A branch-cut-and-price algorithm for the energy minimization vehicle routing problem

Ricardo Fukasawa[*], Qie He[†], Yongjia Song[‡]

August 25, 2014

Abstract

We study a variant of the capacitated vehicle routing problem where the cost over each arc is defined as the product of the arc length and the weight of the vehicle when it traverses that arc. We propose two new mixed integer linear programming formulations for the problem: an arc-load formulation and a set partitioning formulation based on q-routes with additional constraints. A family of cycle elimination constraints are derived for the arc-load formulation. We then compare the linear programming (LP) relaxations of these formulations with the two-index one-commodity flow formulation proposed in the literature. In particular, we show that the arc-load formulation with the new cycle elimination constraints gives the same LP bound as the set partitioning formulation based on 2-cycle-free q-routes, which is stronger than the LP bound given by the two-index one-commodity flow formulation. We propose a branch-and-cut algorithm for the arc-load formulation, and a branch-cut-and-price algorithm for the set partitioning formulation strengthened by additional constraints. Computational results on instances from the literature demonstrate that a significant improvement can be achieved by the branch-cut-and-price algorithm over other methods.

1 Introduction

Vehicle routing problems have been studied extensively in the literature (Toth and Vigo, 2001; Golden et al., 2008; Laporte, 2009). Recently, with an increasing concern for the environment, new vehicle routing models started being considered, where the objective was not only to take into account the travel costs or distances, but also energy consumption and green house gas emissions, see for instance Kara et al. (2008); Bektaş and Laporte (2011); Lysgaard and Wøhlk (2014) and a recent survey Demir et al. (2014).

We study one such variant of the capacitated vehicle routing problem (CVRP), called the energy minimization vehicle routing problem (EMVRP). As in the traditional CVRP, the goal of the EMVRP is to find a set of least costly routes such that (i) each route starts and ends at the depot, (ii) each customer is visited by a single vehicle, and (iii) the total supply of all customers in any route does not exceed the vehicle capacity. The key difference is that the cost of traversing an arc in the EMVRP is defined as the product of the arc length and the vehicle weight, namely the sum of the curb weight and payload, when it traverses that arc. The EMVRP was first introduced

[*]Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo. Email: rfukasaw@math.uwaterloo.ca.

[†]Department of Industrial and Systems Engineering, University of Minnesota. Email: qhe@umn.edu.

[‡]Department of Statistical Sciences and Operations Research, Virginia Commonwealth University. Email: ysong3@vcu.edu.

in (Kara et al., 2007, 2008) and the cost function was motivated by the positive correlation between the fuel consumption rate and the vehicle weight, with other factors, such as vehicle specifications, vehicle speed and road conditions, being constant. In addition to the practical concern of minimizing fuel consumption, the EMVRP is an interesting problem to study on its own since it generalizes several well-known transportation models such as the traveling repairman/deliveryman problem (Afrati et al., 1986; Fischetti et al., 1993; Bianco et al., 1993; Blum et al., 1994) and the multiple traveling repairmen problem (Fakcharoenphol et al., 2003).

Various mixed integer linear programming (MILP) models have been proposed for solving the traditional CVRP (see for instance Letchford and Salazar-González (2006)) and the most successful algorithms in practice are the ones based on a set partitioning formulation using q-routes and/or ng-routes, strengthened by additional constraints, solved in a branch-cut-and-price (BCP) framework (Fukasawa et al., 2006; Baldacci and Mingozzi, 2009; Baldacci et al., 2011; Pecin et al., 2014; Contardo and Martinelli, 2014). These types of formulations have been successful to solve instances of the problem with up to 300 vertices. Moreover, similar techniques have been applied to other routing problems with reasonable success, such as the asymmetric CVRP, the capacitated open vehicle routing problem, the heterogeneous fleet vehicle routing problem and many others (Pessoa et al., 2008), typically solving instances with close to 100 customers.

On the other hand, the existing approach for solving the EMVRP exactly was based on an MILP model that was able to solve only small-size instances with up to 31 customers (Kara et al., 2008). A constant-factor approximation algorithm was developed in (Gaur et al., 2013), and meta-heuristics such as simulated annealing (Xiao et al., 2012) were also applied to solve the EMVRP.

With this in mind, the main contributions of this paper are as follows.

- We study how the approaches for the CVRP can be adapted to solve the EMVRP. Specifically, we study an arc-load formulation and a set partitioning formulation based on 2-cycle-free q-routes with additional constraints.

- We derive a family of cycle elimination constraints for the arc-load formulation based on the cycle elimination constraints proposed for the time-dependent traveling salesman problem (Abeledo et al., 2013).

- We compare the strength of the two proposed formulations with each other and with the two-index one-commodity flow formulation proposed in the literature. In particular, we show that the arc-load formulation with the new cycle elimination constraints gives the same LP bound as the set partitioning formulation based on 2-cycle-free q-routes. This generalizes a result from Abeledo et al. (2013). In addition, these theoretical results can be applied immediately to the analogous formulations for the CVRP.

- We develop a branch-cut-and-price algorithm to solve the set partitioning formulation based on 2-cycle-free q-routes and strengthened by additional constraints. We conduct an extensive computational study on our proposed formulations and show that the BCP algorithm yields a significant improvement over other methods.

The rest of the paper is organized as follows. Section 2 introduces the two-index one-commodity flow formulation and the two new MILP formulations for the EMVRP. Section 3 compares the strength of the LP relaxations of these formulations. Section 4 describes the BCP algorithm for the set partitioning formulation based on 2-cycle-free q-routes, including the column generation and cut separation components. Computational results and analysis are given in Section 5 and we conclude in Section 6.

2 Problem description and formulations

Throughout the paper, we work on the pickup version of the EMVRP with asymmetric distance matrix and homogeneous vehicles. The problem can be described as follows. Let $G = (V, A)$ be a complete directed graph with $V = \{0, 1, 2, \dots, n\}$. Vertex 0 denotes the depot and $V_0 := \{1, 2, \dots, n\}$ denotes the set of customers. For every $(i, j) \in A$, let d_{ij} be the distance from customer i to customer j. For every $i \in V_0$, let q_i be the supply of customer i. We are given K identical vehicles, each of which has curb weight w and capacity Q. Without loss of generality, we assume that all data are integer-valued. We route these K vehicles so that each customer is visited exactly once by some vehicle, i.e., split pickup is not allowed. Each vehicle starts the route carrying only its own weight w, and when it visits customer i, it picks up the supply q_i of customer i and the total weight of this vehicle increases by q_i. Note that with an asymmetric distance matrix, the pickup version of the EMVRP cannot be transformed to an equivalent delivery version in general, as opposed to the CVRP.

2.1 A two-index one-commodity flow formulation

The first MILP formulation of the EMVRP was introduced by Kara et al. (2007). We call it a two-index one-commodity flow formulation, presented in (1), where for each arc $(i, j) \in A$, a binary variable x_{ij} is introduced to represent whether arc (i, j) is used in any route or not, and a continuous variable f_{ij} is introduced to represent the amount of load carried over arc (i, j).

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} d_{ij}(w x_{ij} + f_{ij}) \\
\text{s.t.} \quad & \sum_{j\in V\setminus\{i\}} x_{ij} = 1,\ \forall i \in V_0 && \text{(1a)}\\
& \sum_{j\in V\setminus\{i\}} x_{ji} = 1,\ \forall i \in V_0 && \text{(1b)}\\
& \sum_{j\in V\setminus\{i\}} f_{ij} - \sum_{j\in V\setminus\{i\}} f_{ji} = q_i,\ \forall i \in V_0 && \text{(1c)}\\
& q_i x_{ij} \le f_{ij} \le (Q - q_j) x_{ij},\ \forall (i,j) \in A && \text{(1d)}\\
& \sum_{i\in V_0} x_{i0} = \sum_{i\in V_0} x_{0i} = K && \text{(1e)}\\
& x_{ij}, f_{ij} \ge 0,\ \forall (i,j) \in A && \text{(1f)}\\
& x_{ij} \in \mathbb{Z},\ \forall (i,j) \in A && \text{(1g)}
\end{aligned}
$$

Constraints (1a) and (1b) ensure that each customer is visited exactly once by some vehicle, constraints (1c) ensure that q_i units of supply are collected at customer i, constraints (1d) are the tightened bound constraints on the flow variables f_{ij}, and constraints (1e) ensure that K vehicles are used. This formulation has the same feasible region as that of the two-index one-commodity flow formulation for the CVRP, except that in the objective function we have for each arc $(i, j) \in A$ an additional term $d_{ij} f_{ij}$, and $d_{ij} x_{ij}$ is multiplied by the vehicle weight w. Although the feasible region contains more solutions than the set of feasible routes and corresponding loads (e.g., one can increase the flow values along a route if the upper bound constraints in (1d) are not tight for all arcs on that route), it can be shown that there always exists an optimal solution (x^*, f^*) where f^*_{ij} denotes the actual load carried by a vehicle when it traverses arc (i, j). This motivates the arc-load formulation introduced below, which models the load carried over each arc explicitly.

