# A symmetry-free polynomial formulation of the Capacitated Vehicle Routing Problem

S. L. Gadegaard*, J. Lysgaard

*Department of Economics and Business Economics, Aarhus University, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark*

*Discrete Applied Mathematics, Vol. 296, pp. 179-192, June 2021.*

**Abstract**

In this paper we propose a new polynomially sized formulation of the well known symmetric capacitated vehicle routing problem. Formulations of polynomial size have already been published in the academic literature for this problem, but they all possess the feature that they contain many equivalent solutions. As such, the optimal set of routes will be represented by several equivalent integer feasible solutions to the formulation, potentially leading to excessive computation times. The equivalence between solutions results from the possibility of reversing the order of visit on any route, starting and ending at the depot, without affecting feasibility or route length. In contrast, the formulation proposed in this paper eliminates the existence of equivalent integer solutions. In particular, instead of describing a route as a path starting and ending at the depot, we represent a route as two paths originating from the depot and ending at a socalled peak customer on the route. Moreover, in our formulation there is only one possible peak customer for any such two paths, resulting in a unique representation of any route. Our formulation has shown very competitive computing times compared to a classical formulation of comparable size. Consequently, our formulation can be recommended in combination with the use of algebraic modelling languages for entering a formulation in its entirety into a mixed-integer linear programming solver.

*Keywords:* vehicle routing, integer programming, symmetry breaking, polynomial formulation
*2010 MSC:* 90B06, 90C10, 90C90

*Corresponding author
*Email addresses:* sgadegaard@econ.au.dk (S. L. Gadegaard), lys@econ.au.dk (J. Lysgaard)

## 1. Introduction

The Capacitated Vehicle Routing Problem (CVRP) can be defined as follows (see, e.g., [1, 2]). Let $G = (V, E)$ be an undirected graph with node set $V = \{0, \ldots, n\}$ and edge set $E$. Node 0 represents a depot, whereas each of the nodes in $V_c = \{1, \ldots, n\}$ represents a customer. The symmetric cost of travel between nodes $i$ and $j$ is denoted by $c_{ij}$. A number $K$ of identical vehicles, each of capacity $Q > 0$, is available. Each customer $i \in V_c$ has an integer demand $q_i$, with $0 < q_i \leq Q$, which must be delivered from the depot. Each customer must be served by a single vehicle and no vehicle can serve a set of customers whose demand exceeds its capacity. The task is to find a set of vehicle routes of minimum cost, where each vehicle used leaves from and returns to the depot. The number of routes may or may not be predetermined. In particular, $K$ may be set equal to $n$ if the number of routes is unrestricted, but in many cases, $K$ is set equal to the minimum possible. The material in this paper does not require a particular setting of $K$, but in our computational experiments we shall fix $K$ to the minimum possible for the purpose of comparison with known optimal solutions to certain CVRP instances.

Mathematical formulations of the CVRP can be divided into two categories, *exponential formulations* and *polynomial formulations*, respectively, depending on how the size of the formulation grows with the number of customers in the problem.

Exponential formulations are characterized by an exponential growth of the formulation size as a function of $n$, either by having an exponential number of constraints or an exponential number of decision variables, or both. This exponential behaviour stems from the fact that the formulation contains either constraints or decision variables which are defined for exponentially many subsets of customers. For example, the *two-index formulation* presented in [3] contains $\mathcal{O}(2^n)$ capacity inequalities which are defined for almost all subsets of customers. This obviously poses some challenges with respect to implementation, typically leading to algorithms where constraints are included in the linear programming (LP) relaxation only if and when necessary, as in *branch-and-cut* algorithms. Alternatively, as in [4], the CVRP may be formulated as a Set Partitioning Problem (SPP) where decision variables represent feasible routes. As there in general will be exponentially many feasible routes, this again poses implementational challenges, typically leading to algorithms where decision variables (columns) are generated dynamically in the LP relaxation, as in *branch-and-price* algorithms. Moreover, these two types of formulations may be combined into formulations with exponentially many constraints as well as exponentially many decision variables (see, e.g., [5]), leading to the use of *branch-and-cut-and-price* algorithms. We refer to [6], [7] and [8] for many more details on these formulations and related algorithms.

Polynomial formulations, on the other hand, are characterized by only a polynomial number of constraints as well as only a polynomial number of decision variables as a function of $n$. In [9], a one-commodity flow formulation of the directed CVRP is introduced, and subsequent publications on this and related formulations include [10] and [11]. To our knowledge, all published polynomial formulations build on the linear assignment problem, i.e., the formulations are directed, involve one arrival at and one departure from each customer, and require that all routes begin and end at the depot. Hence, when applied to a symmetric problem, a directed formulation potentially contains exponentially many equivalent solutions which can be obtained by reversing the order of visit on the individual routes. Such symmetry between solutions may lead to excessive computing times.

This paper proposes a polynomial directed formulation of the symmetric CVRP in which the mentioned difficulties of symmetry are avoided. The main contributions of this paper are the following:

- We propose a symmetry-free polynomially sized formulation of the symmetric CVRP;
- We propose a new set of inequalities, of polynomial cardinality, denoted Rounded Peak Count Inequalities, which significantly improves the linear programming bound;
- We show, through computational studies, that the new formulation produces better linear programming lower bounds and leads to significantly smaller computation times than a classical formulation of similar size proposed in the literature;
- A restricted version of our formulation is linked to the Rounded Capacity Inequalities known from the undirected vehicle flow formulation;
- Given that our formulation possesses the two attractive characteristics of being limited in size as well as showing competitive performance, we find it reasonable to recommend our formulation for applications where it is desired to enter a formulation in its entirety into a mixed-integer linear programming solver. This avoids the considerable implementational complexities of dynamically generating rows or columns during the solution process. More generally, considering the several advantages of algebraic modelling languages (see, e.g., [12]), it is indeed desirable to be able to formulate the entire model using such a modelling language, and our formulation with its limited size and competitive performance seems relevant to consider for this purpose.

