11.79</td>
</tr>
</tbody>
</table>

*: Average value is calculated considering test instances in which upper bounds are obtained
Table 2. Computational Results for TS2

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\lambda$</th>
<th>DM</th>
<th>MGM</th>
<th>DRMS</th>
<th>PM1</th>
<th>PM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>27.03</td>
<td>0.40</td>
<td>1309.64 (18)</td>
<td>0.00</td>
<td>54.88 (20)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>21.04</td>
<td>0.00</td>
<td>291.16 (20)</td>
<td>0.00</td>
<td>45.79 (20)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>15.54</td>
<td>0.00</td>
<td>3.19 (20)</td>
<td>12.71</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Average (Total) | 21.20 | 0.13 | 534.66 (58) | 13.95 | 0.00 | 34.47 (60) | 16.81 | 0.00 | 0.73 (60) | 10.50 | 0.00 | 0.21 (60) | 5.19 | 0.00 | 0.38 (60) |

| 15  | 1         | 32.60 | 17.27 | 7200.00 (0) | 18.40 | 4.81 | 7200.00 (0) | 19.70 | 0.00 | 55.37 (20) | 17.51 | 0.00 | 26.78 (20) | 6.00 | 0.00 | 17.89 (20) |
|     | 2         | 25.32 | 9.75 | 6468.72 (3) | 15.10 | 3.49 | 5368.93 (6) | 17.16 | 0.00 | 40.50 (20) | 16.15 | 0.00 | 29.09 (20) | 7.32 | 0.00 | 49.22 (20) |
|     | 4         | 20.53 | 0.63 | 1877.56 (18) | 15.40 | 0.09 | 1077.36 (19) | 17.46 | 0.00 | 1.63 (20) | 9.30 | 0.00 | 3.87 (20) | 5.86 | 0.00 | 5.01 (20) |

Average (Total) | 26.15 | 9.22 | 5182.09 (21) | 16.30 | 2.80 | 4554.76 (25) | 18.11 | 0.00 | 32.50 (60) | 14.32 | 0.00 | 19.91 (60) | 6.39 | 0.00 | 24.04 (60) |

| 20  | 1         | 35.73 | 26.83* | 7200.00 (0) | 19.33 | 11.57* | 7200.00 (0) | 20.34 | 0.37 | 1055.83 (19) | 21.34 | 0.00 | 457.16 (19) | 6.95 | 0.14 | 536.07 (19) |
|     | 2         | 30.48 | 26.87 | 7200.00 (0) | 17.48 | 12.43 | 7200.00 (0) | 18.63 | 0.53 | 3498.55 (14) | 21.43 | 0.96 | 3466.53 (13) | 9.47 | 0.30 | 2525.01 (18) |
|     | 4         | 24.00 | 8.41 | 7045.40 (1) | 16.70 | 10.30 | 7200.00 (0) | 17.92 | 0.00 | 146.44 (20) | 12.44 | 0.00 | 685.71 (19) | 7.40 | 0.00 | 156.19 (20) |

Average (Total) | 30.07 | 20.70 | 7148.47 (1) | 17.84 | 11.43 | 7200.00 (0) | 18.96 | 0.30 | 1566.94 (53) | 18.40 | 0.32 | 1536.47 (51) | 7.94 | 0.15 | 1072.42 (57) |

*: Average value is calculated considering test instances in which upper bounds are obtained
Table 3. Effects of Geographical Scenarios of TS2 on the Performance of Formulations When $\lambda = 1$ for the Vehicle Capacity

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CON</td>
<td>28.80</td>
<td>0.00</td>
<td>640.78</td>
<td>15.50</td>
<td>0.00</td>
<td>64.51</td>
<td>16.64</td>
<td>0.00</td>
<td>0.00</td>
<td>0.60 (10)</td>
</tr>
<tr>
<td></td>
<td>SCA</td>
<td>25.26</td>
<td>0.80</td>
<td>1978.50</td>
<td>15.72</td>
<td>0.00</td>
<td>45.25</td>
<td>18.81</td>
<td>0.00</td>
<td>0.71</td>
<td>0.07 (10)</td>
</tr>
<tr>
<td>15</td>
<td>CON</td>
<td>37.13</td>
<td>15.06</td>
<td>7200.00</td>
<td>19.21</td>
<td>3.31</td>
<td>7200.00</td>
<td>19.79</td>
<td>0.00</td>
<td>5.59</td>
<td>1.70 (10)</td>
</tr>
<tr>
<td></td>
<td>SCA</td>
<td>28.06</td>
<td>19.47</td>
<td>7200.00</td>
<td>17.59</td>
<td>6.31</td>
<td>7200.00</td>
<td>19.61</td>
<td>0.00</td>
<td>105.15</td>
<td>51.87 (10)</td>
</tr>
<tr>
<td>20</td>
<td>CON</td>
<td>38.92</td>
<td>25.60</td>
<td>7200.00</td>
<td>19.67</td>
<td>12.07*</td>
<td>7200.00</td>
<td>20.31</td>
<td>0.00</td>
<td>566.65</td>
<td>55.66 (10)</td>
</tr>
<tr>
<td></td>
<td>SCA</td>
<td>32.53</td>
<td>28.36*</td>
<td>7200.00</td>
<td>18.98</td>
<td>11.13</td>
<td>7200.00</td>
<td>20.37</td>
<td>0.73</td>
<td>1545.02</td>
<td>885.65 (9)</td>
</tr>
</tbody>
</table>