2.2 The arc-load formulation

The arc-load formulation for the EMVRP introduces a binary variable x_{ij}^q for each arc $(i,j) \in A$ and load value $q = 0, 1, \ldots, Q$ to indicate whether or not a vehicle travels from customer i to customer j carrying load that exactly equals to q. The idea of using decisions variables x_{ij}^q's appears in Uchoa (2011) for the CVRP and is based on a formulation for the time-dependent traveling salesman problem (Picard and Queyranne, 1978). Similar formulations based on binary variables x_{ij}^q's have also appeared in other contexts like the traveling salesman problem and the capacitated minimum spanning tree problem (see Gouveia (1995); Fox et al. (1980); Uchoa et al. (2008)). The arc-load formulation for the EMVRP are as follows.

$$\min \sum_{(i,j)\in A} \sum_{q=0}^{Q} d_{ij}(w+q)x_{ij}^q$$

$$\text{s.t.} \quad \sum_{j\in V\setminus\{i\}} \sum_{q=0}^{Q} x_{ij}^q = 1, \ \forall i \in V_0 \tag{2a}$$

$$\sum_{j\in V\setminus\{i\}} x_{ji}^q - \sum_{j\in V\setminus\{i\}} x_{ij}^{q+q_i} = 0, \ \forall i \in V_0, q = 0, 1, \ldots, Q - q_i \tag{2b}$$

$$\sum_{i\in V_0} \sum_{q=0}^{Q} x_{i0}^q = \sum_{i\in V_0} \sum_{q=0}^{Q} x_{0i}^q = K \tag{2c}$$

$$x_{ij}^q \geq 0, \ \forall (i,j) \in A, q = 0, 1, \ldots, Q \tag{2d}$$

$$x_{ij}^q \in \mathbb{Z}, \ \forall (i,j) \in A, q = 0, 1, \ldots, Q. \tag{2e}$$

We also set $x_{ij}^q = 0$ for all $q = 0, 1, \ldots, q_i - 1, Q - q_j + 1, \ldots, Q$ and $j \in V_0 \setminus \{i\}$ and $x_{0i}^q = 0$ for all $q > 0$. Constraints (2a) ensure that each customer is visited exactly once by some vehicle. Constraints (2b) represent the load balance at each customer $i \in V_0$. Constraints (2c) ensure that K vehicles are used.

We next propose a set of cycle elimination constraints to strengthen the arc-load formulation (2).

Definition 1. *An s-cycle (or a cycle of length s) with $s \geq 2$ is a walk $i_1 i_2 \ldots i_s i_{s+1}$ with different customers $i_1, i_2, \ldots, i_s \in V_0$ and $i_1 = i_{s+1}$.*

Observe that a feasible route cannot contain any s-cycle with $s \geq 2$. To avoid a 2-cycle jij, any route that traverses arc (i,j) with load q must arrive at customer i from some customer $k \neq i, j$ with load $q - q_i$. Motivated by this observation, we propose the following 2-cycle elimination constraints:

$$x_{ij}^q \leq \sum_{k\in V\setminus\{i,j\}} x_{ki}^{q-q_i}, \ \forall i,j \in V_0, \ q = q_i, q_i+1, \ldots, Q - q_j. \tag{3}$$

The 2-cycle elimination constraints can be seen a generalization of the 2-cycle elimination constraints for the time dependent traveling salesman problem proposed in (Abeledo et al., 2013).

The separation of 2-cycle elimination constraints can be done in $O(n^2 Q)$ time as follows. Given a fractional solution $\hat{x} = (\hat{x}_{ij}^q)$ obtained by solving the LP relaxation of the arc-load formulation (2), for each $i \in V_0$ and $q = q_i, q_i+1, \ldots, Q$, we compute $\sum_{k\in V\setminus\{i\}} \hat{x}_{ki}^{q-q_i}$ and check if $\max_{j\in V_0\setminus\{i\}}\{\hat{x}_{ij}^q + \hat{x}_{ji}^{q-q_i}\} > \sum_{k\in V\setminus\{i\}} \hat{x}_{ki}^{q-q_i}$. If so, let $j_0 = \arg\max_{j\in V_0\setminus\{i\}}\{\hat{x}_{ij}^q + \hat{x}_{ji}^{q-q_i}\}$. Then we find a 2-cycle elimination constraint $x_{ij_0}^q \leq \sum_{k\in V\setminus\{i,j_0\}} x_{ki}^{q-q_i}$ that is violated by the solution $\hat{x}$.

To eliminate the same 2-cycle jij, we could also use another set of 2-cycle elimination constraints $x_{ji}^q \leq \sum_{k\in V\setminus\{i,j\}} x_{ik}^{q+q_i}$ for $(i,j) \in A$ and $q = q_j, q_j+1, \ldots, Q-q_i$ by a similar argument, but it can be shown that they are already implied by constraints (3) and the flow balance constraints (2b).

2.3 A strengthened set partitioning formulation

The set partitioning formulation has been successfully applied to solve large-size CVRP instances and crew scheduling problems exactly due to its tight LP bound. Motivated by this, we propose a set partitioning formulation for the EMVRP based on q-routes and strengthened by additional constraints.

We define a q-route as a walk $i_0 i_1 i_2 \ldots i_k i_0$ with $i_1, i_2, \ldots, i_k \in V_0$, $i_0 = 0$ and $\sum_{l=1}^k q_{i_l} \leq Q$. We call a q-route *s-cycle-free* if it forbids any cycle of length at most s. For example, a feasible vehicle route, also called an elementary route, is an n-cycle-free q-route, and a q-route can be seen as a 1-cycle-free q-route. Let Ω_s be the set of all s-cycle-free q-routes for $s \geq 1$. Clearly $\Omega_n \subseteq \ldots \subseteq \Omega_2 \subseteq \Omega_1$, where Ω_1 is the set of all q-routes. Let a_{ir} denote the number of times that customer i is visited by a q-route r. We introduce a binary variable z_r for each q-route $r \in \Omega_s$ to indicate if it is chosen or not. Then the set partitioning formulation for the EMVRP is as follows.

$$\min \sum_{r\in\Omega_s} c_r z_r \tag{4a}$$

$$\text{s.t.} \sum_{r\in\Omega_s} a_{ir} z_r = 1, \ \forall i \in V_0 \tag{4b}$$

$$\sum_{r\in\Omega_s} z_r = K, \tag{4c}$$

$$z_r \geq 0, \ \forall r \in \Omega_s, \tag{4d}$$

$$z_r \in \mathbb{Z}, \ \forall r \in \Omega_s, \tag{4e}$$

where c_r is the cost, instead of the total length, of the q-route r as defined in the EMVRP. Constraints (4b) ensure that each customer is visited exactly once. The LP relaxation of (4) needs to be solved by column generation since there are an exponentially number of decision variables. Although formulation (4) is valid for any $s \geq 1$, there is a trade-off between the quality of the LP bound and the difficulty of solving the pricing problem for different s values. With larger s values, the set Ω_s contains less q-routes and the LP bound provided by (4) is tighter, but the pricing problem becomes significantly harder to solve when $s \geq 3$. We choose $s = 2$ for our formulation, namely the set partitioning formulation based on 2-cycle-free q-routes. More details on how to solve the pricing problem for the LP relaxation of (4) will be given in Section 4.