The paper is organized as follows. In Section 2 we present the mathematical models. In Section 5 we present our computational experiments. In Section 6 we provide our conclusions and discuss perspectives for further research.

## 2. Mathematical model

In this section, we first present in Subsection 2.1 a classical formulation which we regard as a key reference among polynomial formulations of the CVRP. This formulation is then used as a starting point for our further developments in subsequent subsections, where we in Subsection 2.2 present the fundamental idea behind our formulation. Then, in Subsection 2.3 we propose our peak based formulation which ensures uniqueness of any feasible integer solution.

### *2.1. A classical flow formulation*

Gavish and Graves introduced in [9] a one-commodity flow formulation of the directed CVRP. For the purpose of describing this formulation, we will need to introduce further notation. At the beginning of this paper we defined the CVRP on an undirected graph $G = (V, E)$. To this graph we now add the set $A$ of directed arcs for the purpose of representing directed formulations. Specifically, for any edge $\{i, j\} \in E$ we have two opposite arcs $(i, j) \in A$ and $(j, i) \in A$. Moreover, we use $A_c$ to denote the subset of arcs connecting two customers, i.e., $A_c = \{(i, j) \in A \mid i, j \in V_c\}$. For each arc $(i, j) \in A$, the cost $c_{ij}$ of travel is the same as for the corresponding edge in $E$. The formulation uses the following variables ([9]).

- For each $(i, j) \in A$, $x_{ij} = 1$ if a vehicle travels along arc $(i, j)$, otherwise $x_{ij} = 0$.
- For each $(i, j) \in A$, if a vehicle travels along arc $(i, j)$, then $f_{ij}$ denotes the total quantity delivered on the route when the vehicle leaves vertex $i$, otherwise $f_{ij} = 0$. Accordingly, we set $f_{0j} = 0\ \forall j \in V_c$. In addition, we set $q_0 = 0$.

For notational convenience, we define $x(A') = \sum_{(i,j)\in A'} x_{ij}$ for any arc set $A' \subseteq A$, and $f(A')$ is defined similarly. Furthermore, we make the following definitions for any $S \subset V$:

$$\begin{aligned}\delta^+(S) &= \{(i, j) \in A \mid i \in S, j \in V \setminus S\},\\ \delta^-(S) &= \{(i, j) \in A \mid i \in V \setminus S, j \in S\},\end{aligned}$$

where for further notational simplicity we write $\delta^+(i)$ and $\delta^-(i)$ instead of $\delta^+(\{i\})$ and $\delta^-(\{i\})$, respectively.

The formulation, which we will refer to as the *classical flow formulation* (CFF), is then as follows.

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.: } x(\delta^+(i)) = 1 \qquad \forall i \in V_c \tag{2}$$

$$x(\delta^-(i)) = 1 \qquad \forall i \in V_c \tag{3}$$

$$x(\delta^+(0)) = K \tag{4}$$

$$x(\delta^-(0)) = K \tag{5}$$

$$f(\delta^+(i)) = f(\delta^-(i)) + q_i \qquad \forall i \in V_c \tag{6}$$

$$q_i x_{ij} \le f_{ij} \le (Q - q_j) x_{ij} \qquad \forall (i, j) \in A \tag{7}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A \tag{8}$$

The objective in (1) expresses the total travel cost. The degree equations (2) ensure one departure from each customer, and (3) ensure one arrival to each customer. Similarly, the degree equations (4) and (5) ensure one departure from and arrival to the depot for each of the $K$ routes. Constraints (6) are flow conservation constraints ensuring the proper accumulation of demands given by the $f$-variables along a route. Constraints (7) establish bounds on the $f$-variables, and (8) are integrality constraints for the $x$-variables.

In the CFF, each route is described as a directed path starting and ending at the depot. Thus, on each individual route, the order of visit may be reversed without changing cost or feasibility given the symmetry of costs in the CVRP. Clearly, reversing the order of visit on a route means that the route will be represented by a different set of 1-valued $x_{ij}$-variables, provided that the route contains more than one customer. Given that such a reversal can be done on each individual route, any feasible solution with $r$ routes having more than one customer is represented in the CFF by as many as $2^r$ different integer solutions. For problem instances where $r$ is not very small, this leads to an overwhelming number of equivalent optimal solutions that must be examined implicitly or explicitly by an integer linear programming solver.

### *2.2. The fundamental idea and earlier work*

When choosing between mathematical formulations for obtaining proven optimal solutions to the CVRP, a number of aspects come into consideration. Exponential formulations typically lead to much faster algorithms, partly due to stronger linear programming relaxations than those obtained from polynomial formulations. On the other hand, the effort required for developing the associated algorithms is typically much greater for exponential formulations than for polynomial formulations. Indeed, polynomial formulations are more convenient as the entire formulation can be entered all at once in mathematical programming software.

In this paper we wish to combine the best of the two classes of formulations, i.e., to obtain a computationally efficient formulation which is of only polynomial size and therefore relatively easy to manage by mathematical programming software. For this purpose, we eliminate the mentioned symmetry which occurs in the CFF by representing any undirected route by two directed paths, and we develop a mathematical formulation which builds on this representation.