*: Average value is calculated considering test instances in which upper bounds are obtained
that of the SCA scenario. For those problems, the two index-formulations need at least two times higher ACT for the problems with SCA scenario than that of the CON scenario. For example, ACTs of the DRMS in the CON and SCA scenarios are 566.65 and 1545.02 seconds for the problems with 20 customers, respectively. Meantime, PM2 has higher LP relaxation gaps for the problems with the SCA scenario. However, similar situation is not observed for the other formulations.

6. Conclusion

In this paper, we have considered VRP with simultaneous pickup and delivery, VRPSPD. After reviewed three formulations for VRPSPD in the literature, we have proposed two new formulations. While the first proposed formulation, PM1, is a two-index node-based formulation, the second proposed formulation, PM2, is a two-index flow-based formulation. We have presented computational results by solving five different formulations and LP relaxations with GAMS/CPLEX on two sets of test problems derived from the literature. The computational results over 360 test problems have shown that the proposed second formulation, PM2, which is two-index flow-based formulation, outperforms the other formulations in terms of the number of optimal solutions obtained within two hours of CPU time, linear relaxation gap, optimality gap and average CPU time. Meanwhile, the proposed first formulation, which is two-index node-based formulation, PM1, exhibits better performance than the formulations with three-index and also similar performance with the two-index flow-based formulation given in the literature. To the best of our knowledge, this paper is the first attempt that compares the computational performance of the formulations developed for VRPSPD. As a future research, the formulations can be compared considering other restrictions in VRPSPD, such as distance constraint, etc. Also, it would be worthwhile developing a branch and cut algorithm based on the proposed formulations for VRPSPD and developing effective valid inequalities (i.e. multistar, comb, hypotour inequalities etc.) and heuristic solution procedures to be used in the branch and cut algorithm.

Acknowledgement:

This research is supported by The Scientific and Technological Research Council of Turkey as Scientific Research Project (No: 108E069). Additionally, we thank to Dr. Jan Dethloff for providing us VRPSPD test problems.
References


Appendix A.

**Proposition 1:** Constraints (38) is not a facet defining inequality for PM1

**Proof:** In any feasible solution of PM1, because of the constraints (22), following inequalities must be satisfied, simultaneously, for any pair of customer pairs \((i, j) \in N_0\).

\[
U_j - U_i \leq Q - d_i - Q(x_{ij} + x_{ji}) + x_{ij}(d_i + d_j) \tag{44}
\]

\[
U_j - U_j \leq Q - d_j - Q(x_{ij} + x_{ji}) + x_{ij}(d_i + d_j) \tag{45}
\]

Then, following inequality is obtained by summing both sides of (44) and (45).

\[
0 \leq 2Q - (d_i + d_j) - 2Q(x_{ij} + x_{ji}) + (x_{ij} + x_{ji})(d_i + d_j) \tag{46}
\]

After some rearrangements, following form of (46) is obtained.

\[
2Q(x_{ij} + x_{ji} - 1) \leq (d_i + d_j)(x_{ij} + x_{ji} - 1) \tag{47}
\]

Let \((x_{ij} + x_{ji} = 1 + \xi)\). Therefore, inequality (47) reduces to

\[
2\xi Q \leq \xi (d_i + d_j) \tag{48}
\]

In this situation, for any value of \(\xi > 0\), inequality (48) is not satisfied. It means that for any feasible solution of PM1, the condition of \((x_{ij} + x_{ji} \leq 1)\) must be satisfied. Therefore, constraints (38) are valid but not facet defining inequality for PM1. ■