To further tighten the lower bound obtained from the set partitioning formulation, we can add additional constraints satisfied by all feasible solutions into (4). One way to obtain additional constraints in the (z_r)-space is through the transformation of valid inequalities in the space of other decision variables. For example, there are many families of well-known valid inequalities in the (x_{ij})-space derived for the CVRP, which are also valid for the EMVRP since their two-index one-commodity flow formulations share the same feasible set. When the (x_{ij})-variables and the (z_r)-variables both represent the same q-route, the coupling constraints between them can be defined as follows. Let a_{ijr} denote the number of times that a q-route r traverses arc (i,j), then $x_{ij} = \sum_{r\in\Omega_s} a_{ijr} z_r$ and any valid inequalities in the (x_{ij})-space can be transformed to a valid inequality in the (z_r)-space. We then obtain a strengthened set partitioning formulation in the

(x_{ij}, z_r)-space for the EMVRP.

$$\min \sum_{r\in\Omega_s} c_r z_r$$

$$\text{s.t.} \sum_{r\in\Omega_s} a_{ir} z_r = 1, \ \forall i \in V_0 \tag{5a}$$

$$\sum_{r\in\Omega_s} z_r = K, \tag{5b}$$

$$x_{ij} - \sum_{r\in\Omega_s} a_{ijr} z_r = 0, \ \forall (i,j) \in A, \tag{5c}$$

$$\sum_{(i,j)\in A} \alpha_{ij}^t x_{ij} \geq \beta^t, t \in \mathcal{T} \tag{5d}$$

$$z_r \geq 0, \ \forall r \in \Omega_s \tag{5e}$$

$$z_r \in \mathbb{Z}, \ \forall r \in \Omega_s. \tag{5f}$$

Constraints (5d) can be any valid inequalities in the (x_{ij})-space, such as the in-degree and out-degree constraints for each customer and rounded capacity constraints for the CVRP Fukasawa et al. (2006). In this paper, we add framed capacity, strengthened comb, multistar, partial multistar, generalized multistar and hypotour cuts, all described in detail in Lysgaard (2003); Fukasawa et al. (2006). In particular, we add the vehicle cardinality constraint $\sum_{i\in V_0} x_{i0} = \sum_{i\in V_0} x_{0i} = K$ so that constraint (5b) becomes redundant. By eliminating constraint (5b) and substituting variable x_{ij} in (5d) with equation (5c), we obtain the strengthened set partitioning formulation in the (z_r)-space.

$$\min \sum_{r\in\Omega_s} c_r z_r$$

$$\text{s.t.} \sum_{r\in\Omega_s} a_{ir} z_r = 1, \ \forall i \in V_0, \tag{6a}$$

$$\sum_{r\in\Omega_s} (\sum_{(i,j)\in A} \alpha_{ij}^t a_{ijr}) z_r \geq \beta^t, t \in \mathcal{T}, \tag{6b}$$

$$z_r \geq 0, \ \forall r \in \Omega \tag{6c}$$

$$z_r \in \mathbb{Z}, \ \forall r \in \Omega. \tag{6d}$$

We call the the LP relaxation of (6) defined by constraints (6a) to (6c) the Dantzig-Wolfe Master problem (DWM). The DWM has an exponential number of variables and constraints, which we will solve in Section 4 by both column generation and cut generation in a BCP framework.

3 A theoretical comparison of different formulations

In this section, we provide a theoretical comparison of the LP bounds given by the two-index one-commodity flow formulation and the two new formulations introduced in Section 2. Since each formulation for the EMVRP has the same feasible region as the CVRP, our comparison results can be applied to the CVRP analogously. Define the following polyhedral sets corresponding to the feasible regions of the LP relaxations for different formulations, respectively.

1. The two-index one-commodity flow formulation. Let

$$P_{1cf} = \{(x, f) \in \Re^{|A|\times|A|} \mid (x, f) \text{ satisfies constraints (1a) to (1f).}\}$$

2. The arc-load formulation. Let

$$P_{arcload} = \{x \in \Re^{|A|\times(Q+1)} \mid x \text{ satisfies constraints (2a) to (2d).}\}$$

3. The arc-load formulation with 2-cycle elimination constraints. Let

$$P_{arcload+2} = \{x \in \Re^{|A|\times(Q+1)} \mid x \text{ satisfies constraints (2a) to (2d) and (3).}\}$$

4. The set partitioning formulation with q-routes. Let

$$P_{qroute} = \{z \in \Re^{|\Omega_1|} \mid z \text{ satisfies constraints (4b) to (4d) with } s = 1.\}$$

5. The set partitioning formulation with 2-cycle-free q-routes. Let

$$P_{qroute+2} = \{z \in \Re^{|\Omega_2|} \mid z \text{ satisfies constraints (4b) to (4d) with } s = 2.\}$$

Let $Z_{1cf}, Z_{arcload}, Z_{arcload+2}, Z_{qroute}$ and $Z_{qroute+2}$ denote the optimal objective value of the corresponding LP relaxation, respectively.

Proposition 1. $Z_{1cf} \leq Z_{arcload}$.

Proof. Given any point $\hat{x} \in P_{arcload}$, we construct $x_{ij} = \sum_{q=0}^{Q} \hat{x}_{ij}^q$ and $f_{ij} = \sum_{q=0}^{Q} q \cdot \hat{x}_{ij}^q,\ \forall (i,j) \in A$. The constructed (x, f) yields the same objective value as $\hat{x}$, since:

$$\sum_{(i,j)\in A} d_{ij}(wx_{ij} + f_{ij}) = \sum_{(i,j)\in A} d_{ij}(w\sum_{q=0}^{Q}\hat{x}_{ij}^q + \sum_{q=0}^{Q}\hat{x}_{ij}^q q) = \sum_{(i,j)\in A}\sum_{q=0}^{Q} d_{ij}(w+q)\hat{x}_{ij}^q.$$

We next show that $(x, f) \in P_{1cf}$, which implies that $Z_{1cf} \leq Z_{arcload}$.

(1) The out-degree constraints (1a) are satisfied since $\sum_{j\in V\setminus\{i\}} x_{ij} = \sum_{j\in V\setminus\{i\}} \sum_{q=0}^{Q} \hat{x}_{ij}^q = 1,\ \forall i \in V_0$.

(2) The in-degree constraints (1b) are satisfied since $\forall i \in V_0$:

$$\sum_{j\in V\setminus\{i\}} x_{ji} = \sum_{j\in V\setminus\{i\}}\sum_{q=0}^{Q}\hat{x}_{ji}^q = \sum_{q=0}^{Q}\sum_{j\in V\setminus\{i\}}\hat{x}_{ji}^q = \sum_{q=0}^{Q}\sum_{j\in V\setminus\{i\}}\hat{x}_{ij}^{q+q_i}$$
$$= \sum_{j\in V\setminus\{i\}}\sum_{q=0}^{Q}\hat{x}_{ij}^{q+q_i} = \sum_{j\in V\setminus\{i\}}\sum_{q=0}^{Q}\hat{x}_{ij}^q = 1.$$

The third equation follows from the flow balance constraints (2b).

(3) The flow balance constraints (1c) are satisfied since $\forall i \in V_0$:

$$\begin{aligned}
\sum_{j\in V\setminus\{i\}} f_{ij} - \sum_{j\in V\setminus\{i\}} f_{ji} &= \sum_{j\in V\setminus\{i\}} \sum_{q=0}^{Q} q \cdot \hat{x}_{ij}^q - \sum_{j\in V\setminus\{i\}} \sum_{q=0}^{Q} q \cdot \hat{x}_{ji}^q \\
&= \sum_{q=0}^{Q} q \sum_{j\in V\setminus\{i\}} \hat{x}_{ij}^q - \sum_{q=0}^{Q} q \sum_{j\in V\setminus\{i\}} \hat{x}_{ji}^q \\
&= \sum_{q=0}^{Q} q \sum_{j\in V\setminus\{i\}} \hat{x}_{ji}^{q-q_i} - \sum_{q=0}^{Q} (q - q_i) \sum_{j\in V\setminus\{i\}} \hat{x}_{ji}^{q-q_i} \\
&= \sum_{q=0}^{Q} q_i \sum_{j\in V\setminus\{i\}} \hat{x}_{ji}^{q-q_i} = q_i \sum_{q=0}^{Q} \sum_{j\in V\setminus\{i\}} \hat{x}_{ji}^{q-q_i} \\
&= q_i \sum_{q=0}^{Q} \sum_{j\in V\setminus\{i\}} \hat{x}_{ij}^q = q_i.
\end{aligned}$$

At the third step $\sum_{q=0}^{Q} q \sum_{j\in V\setminus\{i\}} \hat{x}_{ji}^q = \sum_{q=0}^{Q}(q - q_i) \sum_{j\in V\setminus\{i\}} \hat{x}_{ji}^{q-q_i}$ since $\hat{x}_{ji}^q = 0$ for $q \leq q_j - 1$ and $q \geq Q - q_i + 1$.