The idea of representing an undirected route by two directed paths has been used before in the literature. In [13], a description is proposed of the symmetric Traveling Salesman Problem (STSP) as two directed paths with common starting and ending nodes, and so that any other node is contained on one of the two paths. Given the symmetric costs, such a pair of paths can be converted into a Hamiltonian circuit by reversing one of the two paths, and the total cost of this circuit equals the total cost of the two paths. Finally, disregarding the direction of the circuit provides an undirected solution for the STSP, as desired. In [14], this representation was used for each route in the symmetric CVRP, where a number of socalled *special cities* were determined heuristically for the CVRP. Then, any vehicle route is represented by two paths from the depot to one of the

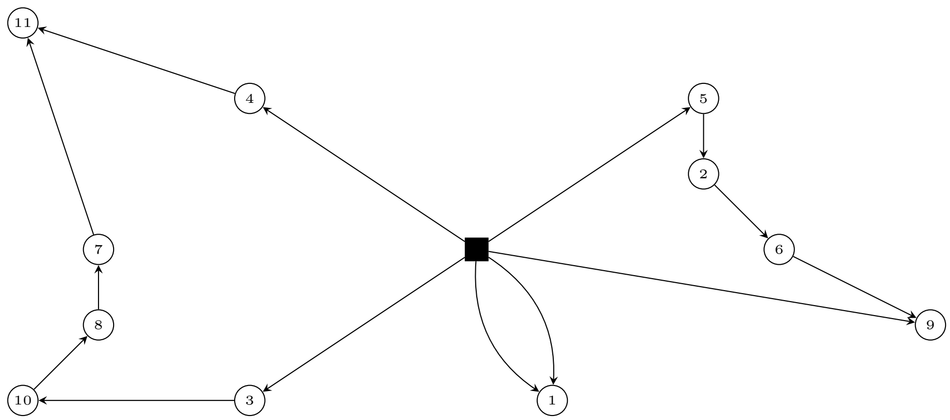



Figure 1: Structure of a solution obtained by our formulation

special cities. As in [13], the algorithm in [14] was based on a linear assignment model which resulted in the possibility of subtours as well as violated capacity inequalities. The special cities in [14] may in themselves be seen as being similar to what we will introduce as *peak* customers, but unlike the heuristic choice of special cities in [14], our approach differs significantly from [14] in the sense that the identification of our peak customers is part of the decisions made in our model and therefore guaranteed to be part of an optimal solution.

In a different but related context, a bi-directional search using dynamic programming was proposed in [15] for obtaining routes by joining forward and backward paths, which was found to be more efficient than a more traditional dynamic programming algorithm.

### *2.3. Our formulation*

We define a route as a sequence of nodes $(0, i_1, i_2, \ldots, i_k, 0)$ containing $k$ distinct customers. As a key element in our formulation, we introduce the concept of a *peak customer*, which is defined as the customer with the largest index among all customers on a route. Formally, the definition is as follows.

**Definition 1** (Peak customer)**.** The *peak customer* on any route $(0, i_1, i_2, \ldots, i_k, 0)$ is the customer $i^* = \max\{i_1, \ldots, i_k\}$.

The fundamental idea in our formulation is that we describe a route as being composed of two paths from the depot to the route's peak customer, and so that the two paths are node disjoint except for their common starting and ending points. Figure 1 illustrates how our formulation represents a solution. We will use this example to emphasize the main points in our formulation.

The solution in Figure 1 contains three routes $(0, 3, 10, 8, 7, 11, 4, 0)$, $(0, 5, 2, 6, 9, 0)$, and $(0, 1, 0)$. We note that customer 11 is the customer with the largest index on the first route, hence customer 11 is the peak customer on this route. Consequently, we represent this route by the two directed paths $(0, 3, 10, 8, 7, 11)$ and

$(0, 4, 11)$, as indicated by the arrows on the arcs in the figure. Similarly, the second route is represented by the two directed paths $(0, 5, 2, 6, 9)$ and $(0, 9)$, where customer 9 is the peak customer. Finally, the third route illustrates the possibility of a single-customer route, which is represented by the two parallel arcs from the depot to customer 1, which is the only and therefore also the peak customer on the route.

The essential characteristic of our representation is that any set of routes can be represented in only one way due to the definition of the peak customer. Therefore, for our formulation, the computational effort required in a branch and bound algorithm involves only the identification of a single integer solution for any set of undirected routes describing a feasible solution. Potentially, this may lead to a significant reduction in the overall computional effort.

For our formulation we introduce the following additional decision variables.

- For each $i \in V_c \setminus \{n\}$, the variable $u_i$ denotes the largest customer index visited on the path from the depot to and including customer $i$. We formulate our model without a variable $u_n$ which could have been fixed to $u_n = n$ if it existed.
- For each $i \in V_c$, the variable $p_i$ denotes whether or not node $i$ is a peak customer. Specifically, $p_i = 1$ if node $i$ is a peak customer and $p_i = 0$ otherwise, where we have fixed $p_n = 1$. Any customer which is not a peak customer will be called a *non-peak* customer in the following.
- For each $i \in V_c$, if $p_i = 1$, then $t_i$ represents the total demand on the route which services node $i$, otherwise $t_i = 0$.

We then obtain the following formulation which we refer to as our *basic peak* *formulation* (BPF).

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{9}$$

$$\text{s.t.: } x(\delta^+(i)) + p_i = 1 \qquad \forall i \in V_c \tag{10}$$

$$x(\delta^-(i)) - p_i = 1 \qquad \forall i \in V_c \tag{11}$$

$$t_i + f(\delta^+(i)) = f(\delta^-(i)) + q_i \qquad \forall i \in V_c \tag{12}$$

$$q_i x_{ij} \leq f_{ij} \leq (Q - q_j) x_{ij} \qquad \forall (i,j) \in A_c \tag{13}$$

$$q_i p_i \leq t_i \leq Q p_i \qquad \forall i \in V_c \tag{14}$$

$$x(\delta^+(0)) = 2K \tag{15}$$

$$\sum_{i \in V_c} p_i = K \tag{16}$$

$$i \leq u_i \leq i p_i + (n-1)(1-p_i) \qquad \forall i \in V_c \setminus \{n\} \tag{17}$$

$$u_i - u_j + (n-j-1) x_{ij} \leq n-j-1 \qquad \forall i, j \in V_c \setminus \{n\},\ i \neq j \tag{18}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A_c \tag{19}$$

$$x_{0j} \in \{0,1,2\} \qquad \forall j \in V_c \tag{20}$$

$$p_i \in \{0,1\} \qquad \forall i \in V_c \tag{21}$$

The objective in (9) expresses the total travel cost. The degree equations (10) ensure one departure from each non-peak customer and that there are no departures from any peak customer. The degree equations (11) ensure one arrival at each non-peak customer and two arrivals at each peak customer.

Constraints (12) ensure the proper accumulation of demands given by the $f$-variables along a path. At each peak customer, the total accumulated demand on the route is given by the value of $t_i$, which is composed of the demands on the two paths ending at this peak customer together with the demand of the peak customer itself. Constraints (13) establish bounds on the $f$-variables.

Constraints (14) ensure that $t_i$ is zero for any non-peak customer and that $t_i$ for each peak customer does not exceed the vehicle capacity. Moreover, constraints (14) also implies $q_i$ as a lower bound on $t_i$ for each peak customer.

For each of the $K$ routes, there must be two departures from the depot as well as one peak customer. Acccordingly, constraints (15) express that the total flow out of the depot must equal $2K$, and constraints (16) express that the total number of peak customers must equal $K$.

Constraints (17)–(18) are required to ensure that peaks are identified according to Definition 1, so that the model contains only one representation of any route with more than one customer, as described in more detail in Proposition 1. Inequalities (17) ensure, for each customer $i$, that $u_i = i$ if $p_i = 1$ and that $i \leq u_i \leq n-1$ if $p_i = 0$, considering that $n-1$ is the largest possible index for a non-peak customer. We note that inequality (17) for $i = n-1$ implies $u_{n-1} = n-1$. Inequalities (18) ensure that $u$-values are nondecreasing along any path. Specifically, $x_{ij} = 1$ implies $u_j \geq u_i$ and $x_{ij} = 0$ implies

$u_i - u_j \leq (n-1) - j$, where the implication of $x_{ij} = 0$ is derived from $u_i \leq n-1$ and $u_j \geq j$.

Finally, (19)–(21) are integrality constraints.

We now wish to verify that BPF serves its purpose of providing unique representations of undirected CVRP solutions. Hence, we present propositions 1 and 2.

**Proposition 1.** *Our BPF* (9)–(21) *ensures, for each customer $i$, that $p_i = 1$ if and only if customer $i$ is the peak customer on its route.*

*Proof.* Proof of Proposition 1. If $p_i = 1$, then it is implied by (17) that $u_i = i$ and by (10)–(11) that there are two paths ending at customer $i$. For any customer $j$ on one of these two paths, it follows from (17)–(18) that $j \leq u_j \leq u_i = i$, confirming that $i$ is indeed the peak on the route. Conversely, if $i$ is the peak customer on the route, it is implied that $p_i = 1$ as $p_i = 0$ is not possible which can be seen as follows. If $p_i = 0$, then there would be a path from $i$ (due to (10)) which necessarily would end at some customer $j$ with $p_j = 1$ implying $u_j = j$. Such a path should have nondecreasing $u$-values (due to (18)) starting with $u_i \geq i$, which would imply $i \leq u_i \cdots \leq u_j = j$, contradicting the assumption that $i$ is the highest index on the route. □

As a consequence of Proposition 1, our BPF contains a unique $p$ vector for any set of undirected routes to the CVRP.

**Proposition 2.** *There is a one-to-one correspondence between feasible solutions to the undirected CVRP and solutions to our BPF* (9)–(21).

*Proof.* Proof of Proposition 2. Suppose that we are given a feasible set of undirected routes to the CVRP. As shown in Proposition 1, the $p$-vector then follows directly. Moreover, for each undirected route, the $x$-values follow immediately given that the peak customer is known. Finally, having determined the vectors $p$ and $x$, the $f$ vector and the $t$ vector follow from constraints (12) and (14), given that $f_{0j} = 0 \; \forall j \in V_c$ as in the CFF. Conversely, for a given solution to our BPF, we can establish a unique feasible solution to the undirected CVRP simply by disregarding the direction of travel in our solution and letting $x_{ij} + x_{ji}$ represent the number of times a vehicle travels along edge $\{i, j\}$ in the undirected CVRP. □

## 3. Strengthening the formulation

In addition to the BPF itself, we also introduce two ways of strengthening the BPF as described in the following two subsections.