(4) Constraints (1d) are satisfied since $\forall (i, j) \in A$:

$$f_{ij} = \sum_{q=0}^{Q} q \cdot \hat{x}_{ij}^q = \sum_{q=q_i}^{Q-q_j} q \cdot \hat{x}_{ij}^q \leq (Q - q_j) \sum_{q=q_i}^{Q-q_j} \hat{x}_{ij}^q = (Q - q_j)x_{ij}, \text{ and}$$

$$f_{ij} = \sum_{q=0}^{Q} q \cdot \hat{x}_{ij}^q = \sum_{q=q_i}^{Q-q_j} q \cdot \hat{x}_{ij}^q \geq q_i \sum_{q=q_i}^{Q-q_j} \hat{x}_{ij}^q = q_i x_{ij}.$$

(5) The vehicle cardinality constraint (1e) is satisfied since $\sum_{i\in V_0} x_{i0} = \sum_{i\in V_0} \sum_{q=0}^{Q} \hat{x}_{i0}^q = K$.

We also note that there exist instances for which the inequality $Z_{1cf} \leq Z_{arcload}$ is strict. □

The following result was mentioned in Pessoa et al. (2008) for the CVRP, and it also applies to the EMVRP.

Proposition 2. *For any point $\hat{x} \in P_{arcload}$, there exists a point $\hat{z} \in P_{qroute}$ with the same objective value, and vice versa. Therefore, $Z_{arcload} = Z_{qroute}$.*

We next show that the arc-load formulation with 2-cycle elimination constraints gives exactly the same LP bound as the set partitioning formulation based on 2-cycle-free q-routes.

Theorem 1. *Given any point $\hat{z} \in P_{qroute+2}$, there exists an $\hat{x} \in P_{arcload+2}$ with the same objective value, and vice versa. Therefore, $Z_{arcload+2} = Z_{qroute+2}$.*

Proof. We first show that given any $\hat{z} \in P_{qroute+2}$ there exists a feasible point $\hat{x} \in P_{arcload+2}$ with the same objective value. Let $a_{ijr}^q = 1$ indicate that the 2-cycle-free q-route r traverses arc (i, j) with load q and 0 otherwise. Let $\hat{x}_{ij}^q = \sum_{r\in\Omega_2} a_{ijr}^q \hat{z}_r$, $\forall (i, j) \in A, \forall q = 0, 1, \ldots, Q$. It is not difficult to show that $\hat{x}$ satisfies the out-degree constraints (2a), the flow balance constraints (2b) and the vehicle cardinality constraints (2c). The point $\hat{x}$ also satisfies the 2-cycle elimination constraints (3), since $\forall (i,j) \in A$ and $q = q_i, q_i+1, \ldots, Q-q_j$:

$$\begin{aligned}\sum_{k\in V\setminus\{i,j\}} \hat{x}_{ki}^{q-q_i} - \hat{x}_{ij}^{q} &= \sum_{k\in V\setminus\{i,j\}} \sum_{r\in\Omega_2} a_{kir}^{q-q_j} \hat{z}_r - \sum_{r\in\Omega_2} a_{ijr}^{q} \hat{z}_r \\ &= \sum_{r\in\Omega_2} (\sum_{k\in V\setminus\{i,j\}} a_{kir}^{q-q_i} - a_{ijr}^{q}) \hat{z}_r = 0,\end{aligned}$$

since route r is a 2-cycle-free q-route.

On the other hand, given any $\hat{x} \in P_{arcload+2}$, we construct a $\hat{z} \in P_{qroute+2}$ as follows. Select a nonzero component $\hat{x}_{i_1 j_1}^{q}$ of $\hat{x}$ with the load q being the largest. Then vertex j_1 must be the depot, since otherwise we can find another nonzero component $\hat{x}_{j_1 k}^{q'}$ of $\hat{x}$ with a larger load q' by the flow balance constraint (2b) for vertex j_1. Since $\hat{x}$ satisfies the 2-cycle elimination constraint $\sum_{k\in V\setminus\{i,0\}} \hat{x}_{ki_1}^{q-q_{i_1}} \geq \hat{x}_{i_1 0}^{q}$ and $\hat{x}_{i_1 0}^{q} > 0$, there exists a vertex $i_2 \neq 0$ such that $\hat{x}_{i_2 i_1}^{q-q_{i_1}} > 0$. Similarly by the 2-cycle elimination constraint $\sum_{k\in V\setminus\{i_1,i_2\}} \hat{x}_{ki_1}^{q-q_{i_1}-q_{i_2}} \geq \hat{x}_{i_2 i_1}^{q-q_{i_1}}$, there exists a vertex $i_3 \neq i_1$ such that $\hat{x}_{i_3 i_2}^{q-q_{i_1}-q_{i_2}} > 0$. If $i_3 = 0$, we stop. Otherwise repeat the previous process until we find a vertex $i_l = 0$. Then the fact that $\hat{x}_{0 i_{l-1}}^{q-q_{i_1}-\ldots-q_{i_{l-1}}} > 0$ implies that $q - q_{i_1} - \ldots - q_{i_{l-1}} = 0$ and the route $r_1 = \{0, i_{l-1}, \ldots, i_1, 0\}$ is a 2-cycle-free q-route. Let $\hat{z}_{r_1} = \min\{\hat{x}_{i_1 0}^{q}, \hat{x}_{i_2 i_1}^{q-q_{i_1}}, \ldots, \hat{x}_{0 i_{l-1}}^{0}\} > 0$ and $b_{r_1} = (b_{ijr_1}^{q})$ be the indicator vector of route r_1 in the arc-load space, where for $(i,j) \in A$, $q = 0, 1, \ldots, Q$, $b_{ijr_1}^{q} = 1$ indicates route r_1 traverses arc (i,j) with load q. Set $\hat{x}^1 = \hat{x} - \hat{z}_{r_1} b_{r_1}$. By the definition of $\hat{z}_{r_1}$, $\hat{x}^1 \geq 0$ and the number of nonzero elements in $\hat{x}^1$ is at least one less than that of $\hat{x}$. Moreover, vector $\hat{x}^1$ satisfies all the flow balance constraints and 2-cycle elimination constraints. If $\hat{x}^1 > 0$, we repeat the previous process to find another 2-cycle-free q-route until vector $\hat{x}^k = 0$ at some iteration k. Let $r_2, r_3, \ldots, r_m$ be the routes obtained and $\hat{z}_{r_2}, \hat{z}_{r_3}, \ldots, \hat{z}_{r_m}$ be the components in vector $\hat{z}$ constructed as $\hat{z}_{r_1}$. It is clear that these are distinct routes, since when one component $\hat{x}_{ij}^{q}$ becomes 0, that arc-load pair appears in the route found at that iteration will not show up in any of the routes at later iterations. Therefore, the number of routes m is no more than the number of nonzero components in $\hat{x}$. Finally, we set $\hat{z}_r = 0$ for all other 2-cycle-free q-routes $r \in \Omega_2$ that are not found by this procedure. It is clear that the constructed $\hat{z}$ is a feasible solution of $P_{qroute+2}$, and it has the same objective value as $\hat{x}$. □

The following proposition summarizes the results derived in this section.

Proposition 3. $Z_{1cf} \leq Z_{arcload} = Z_{qroute} \leq Z_{arcload+2} = Z_{qroute+2}$.

Once we add additional constraints in the (x_{ij})-space to strengthen the set partitioning formulation based on 2-cycle-free q-routes, as in (6), then the corresponding LP bound will be even tighter than $Z_{qroute+2}$. We develop a BCP algorithm to solve (6) in Section 4.

4 The branch-cut-and-price algorithm

We describe a BCP algorithm for the set partitioning formulation (6) based on 2-cycle-free q-routes and strengthened with additional constraints in the (x_{ij})-space. The BCP algorithm is based on a branch-and-bound tree, where at each node of the tree the DWM (the LP relaxation of (6)) is solved first by column generation and then by cut separation. Because of the difference of calculating the route cost between the EMVRP and the CVRP, we focus on describing how to solve the new pricing problem for the EMVRP. We also briefly mention other main ingredients of the algorithm, and refer the readers to Fukasawa et al. (2006) for further details of the BCP algorithm for solving CVRP.