### *3.1. Peak inequalities.*

It follows from Definition 1 that any customer $j$ is visited on a route where the peak customer has index $j$ or greater. As such, for any customer $j \in V_c$, all customers in $S_j = \{j, \ldots, n\}$ must be serviced on routes where the peak customer has index $j$ or greater. This leads to the following inequalities which we refer to as *Rounded Peak Count Inequalities* (RPCIs):

$$\sum_{i \in S_j} p_i \geq \lceil \sum_{i \in S_j} q_i / Q \rceil \qquad \forall j = 1, \ldots, n \tag{22}$$

It follow as a special case that $p_n = 1$. In general, the left hand side of (22) is the number of routes which can possibly service one or more customers in $S_j$. This number must be large enough to allow the vehicles to service the entire demand of that customer set, as represented by the right hand side of (22). In general, several of the inequalities in (22) are dominated by others. Specifically, the only non-dominated inequalities in (22) are for sets $S_j$ requiring one more vehicle than $S_{j+1}$.

*3.2. Lifted inequalities.*

Another approach to strengthening the formulation is to lift already existing inequalities. We consider here the possibilities for lifting inequalities (18).

**Proposition 3.** *The constraints*

$$u_i - u_j + (n - j - 1)x_{ij} + (n - \max\{i, j\} - 1)x_{ji} \leq n - j - 1 \tag{23}$$

*are valid for all* $i, j \in V_c \setminus \{n\}$.

*Proof.* Proof of Proposition 3. We seek the largest coefficient $\alpha_{ji}$ such that the following inequality is valid:

$$u_i - u_j + (n - j - 1)x_{ij} + \alpha_{ji} x_{ji} \leq n - j - 1 \qquad \forall i, j \in V_c \setminus \{n\} \tag{24}$$

For $x_{ji} = 0$, (24) reduces to (18) and is therefore trivially valid for any $\alpha_{ji}$. For $x_{ji} = 1$, we have that $x_{ij} = 0$ is implied, so (24) reduces to $u_i - u_j + \alpha_{ji} \leq n - j - 1$. We consider two cases:

- $i < j$: In this case, it is valid to require that $u_i = u_j$, as traveling from a larger index $j$ to a smaller index $i$ means that the highest visited index so far remains unchanged when traveling from $j$ to $i$. In this case we can therefore set $\alpha_{ji} = n - j - 1$ as it results in $u_j \geq u_i$;

- $i > j$: In this case, a vehicle travels from a smaller index $j$ to a larger index $i$, meaning that the highest visited index so far possibly increases. Generally we have that $u_i = \max\{i, u_j\}$. If $u_i = i$, then it follows by observing $u_j \geq j$ that $u_i - u_j \leq i - j$, corresponding to $\alpha_{ji} = n - i - 1$. If, on the other hand, $u_i = u_j$, the largest possible value of $\alpha_{ji}$ is $n - j - 1$.

In order to cover all possibilities, we must ensure that $\alpha_{ji} \leq \min\{n - i - 1, n - j - 1\}$, resulting in $\alpha_{ji} = n - \max\{i, j\} - 1$ as the largest valid value of $\alpha_{ji}$. □

**Proposition 4.** *The constraints*

$$u_i - u_j + (n-j-1)x_{ij} + (n-i-1)p_i \leq n-j-1 \tag{25}$$

*are valid for all* $i, j \in V_c \setminus \{n\}$.

*Proof.* Proof of Proposition 4. We seek the largest coefficient $\beta_{ij}$ so that the following inequality is valid:

$$u_i - u_j + (n-j-1)x_{ij} + \beta_{ij}p_i \leq n-j-1 \qquad \forall i, j \in V_c \setminus \{n\} \tag{26}$$

For $p_i = 0$, (26) reduces to (18) and is therefore trivially valid for any $\beta_{ij}$. For $p_i = 1$, we have that $x_{ij} = 0$ as well as $u_i = i$ are implied, so (26) reduces to $i - u_j + \beta_{ij} \leq n - j - 1$. By rearranging we obtain $\beta_{ij} \leq n - i - 1 + u_j - j$, and observing that $u_j \geq j$ results in $\beta_{ij} = n - i - 1$ as the largest valid value of $\beta_{ij}$. □

For completeness we have also analyzed the possibility of lifting $p_j$ instead of $p_i$ in (26), and we have found that it is not possible to obtain a positive lifting coefficient for $p_j$.

## 4. Relations to other formulations

We find it appropriate to note that our BPF as well as the strengthenings bear some resemblance to certain previous efforts on modelling in vehicle routing. Indeed, our introduction of $u$-variables has certain similarities to elements of the MTZ-formulation proposed in [16] of the Asymmetric Traveling Salesman Problem, and our Proposition 3 is similar to the identification of a lifting coefficient for the MTZ-formulation in [17] and [18]. Moreover, our RPCIs may be viewed as a specialized version of Rounded Capacity Inequalities used in the two-index formulation of the CVRP (see, e.g., [3, 19]).

### *4.1. Formulation sizes.*

Although both the CFF and our BPF are polynomially sized formulations, there could still be potentially large differences in actual formulation sizes. For a more detailed comparison of formulation sizes we consider Table 1.

Given that $f_{0j} = 0$ for all $j \in V_c$, the CFF contains $n^2 + n$ 0-1 $x$-variables and $n^2$ continuous $f$-variables. The number of constraints in CFF is $2n^2 + 2n + 2$, given that the upper bounds on $f_{ij}$ are excluded for $i = 0$.

In the BPF we have $n^2 - 2$ integer variables $x$ and $p$, not counting $p_n$ which is fixed to 1, of which $n$ variables can take the values 0,1,2 and the rest are 0-1 variables. In addition, we have $n^2 - 2$ continuous variables $f$, $u$, and $t$. It is noted that there are no arcs leading back to the depot in the BPF, which therefore does not contain any $f_{j0}$-variables or $x_{j0}$-variables.

Thus, both models contain $\mathcal{O}(n^2)$ variables and $\mathcal{O}(n^2)$ constraints. Our BPF contains $\mathcal{O}(n^2)$ more constraints than the CFF, which on the other hand contains $\mathcal{O}(n)$ more integer variables than our BPF. Overall, we consider it fair to say that the two formulations are very similar in size.

Table 1: Number of variables and constraints in formulations.

| Elements | CFF | BPF |
|---|---|---|
| Variables | | |
| $x$ | $n^2 + n$ | $n^2 - n - 1$ |
| $p$ | | $n - 1$ |
| $f$ | $n^2$ | $n^2 - 2n - 1$ |
| $u$ | | $n - 1$ |
| $t$ | | $n$ |
| Constraints | $2n^2 + 2n + 2$ | $3n^2 + n + 3$ |

## 5. Computational experiments

In this section we report on the computational experiments conducted with the new formulation. The experiments serve the purpose of investigating the following points:

1. Whether the LP relaxation of our BPF is stronger than the LP relaxation of the CFF and to investigate the effect on the LP bound (i.e., the objective value of the LP relaxation) of sorting the customers in nondecreasing order of distance from the depot.
2. The effect on the performance of CPLEX obtained by using our BPF instead of the CFF.
3. The effect of strengthening the BPF as described in subsections 3.1 and 3.2.
4. The ability of solving more problems from the standard sets of test instances, and in smaller computation times, compared to the CFF formulation.
5. The increased ability of solving the model if the peak variables have been fixed in advance.

We have conducted experiments on certain classes of CVRP instances, specifically the `A`, `B`, `E`, and `P` instances which are available at CVRPLIB (`http://vrp.atd-lab.inf.puc-rio.br/index.php/en/`).

In the following, the strengthened version of BPF obtained by adding the RPCIs (22) and replacing (18) by (23) will be referred to as our *full peak formulation* (FPF).

### *5.1. The relative strengths of the LP relaxations*

In this section we report on the value of the LP relaxation obtained with our formulation compared to the LP relaxation of the CFF. For each problem instance, we have run the CFF as well as four versions of our formulation. Our four versions are run in two scenarios, one with the original numbering of customers and another where customers are sorted in nondecreasing order of distance from the depot. In the latter scenario, peak customers will tend to be located relatively far from the depot, which intuitively seems promising in

Table 2: Improvement of LP objective value compared to the CFF.

| | Customers unsorted | | | | Customers sorted | | | |
|---|---|---|---|---|---|---|---|---|
| Class | BPF | $p_i$ | $x_{ji}$ | FPF | BPF | $p_i$ | $x_{ji}$ | FPF |
| A | 0.20% | 0.20% | 1.14% | 1.14% | 0.72% | 0.72% | 1.96% | 3.65% |
| B | 0.35% | 0.35% | 1.03% | 1.05% | 1.12% | 1.12% | 2.41% | 6.97% |
| E | 1.27% | 1.27% | 3.25% | 3.62% | 1.73% | 1.73% | 3.17% | 4.32% |
| P | 0.56% | 0.56% | 1.50% | 1.54% | 1.26% | 1.26% | 2.25% | 2.54% |

terms of ensuring a certain amount of flow from the depot to distant customers and thereby also leading to a certain level of total cost. To accompany our intuition with a quantitative analysis, we have investigated both of the mentioned scenarios in our experiments. In addition, we compare the LP relaxation values of our peak-based formulations to that of the CFF. In recent years, significantly stronger polynomial formulations than the CFF have been proposed (see, e.g., [20] and [21]), but these come at the cost of increases in the problem formulation by additional constraints and/or variables. The development of these formulations has specifically focused on improving the LP bound, whereas the purpose of our formulation is to remove symmetry. Comparing the LP bound of our formulation with these stronger formulations will naturally lead to the conclusion that our formulation has a weaker lower bound, and thus no new information is gained. The CFF, on the other hand, is very similar in size, as has been argued in Section 4. In addition, it is well known that the LP bound of the CFF is equal to the LP bound obtained from the two-index vehicle flow formulation where Generalized Large Multistar (GLM) inequalities are used as subtour elimination constraints (see [11]). Although the GLM bound is not the strongest known for the CVRP, it is significantly better than for example the MTZ bound. Thus, we have chosen to compare with the CFF as it is of similar size to our formulation and is known to provide a fairly strong polynomial formulation of the CVRP.

In Table 2 we report the improvements obtained with our formulation relative to the CFF. The first column shows the instance class, the next four columns show the results obtained with the original numbering of customers, and the last four columns show the results where customers are sorted in nondecreasing order of distance from the depot. As a tie-breaking rule, if two or more customers have the same distance from the depot, then their relative ordering after sorting is the same as before sorting. Each number in the table is the average of all relative differences within the particular instance class.

Columns *BPF* are obtained using our BPF, columns $p_i$ are obtained with the formulation where (18) is replaced by (26), columns $x_{ji}$ are obtained with the formulation where (18) is replaced by (23), and columns *FPF* are obtained with our FPF. In the following we will refer to these four formulations as *peak based formulations*.

A general message from Table 2 is that all values are positive, demonstrating that our peak based formulations produce better lower bounds than the CFF.