4.1 Column generation and the pricing problem

Given a primal LP optimal solution for the restricted DWM, the pricing problem tries to find a new variable (a column) with a negative reduced cost if there exists one. For the CVRP, the pricing problem is to find a minimum cost (s-cycle-free) q-route over the original graph (V, A), where the cost over each arc is defined according to the optimal dual variables first and then the cost of a q-route is simply the sum of costs of arcs that constitute that q-route. This way of calculating the q-route cost cannot be applied to the EMVRP. Specifically, the cost over an arc in the EMVRP cannot be determined without knowing which q-route traverses that arc, since the cost depends on the load carried over that arc, namely the cumulative supply picked up before the arc is traversed. We will show below that the pricing problem can be formulated as a shortest q-route problem over a multigraph.

Let the optimal dual variables associated with constraints (6a) and (6b) be μ_i for $i \in V$ and π_t for $t \in \mathcal{T}$, respectively. For notational convenience, we also define a dual variable μ_0 for the depot, and set its value to 0. The pricing problem can be formulated as follows.

$$\min_{r \in \Omega_s} \bar{c}_r := c_r - \sum_{i \in V} \mu_i a_{ir} - \sum_{t \in \mathcal{T}} \pi_t \sum_{(i,j) \in A} \alpha_{ij}^t a_{ijr} \tag{7}$$

Proposition 4. *The pricing problem* (7) *can be reformulated as a shortest q-route problem over a multigraph with additional constraints.*

Proof. We first write down the calculation of c_r in (7) explicitly. Given a q-route r, let $a_{ijr}^q = 1$ denote that the q-route r traverses arc (i, j) with load q and 0 otherwise. The cost c_r in the EMVRP can be calculated as follows.

$$c_r = \sum_{(i,j) \in A} \sum_{q=0}^{Q} d_{ij}(w + q) a_{ijr}^q. \tag{8}$$

The term $\sum_{i \in V} \mu_i a_{ir}$ in (7) calculates the cumulative μ_i's on the q-route r. Note that whenever a customer $i \in V_0$ is visited by the q-route r, there are two arcs adjacent to customer i on that q-route. Then the cumulative μ_i's can be also computed through arcs by assigning each arc (i, j) a weight $\frac{\mu_i}{2} + \frac{\mu_j}{2}$. Therefore,

$$\sum_{i \in V} \mu_i a_{ir} = \sum_{(i,j) \in A} (\frac{\mu_i}{2} + \frac{\mu_j}{2}) a_{ijr} = \sum_{(i,j) \in A} \sum_{q=0}^{Q} (\frac{\mu_i}{2} + \frac{\mu_j}{2}) a_{ijr}^q, \tag{9}$$

where the second equation holds since $a_{ijr} = \sum_{q=0}^{Q} a_{ijr}^q$. Substitute (8) and (9) into (7), and the reduced cost of a q-route r can be calculated as

$$\begin{aligned} \bar{c}_r &= \sum_{(i,j) \in A} \sum_{q=0}^{Q} d_{ij}(w + q) a_{ijr}^q - \sum_{i \in V} \mu_i a_{ir} - \sum_{t \in \mathcal{T}} \pi_t \sum_{(i,j) \in A} \alpha_{ij}^t a_{ijr} \\ &= \sum_{(i,j) \in A} \sum_{q=0}^{Q} d_{ij}(w + q) a_{ijr}^q - \sum_{(i,j) \in A} \sum_{q=0}^{Q} (\frac{\mu_i}{2} + \frac{\mu_j}{2}) a_{ijr}^q - \sum_{t \in \mathcal{T}} \pi_t \sum_{(i,j) \in A} \alpha_{ij}^t \sum_{q=0}^{Q} a_{ijr}^q \\ &= \sum_{(i,j) \in A} \sum_{q=0}^{Q} [d_{ij}(w + q) - \frac{\mu_i}{2} - \frac{\mu_j}{2} - \sum_{t \in \mathcal{T}} \pi_t \alpha_{ij}^t] a_{ijr}^q \\ &= \sum_{(i,j) \in A} \sum_{q=0}^{Q} \bar{c}_{ij}^q a_{ijr}^q, \end{aligned} \tag{10}$$

where $\bar{c}^q_{ij} := d_{ij}(w+q) - \frac{\mu_i}{2} - \frac{\mu_j}{2} - \sum_{t\in\mathcal{T}} \pi_t \alpha^t_{ij}$ is defined for each pair of arc $(i,j) \in A$ and load $q = 0, 1, \ldots, Q$. Then the pricing problem (7) transforms to

$$\min_{r\in\Omega_s} \bar{c}_r = \sum_{(i,j)\in A} \sum_{q=0}^{Q} \bar{c}^q_{ij} a^q_{ijr}. \tag{11}$$

The pricing problem (11) can be seen as finding a shortest (s-cycle-free) q-route over a multigraph (V, A_Q) with additional constraints, constructed as follows. For the arc set A_Q, between each pair of vertices $i, j \in V_0$, there are $Q+1$ parallel arcs denoted by $(i,j)^q$ for $q = 0, \ldots, Q$, each of which has a length $\bar{c}^q_{ij}$. The additional constraints make sure that each feasible (s-cycle-free) q-route in (V, A_Q) corresponds to a (s-cycle-free) q-route in the original graph (V, A). Specifically, each feasible (s-cycle-free) q-route in (V, A_Q) starts from the depot 0, goes to some vertex i_1 through the arc $(0, i_1)^0$, then to vertex i_2 through the arc $(i_1, i_2)^{q_{i_1}}$, and so on before coming back to the depot 0 from some vertex i_l through the arc $(i_l, 0)^{q_{i_1}+\cdots+q_{i_l}}$. □

Based on Proposition 4, the shortest s-cycle-free q-route in (V, A_Q) with additional constraints can be solved by dynamic programming in pseudo-polynomial time for a given s. In this paper, we solve the pricing problem for 2-cycle-free q-routes. We first introduce some notation before presenting the dynamic programming algorithm. We call a walk $0 i_1 i_2 \ldots i_k$ in (V, A_Q) *feasible* only if it traverses arcs whose corresponding loads respect the load balance constraints: arc $(0, i_1)^0$, arc $(i_1, i_2)^{q_{i_1}}$, ..., and arc $(i_{k-1}, i_k)^{q_{i_1}+\cdots+q_{i_{k-1}}}$. Construct a $Q \times n$ matrix M, let each entry $M(q, v)$ denote the length of the shortest 2-cycle-free feasible walk that reaches v with cumulative load q, and let $\text{prec}(q, v)$ denote the last vertex before reaching v on that shortest 2-cycle-free feasible walk. Then $L_v = \min_{1\le q\le Q}\{M(q,v) + \bar{c}^q_{v0}\}$ is the length of the shortest 2-cycle-free feasible q-route whose last visit is vertex v before back to the depot. Let $q(v) = \arg\min_{1\le q\le Q}\{M(q,v) + \bar{c}^q_{v0}\}$ denote the cumulative load q over the shortest 2-cycle-free q-route whose last visit is v before back to the depot. For each $v \in V_0$, Algorithm 1 outputs a feasible 2-cycle-free q-route which attains L_v.

The 2-cycle-free q-routes with $L_v < 0$ are added as new columns to the DWM, retrieved backwards from vertex v, $\text{prec}(q(v), v)$, $\text{prec}(q(v) - q_v, \text{prec}(q(v), v))$, and so on. The pricing problem can be solved in $O(n^2 Q)$ time.

Note that although using s-cycle-free q-routes with $s \ge 3$ will give tighter LP bound, the pricing algorithm needs more bookkeeping and becomes significantly more complicated, similar to the case in the CVRP as shown by Irnich and Villeneuve (2006), so we choose $s = 2$ in this paper.

4.2 Cut generation

The valid inequalities we add include rounded capacity inequalities, framed capacity inequalities, strengthened comb inequalities, multistar inequalities, partial multistar inequalities, generalized multistar inequalities and hypotour inequalities, all described in detail in Lysgaard (2003); Fukasawa et al. (2006).

4.3 Branching rule

We use the same branching rule as in Fukasawa et al. (2006). The branching candidate set $S \subseteq V_0$ is chosen such that $2 < \sum_{i\in S}\sum_{j\notin S} x_{ij} + \sum_{i\in S}\sum_{j\notin S} x_{ji} < 4$ and we branch by imposing the disjunction $(\sum_{i\in S}\sum_{j\notin S} x_{ij} + \sum_{i\in S}\sum_{j\notin S} x_{ji} = 2) \vee (\sum_{i\in S}\sum_{j\notin S} x_{ij} + \sum_{i\in S}\sum_{j\notin S} x_{ji} \ge 4)$. We use strong branching to select which set to branch on.