To this we can add that the peak based formulations never produced a smaller LP objective value than the CFF for any of the 85 instances tested here.

Table 2 also demonstrates that the sorting of customers has a significant effect on the LP objective value. In fact, for the `B` instances, the LP objective value is on average improved more by sorting the customers than by lifting the variable $x_{ji}$ in constraints (18).

The RPCIs have a much larger effect when the customers are sorted. The intuitive explanation is that the sorting moves the peak customers relatively far away from the depot, where it has a larger effect to require a certain amount of traffic from the depot to the peaks. We would like to note that in 13 out of the 85 instances, sorting the customers results in worse LP bounds than keeping the original ordering when solving BPF, and in 5 instances the bound is also worse with sorting when solving the model where $x_{ji}$ is lifted. However, when adding the RPCIs, it is in all instances better to sort the customers, confirming our intuitive explanation of the effect of the RPCIs.

The last insight emphasized here is that lifting the variable $p_i$ as described in Proposition 4 has no effect on the LP bound. This is in spite of the fact that initial computational experiments confirmed that the coefficient identified in Proposition 4 is indeed the largest valid lifting coefficient for $p_i$ in equalities (18), and that, to the best of our knowledge, constraints (25) are not implied by the other constraints in the model.

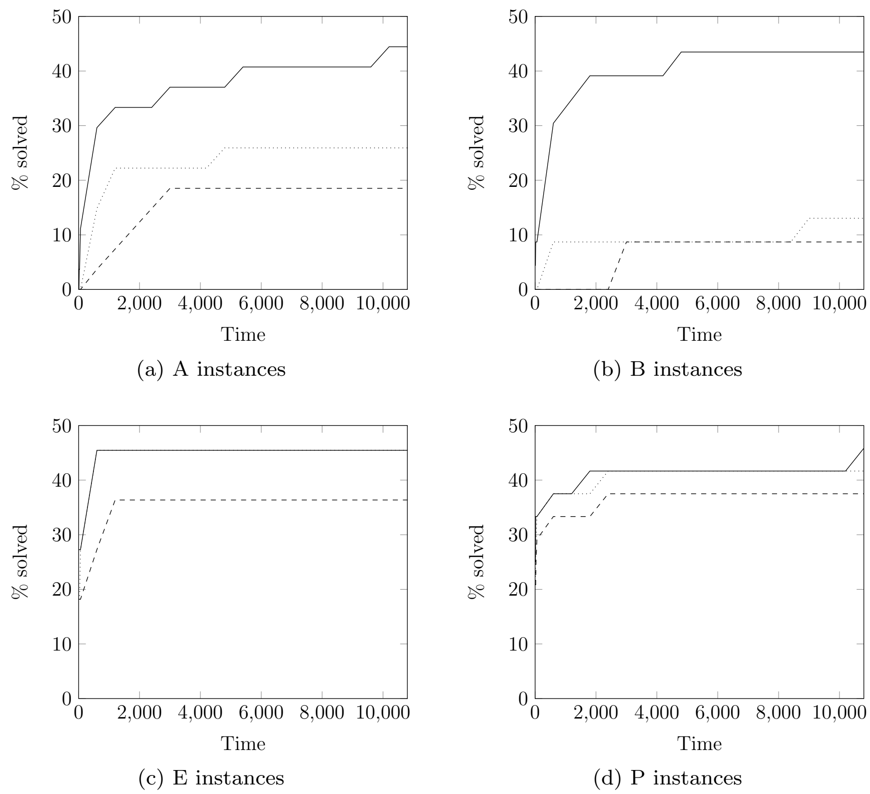
### *5.2. Solving the formulations using CPLEX*

In this section, we investigate the performance of the peak based formulations for solving the CVRP to optimality. All formulations are coded using the C++ API for CPLEX 12.7 concert technology and all experiments are carried out on a PC with a 2.6 GHz Intel Core i7 processor and 12 GB RAM running Windows 10. The code is compiled using Microsoft Visual Studio 2015 with optimization option O2. All settings in CPLEX are at their default except that we set a time limit of 3 hours for each instance.

We have strengthened the peak based formulations by reducing the upper bound on $x_{0j}$ to one if it can be observed that it is not feasible to serve customer $j$ as the only customer on a route. With $K$ vehicles in total, this will be the case if the total capacity of $K-1$ vehicles is not sufficient to service all other customers. That is, we set the upper bound on $x_{0j}$ to one if $\sum_{i \in V_c \setminus \{j\}} q_i > (K-1)Q$.

In preliminary experiments, we saw that CPLEX quickly found good solutions for the peak based formulations but not for the CFF. In order to dampen the effect of CPLEX' internal heuristics, we implemented a simple deterministic heuristic for generating a starting solution of reasonable quality which we used across all models, including both the peak based formulations and the CFF.

We have applied the three formulations CFF, BPF, and FPF, to each of the test instances. For BPF and FPF, given the effect of sorting as shown in Table 2, we have sorted the customers in nondecreasing order of distance from the depot. For each of the four classes of instances, we calculated the fraction of all instances in the class which have been solved within a given time limit of 3 hours.




(a) A instances (b) B instances

(c) E instances (d) P instances

Figure 2: Performance profiles for the three models FPF (——), BPF (· · · · · ·), and CFF (– – –). The diagrams show the percentage of solved instances as a function of time in CPU seconds.