Algorithm 1 The pricing algorithm for 2-cycle-free feasible q-routes in (V, A_Q)

Input: the length $\vec{c}_{ij}^q$ of arc $(i,j)^q$ for $(i,j) \in A$ and $q = 0, 1, \ldots, Q$, and supply q_i for $i \in V_0$.
Output: L_v, $q(v)$ and $\text{prec}(q, v)$ for $q = 1, \ldots, Q$ and $v \in V_0$.

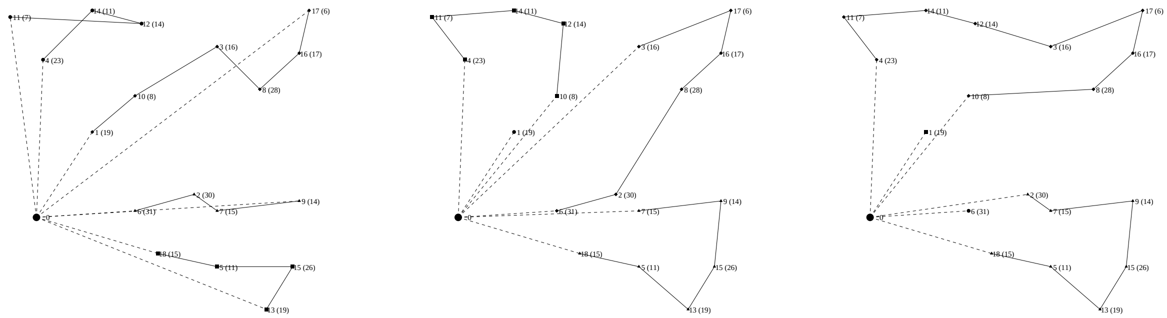
```
function PRICING(c̄^q_ij, q_i)
    Initialization: M(q,v) ← ∞, prec(q,v) ← ∅ for q = 1,...,Q and v = 1,...,n
    for v = 1 to n do
        M(q_v, v) ← c̄^0_0v
        prec(q_v, v) ← 0
    end for
    for q = 1 to Q do
        for v = 1 to n do
            for w = 1 to n with w ≠ v, prec(q,v) and q + q_w ≤ Q do
                if M(q + q_w, w) < M(q,v) + c̄^q_vw then
                    M(q + q_w, w) ← M(q,v) + c̄^q_vw
                    prec(q + q_w, w) ← v
                end if
            end for
        end for
    end for
    L_v ← min_{1≤q≤Q}{M(q,v) + c̄^q_v0} and q(v) ← arg min_{1≤q≤Q}{M(q,v) + c̄^q_v0} for v ∈ V_0
    return L_v, q(v) and prec(q,v) for q = 1,...,Q and v ∈ V_0
end function
```

4.4 Node selection and primal bounds

Our node selection strategy is depth first search. Since the focus of this paper is to develop exact solution methods, we do not exploit any primal heuristics. Instead, we solve the arc-load formulation (2) of the EMVRP by CPLEX under the default setting within one hour time limit, and use the best feasible solution obtained as the initial primal solution of our BCP algorithm. If no feasible solution is obtained from the arc-load formulation, then no primal bound is given. Good primal heuristics will certainly help improve the performance of the BCP algorithm.

5 Computational results

We tested our algorithms on a set of CVRP instances series A, B, E, V, and P, available at `www.branchandcut.org`, with a chosen vehicle curb weight. All tests were performed in a Linux workstation with eight 3.00 GHz processors (only one thread is used) and 7.8 Gb memory. The branch-cut-and-price algorithm is implemented in C++, with linear programs solved by CPLEX 12.5. Whenever CPLEX MIP solver was used for comparison purposes, its configuration was set to default. We perform two sets of experiments. In the first experiment, we test how the vehicle curb weight affects the optimal routing decision in the EMVRP. In the second experiment, we compare the computational performances of the proposed algorithms with the CPLEX MIP solver on a set of instances.



(a) Vehicle weight $w = 0$ (b) Vehicle weight $w = 0.5 \times Q$ (c) Vehicle weight $w = 30 \times Q$

Figure 1: Optimal vehicle routes with different vehicle curb weights

5.1 The impact of curb weight on the optimal routing decision

We first show how the vehicle curb weight w affects the optimal routing decision in the context of the EMVRP. We choose instance P-n19-k2 from the instance pool, which contains 19 vertices and a maximum of $k = 2$ vehicles. We relax the instance by setting the maximum number of vehicles allowed to $k = 4$ to better illustrate how the optimal routes change with the increase of vehicle curb weight. The optimal routes for three different vehicle curb weights $w = 0$, $w = 0.5 \times Q$ and $w = 30 \times Q$ are plotted in Figure 1. The pair of numbers $a(b)$ in Figure 1 denotes that the supply of customer a is b.

We see from Figure 1 that when vehicle curb weight $w = 0$, each route in the optimal solution is more balanced in terms of total supply picked up and the number of customers visited. Moreover, some route even contain a “crossover”, for example the subpath 11-12-14-4, which will never happen in the CVRP with arc costs satisfying the triangle inequalities. When vehicle curb weight $w = 0.5 \times Q$, routes in the optimal solution start to diverge in the total length and total supply picked up. At an extreme case, when vehicle curb weight $w = 30 \times Q$, two routes in the optimal solution consist of a single customer each. Moreover, this optimal solution is identical to the optimal solution in the CVRP. Figure 1 illustrates the bi-criteria nature of the EMVRP, where there is a trade-off between the cumulative weighted load and the total traveling distance in the objective function. Theoretically, as the curb weight w increases to infinity, the part of the total traveling distance dominates and the optimal solution of the EMVRP should converge to the optimal solution of the CVRP.

5.2 The computational performance of proposed algorithms

We now present the computational performance of the proposed algorithms on the A, B, E, V, and P instance sets. We set the vehicle curb weight $w = \rho Q$ and use the same ratio $\rho = 0.15$ for all instances as in Kara et al. (2007). We first solve the arc-load formulation (2) by the CPLEX MIP solver with a one hour time limit, and use the best feasible solution obtained as the initial primal solution for each of the three formulations introduced in Section 2. If no feasible solution is obtained within one hour, then no initial feasible solution is given. We then solved the two-index one-commodity flow formulation (1) and the arc-load formulation (2) by a branch-and-cut algorithm with the CPLEX MIP solver, and solved the strengthened set partitioning formulation (6) by the BCP algorithm described in Section 4. For the two-index one-commodity flow formulation (1), we generate cuts based on the cut separation package (Lysgaard, 2003) for the traditional CVRP problems. All valid inequalities for the CVRP remain valid for the EMVRP since they share the

Instances	Two-index			Arc-load			BCP	
	Time	Nodes	O/R-Gap	Time	Nodes	O/R-Gap	Time	O/R-Gap
P-n16-k8	0.0	1	0.1%	0.0	1	0.0%	0.0	0.0%
P-n22-k8	2.1	155	2.5%	0.2	1	0.5%	1.8	0.0%
P-n50-k7	-	35638	5.9%	184.9	1	0.1%	1.9	0.0%
P-n55-k7	-	32770	9.0%	6370.1	450	0.6%	300.4	0.5%
A-n32-k5	145.8	10530	3.0%	95.0	71	0.9%	5.5	0.3%
A-n37-k5	-	92500	3.3%	60.5	4	0.7%	2.7	0.0%
A-n45-k6	-	21400	12.6%	-	1130	5.8%	215.7	1.7%
A-n54-k7	-	22000	3.3%	3058.5	598	0.8%	109.0	0.4%
B-n31-k5	-	109200	0.6%	-	7204	1.8%	-	0.9%
B-n41-k6	2527.2	20168	3.6%	-	1870	4.5%	9.2	0.3%
B-n50-k8	-	23942	0.7%	-	1205	0.4%	34.7	0.1%
B-n57-k9	-	34400	0.9%	-	278	6.7%	316.5	0.2%
E-n13-k4	0.6	49	4.5%	0.1	1	0.7%	0.1	0.1%
E-n33-k4	-	43500	1.8%	-	0	1.4%	393.7	0.5%
E-n76-k7	-	19521	22.2%	-	1	40.3%	-	0.6%
gr-n48-k3	5.6	141	9.2%	5.6	1	2.6%	1.2	0.0%

Table 1: Computational performance of the branch-and-cut algorithm for the two-index one-commodity flow formulation (1), the arc-load formulation (2), and the BCP algorithm for the set partitioning formulation (6) on a set of test instances

same feasible region. For the arc-load formulation (2), we add the 2-cycle elimination constraints (3) as user cuts throughout the branch-and-bound tree using a cut callback routine. We use a time limit of two hours for all formulations that we test on.

We present in Table 1 the computational time, number of nodes processed in the branch-and-bound tree, and the optimality gap (when the instance is not solved to optimality within the time limit) or the root gap (when the instance is solved to optimality within the time limit) for the three formulations. These instances are chosen randomly from our pool of test instances. We use symbol "-" to denote the case when the time limit of two hours is exceeded. The column "O/R-Gap" shows the root gap (final gap) when the instance is (not) solved to optimality within the two-hour limit. We see that the two-index one-commodity flow formulation is not competitive even for modest-size instances, which was also observed in Bektaş and Laporte (2011). The main reason is that the LP relaxation bound of (1) is so weak that a large number of nodes needs to be processed to achieve optimality. The arc-load formulation yields a much tighter LP relaxation bound, but due to the large size of the formulation, significant computational effort is spent on each node for solving the large-size LP. The BCP algorithm for the strengthened set partitioning formulation, however, is able to solve most of these instances to optimality within 400 seconds, and yields a small optimality gap for instances that are not solved to optimality within the time limit. It is clear that the BCP algorithm outperforms the other two alternative approaches.

We next show the performance of the BCP algorithm over the entire pool of test instances in Table 2 to Table 5. We use symbol "*" to denote the case when no feasible solution is found within the time limit.

Table 2 gives results for the P instances. We see that most instances are solved to optimality within the time limit, except two largest instances P-n76-k4 and P-n76-k5. The largest instance

Instance	Time	Nodes	O-Gap	R-Gap	SB-Time	Col-Time	Cut-Time
P-n16-k8	0.0	1	-	0.0%	0.0	0.0	0.0
P-n19-k2	0.5	1	-	0.0%	0.0	0.5	0.0
P-n20-k2	3.9	5	-	0.9%	2.5	0.9	0.4
P-n21-k2	4.6	3	-	0.3%	3.5	0.7	0.5
P-n22-k2	0.8	1	-	0.0%	0.0	0.8	0.0
P-n22-k8	1.8	1	-	0.0%	0.0	0.0	1.7
P-n23-k8	1.0	3	-	0.8%	0.2	0.0	0.8
P-n40-k5	42.6	13	-	0.7%	37.6	3.8	1.0
P-n45-k5	8.3	5	-	0.1%	0.0	6.5	1.6
P-n50-k10	164.4	109	-	1.1%	148.5	8.9	5.1
P-n50-k7	1.9	1	-	0.0%	0.0	1.9	0.0
P-n50-k8	3331.0	1035	-	2.3%	3129.9	121.8	39.3
P-n51-k10	284.8	267	-	1.6%	227.3	38.8	13.7
P-n55-k10	48.2	45	-	0.4%	23.4	21.2	2.7
P-n55-k15	63.8	81	-	1.3%	28.4	9.4	24.1
P-n55-k7	300.4	35	-	0.5%	289.6	7.8	2.2
P-n55-k8	1438.5	195	-	1.1%	1413.6	17.6	2.1
P-n60-k10	392.0	89	-	0.8%	372.4	12.4	4.8
P-n60-k15	33.5	55	-	0.6%	16.6	10.3	5.8
P-n65-k10	5181.0	656	-	0.8%	4985.5	130.6	28.4
P-n70-k10	1703.8	189	-	1.2%	1614.8	56.2	21.4
P-n76-k4	-	11	*	*	-	436.9	5.4
P-n76-k5	-	21	1.1%	-	-	101.6	5.1

Table 2: Computational performance of the BCP algorithm for set partitioning formulation (6) on the P instances

that we can solve in this instance set has 70 customers. For the instances that are solved to optimality, we see that the number of nodes processed is very small, and the root optimality gap is close to 0. Most of the solution time is spent on strong branching, but in most cases, strong branching does help to keep the size of the branch-and-bound tree small.

Next, Table 3 gives results for the A instances. We see that most instances are solved to optimality within the time limit, except four relatively large instances. The largest instance that we can solve in this instance set has 69 customers. We observe similar results as the ones shown in Table 2.

In Table 4, we show results for the B instances. The performance of the BCP algorithm on this set of instances has more variability than the P instances and A instances. The optimality is not achieved within the time limit for the smallest instance B-n31-k5, although the final optimality gap is quite small. The largest instance that we can solve has 63 customers. For the instances that are solved within the time limit, the root LP bound is very tight, and the number of nodes processed is small.

Finally, we present in Table 5 the results for the E instances and the remaining instances in our test instance pool. The BCP algorithm can solve all instances that do not belong to the E instance set within one minute, using less than 15 branch-and-bound nodes. The E instances are much more challenging to solve, especially the larger ones. The largest instance that we can solve in this set has 76 customers.

Overall, we observe that the BCP algorithm yields significant improvements over the two-index one-commodity flow formulation and the arc-load formulation on all instance sets. It provides a promising way to obtain stronger lower bounds for large-size instances within a reasonable time limit.

Instance	Time	Nodes	O-Gap	R-Gap	SB-Time	Col-Time	Cut-Time
A-n32-k5	5.5	7	-	0.3%	3.0	1.6	0.9
A-n33-k5	0.6	1	-	0.0%	0.0	0.6	0.0
A-n33-k6	4.6	3	-	0.3%	2.8	0.5	1.2
A-n34-k5	1.5	1	-	0.0%	0.0	0.7	0.8
A-n36-k5	3.3	5	-	0.1%	0.0	2.3	1.0
A-n37-k5	2.7	3	-	0.0%	0.0	2.1	0.6
A-n37-k6	69.9	85	-	1.2%	42.6	24.0	1.9
A-n38-k5	172.6	63	-	1.3%	164.3	5.0	2.1
A-n39-k5	27.9	17	-	0.5%	19.1	7.2	1.3
A-n44-k6	1.1	1	-	0.0%	0.0	1.1	0.0
A-n45-k6	215.7	61	-	1.7%	198.5	9.6	5.9
A-n45-k7	51.4	57	-	0.6%	26.3	21.1	2.9
A-n46-k7	3.9	3	-	0.1%	0.0	2.4	1.5
A-n48-k7	556.0	357	-	1.1%	360.3	182.5	5.4
A-n53-k7	2402.9	271	-	1.1%	2336.1	36.1	18.6
A-n54-k7	109.0	75	-	0.4%	32.2	67.0	7.8
A-n55-k9	838.3	203	-	0.7%	803.1	22.0	6.7
A-n60-k9	-	742	1.1%	-	7034.4	101.3	24.4
A-n61-k9	-	1073	2.5%	-	6849.7	233.7	61.3
A-n62-k8	66.2	29	-	0.2%	0.2	57.2	8.1
A-n63-k9	234.4	39	-	0.3%	181.1	42.4	9.2
A-n63-k10	2350.3	393	-	0.9%	2262.8	55.7	15.0
A-n64-k9	-	143	2.2%	-	7167.5	56.1	11.2
A-n65-k9	373.4	35	-	0.3%	353.9	12.6	5.2
A-n69-k9	960.9	85	-	0.8%	916.6	28.9	10.9
A-n80-k10	-	24	*	*	-	26.7	11.0

Table 3: Computational performance of the BCP algorithm for set partitioning formulation (6) on the A instances

Instance	Time	Nodes	O-Gap	R-Gap	SB-Time	Col-Time	Cut-Time
B-n31-k5	-	3378	0.9%	-	6921.1	191.7	12.5
B-n34-k5	34.4	25	-	0.5%	23.6	6.8	3.7
B-n35-k5	-	1490	4.3%	-	7073.3	68.5	4.4
B-n38-k6	5.0	11	0.2%	-	0.0	3.5	1.3
B-n39-k5	737.5	381	-	1.7%	571.4	152.8	4.4
B-n41-k6	9.2	3	-	0.3%	6.0	1.6	1.5
B-n43-k6	209.1	71	-	1.0%	200.3	5.8	1.8
B-n44-k7	40.0	15	-	0.2%	35.5	1.9	2.4
B-n45-k5	2598.7	407	-	1.0%	2538.7	38.5	6.3
B-n45-k6	-	531	11.2%	-	7060.3	95.8	9.2
B-n50-k8	34.7	7	-	0.1%	29.9	2.2	2.5
B-n51-k7	-	198	12.9%	-	7095.9	103.4	7.6
B-n52-k7	-	1051	0.6%	-	7044.4	104.6	12.6
B-n56-k7	124.3	49	-	0.2%	53.8	66.6	2.8
B-n57-k7	-	67	7.1%	-	7131.4	72.5	19.7
B-n57-k9	316.5	65	-	0.2%	302.1	9.0	3.7
B-n63-k10	688.4	121	-	0.4%	655.9	20.4	7.3
B-n64-k9	-	164	5.6%	-	7134.5	113.0	19.3
B-n66-k9	-	222	0.9%	-	7169.3	52.2	9.1
B-n67-k10	-	65	6.6%	-	7155.6	94.3	22.2
B-n68-k9	-	234	1.0%	-	7058.4	118.2	12.3

Table 4: Computational performance of the BCP algorithm for set partitioning formulation (6) on the B instances

Instance	Time	Nodes	O-Gap	R-Gap	SB-Time	Col-Time	Cut-Time
E-n13-k4	0.1	3	-	0.1%	0.0	0.0	0.1
E-n22-k4	0.1	1	-	0.0%	0.0	0.1	0.0
E-n23-k3	13.9	1	-	0.0%	0.0	13.8	0.0
E-n30-k3	-	909	6.2%	-	7062.8	90.5	5.9
E-n31-k7	0.7	1	-	0.0%	0.0	0.6	0.0
E-n33-k4	393.7	25	-	0.5%	328.2	62.7	0.9
E-n51-k5	-	102	51.3%	-	7098.7	142.1	7.9
E-n76-k10	3038.1	245	-	1.0%	2919.7	81.7	19.9
E-n76-k7	-	128	0.6%	-	7116.9	71.8	5.5
E-n76-k8	-	24	4.0%	-	-	47.6	5.9
E-n101-k8	-	14	*	*	-	126.7	16.0
E-n101-k14	-	31	3.1%	-	-	93.6	36.7
att-n48-k4	41.5	13	-	1.2%	36.6	4.0	0.8
bayg-n29-k4	2.2	9	-	0.6%	0.9	0.5	0.8
bays-n29-k5	0.0	1	-	0.0%	0.0	0.0	0.0
dantzig-n42-k4	7.3	3	-	0.5%	4.1	0.7	2.4
fri-n26-k3	0.1	1	-	0.0%	0.0	0.1	0.0
gr-n17-k3	0.0	1	-	0.0%	0.0	0.0	0.0
gr-n21-k3	1.3	5	-	3.3%	0.8	0.0	0.5
gr-n24-k4	2.5	13	-	4.8%	2.0	0.1	0.3
gr-n48-k3	1.2	1	-	0.0%	0.0	1.1	0.1
hk-n48-k4	0.7	1	-	0.0%	0.0	0.7	0.0
swiss-n42-k5	15.0	11	-	1.0%	13.6	0.3	1.1

Table 5: Computational performance of the BCP algorithm for set partitioning formulation (6) on the E and other instances

6 Conclusions

We introduce two new MILP formulations for the EMVRP and made a theoretical comparison on the LP bounds provided by all the formulations. We propose a new BCP algorithm to solve the set partitioning formulation based on 2-cycle-free q-routes for the EMVRP. Computational results demonstrate that the BCP algorithm can solve instances of much larger sizes to optimality than the ones reported in the literature.

We consider our proposed models and algorithm a first step towards solving other green transportation models. For example, vehicle speed is another major factor that affects energy consumption and emissions, and it is included as a decision variable in several recently proposed transportation models such as the pollution routing problem (Bektaş and Laporte, 2011). In future research, we will investigate how to adapt our approach to handle green transportation models with more decision variables and side constraints.

References

Hernán Abeledo, Ricardo Fukasawa, Artur Pessoa, and Eduardo Uchoa. The time dependent traveling salesman problem: polyhedra and algorithm. *Mathematical Programming Computation*, 5(1):27–55, 2013.

Foto Afrati, Stavros Cosmadakis, Christos Papadimitriou, George Papageorgiou, and Nadia Papakostantinou. The complexity of the travelling repairman problem. *Theoretical Informatics and Applications*, 20(1):79–87, 1986.

Roberto Baldacci and Aristide Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380, 2009.

Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations research*, 59(5):1269–1283, 2011.

Tolga Bektaş and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

Lucio Bianco, Aristide Mingozzi, and Salvatore Ricciardelli. The traveling salesman problem with cumulative costs. *Networks*, 23(2):81–91, 1993.

Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 163–171. ACM, 1994.

Claudio Contardo and Rafael Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014.

Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014.

Jittat Fakcharoenphol, Chris Harrelson, and Satish Rao. The k-traveling repairman problem. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 655–664. SIAM, 2003.

Matteo Fischetti, Gilbert Laporte, and Silvano Martello. The delivery man problem and cumulative matroids. *Operations Research*, 41(6):1055–1064, 1993.

Kenneth R Fox, Bezalel Gavish, and Stephen C Graves. Technical notean n-constraint formulation of the (time-dependent) traveling salesman problem. *Operations Research*, 28(4):1018–1021, 1980.

Ricardo Fukasawa, Humberto Longo, Jens Lysgaard, Marcus Poggi de Aragão, Marcelo Reis, Eduardo Uchoa, and Renato F Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming*, 106(3):491–511, 2006.

Daya Ram Gaur, Apurva Mudgal, and Rishi Ranjan Singh. Routing vehicles to minimize fuel consumption. *Operations Research Letters*, 41(6):576–580, 2013.

Bruce Golden, Subramanian Raghavan, and Edward Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43. Springer, 2008.

Luis Gouveia. A 2n constraint formulation for the capacitated minimal spanning tree problem. *Operations Research*, 43(1):130–141, 1995.

Stefan Irnich and Daniel Villeneuve. The shortest-path problem with resource constraints and k-cycle elimination for k 3. *INFORMS Journal on Computing*, 18(3):391–406, 2006.

İmdat Kara, Bahar Y Kara, and M Kadri Yetiş. Energy minimizing vehicle routing problem. In *Combinatorial Optimization and Applications*, pages 62–71. Springer, 2007.

İmdat Kara, Bahar Y Kara, and M Kadri Yetiş. Cumulative vehicle routing problems. In Tonci Caric and Hrvoje Gold, editors, *Vehicle Routing Problem*, pages 85–98. In-Teh, 2008.

Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

Adam N Letchford and Juan-José Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105(2-3):251–274, 2006.

Jens Lysgaard. Cvrpsep: A package of separation routines for the capacitated vehicle routing problem. Working paper 03-04, Department of Management Science and Logistics, Aarhus School of Business, 2003. `www.hha.dk/~lys`.

Jens Lysgaard and Sanne Wøhlk. A branch-and-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *European Journal of Operational Research*, 236(3):800–810, 2014.

Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. In Jon Lee and Jens Vygen, editors, *Proceedings of the 17th Conference on Integer Programming and Combinatorial Optimization*, pages 393–403. Springer, 2014.

Artur Pessoa, Marcus Poggi de Aragao, and Eduardo Uchoa. Robust branch-cut-and-price algorithms for vehicle routing problems. In Golden et al. (2008), pages 297–325.

Jean-Claude Picard and Maurice Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1): 86–110, 1978.

Paolo Toth and Daniele Vigo. *The Vehicle Routing Problem*. SIAM, 2001.

Eduardo Uchoa. Cuts over extended formulations by flow discretization. In A Ridha Mahjoub, editor, *Progress in Combinatorial Optimization*, pages 255–282. Wiley, 2011.

Eduardo Uchoa, Ricardo Fukasawa, Jens Lysgaard, Artur Pessoa, Marcus Poggi De Aragão, and Diogo Andrade. Robust branch-cut-and-price for the capacitated minimum spanning tree problem over a large extended formulation. *Mathematical Programming*, 112(2):443–472, 2008.

Yiyong Xiao, Qiuhong Zhao, Ikou Kaku, and Yuchun Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39(7):1419–1431, 2012.