Table 3: Average speeedup factors.

| Instances | A | B | E | P |
|---|---|---|---|---|
| BPF | 3.5 | 10.6 | 4.6 | 3.0 |
| FPF | 26.7 | 103.1 | 25.1 | 6.5 |

The results are summarized in Figure 2. As expected, we notice that the peak based formulations perform better than the CFF, and it is encouraging to see that even the basic formulation BPF consistently solves more problems within the same time limit compared to the CFF.

However, on a number of instances, a comparison between different peak based formulations shows only little difference in the performance between BPF and FPF. This happens in particular on the `E` instances, although the LP bound is improved quite significantly from BPF to FPF for these instances. The explanation is that the cuts added by CPLEX compensate to a large extent for a weaker BPF formulation, which leads the two formulations to exhibit approximately the same bound after CPLEX' cuts have been added.

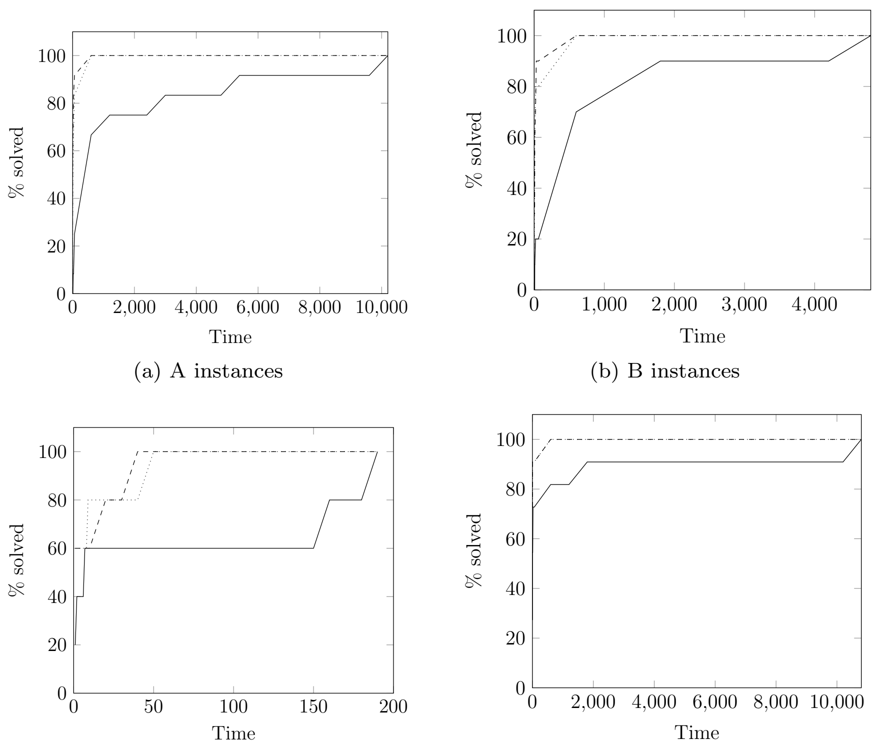
A very important insight is, that for some of the instances, the CFF formulation exhibits an exorbitant memory consumption. In the case of `E-n76-k14`, the full consequence of the exponential number of solutions representing the same set of undirected routes is seen; before the three hours time limit is reached, CPLEX has built a 40 GB branching tree where the global lower bound does not come close to that of the Full formulation. Eventually, the operating system shut down the process due to excessive memory consumption. In several other cases, the branching trees for CFF reached more than one million active nodes. This never happened with the peak based formulations, indicating that avoiding symmetries in the formulation has a very pronounced effect.

Finally, in Table 3 we show the average speedup factors obtained over all instances which were solved by the CFF. All of these instances were also solved by both the BPF and FPF formulations. For each instance, we calculated the ratio between the time spent by the CFF and the BPF. The *BPF* row in Table 3 shows the average of these ratios within each instance class. Similar calculations were done for producing the *FPF* row. The numbers show an average speedup factor of 3.0–10.6 obtained by using the BPF instead of the CFF, and an average speedup factor of 6.5–103.1 obtained by using the FPF formulation instead of the CFF. We consider these results to be strong indications of the performance gains obtained by using our peak based formulations compared to using the CFF.

In the Appendix we provide more detailed results on each individual instance obtained with the three formulations.

### *5.3. The effect of knowing the peaks a priori*

In Subsection 2.2 we mentioned the heuristic from [14], in which the initial heuristic choice of special cities reduces the remaining search for a solution.




(a) A instances (b) B instances

(c) E instances (d) P instances

Figure 3: Performance profiles of FPF (——), BPFFP (······), and FPFFP (– – –). The diagrams show the percentage of solved instances as a function of time in CPU seconds.

Along the same lines, an initial identification of peak customers in our formulation might significantly reduce the remaining computational effort. In this Subsection we investigate the effect of having identified the peak customers in advance.

Specifically, we have resolved the instances which were solved to proven optimality by FPF and observed the optimal values of peak variables. Then the models are run again, this time with all peak variables fixed in advance to their optimal values. For ease of presentation, we let BPFFP and FPFFP denote the BPF and FPF, respectively, with peak variables fixed to their optimal values.

A total of 39 instances were analyzed in this way. Out of these, BPFFP and FPFFP solve as many as 13 and 17 instances, respectively, without branching, whereas BPF and FPF solve only 3 and 4 instances, respectively, without branching.

Moreover, in Figure 3 we show the percentage of the problems solved as a function of the computation time in seconds.

Both BPFFP and FPFFP are extremely efficient in solving the remaining problem. As can be seen in Figure 3, all instances were solved by both BPFFP and FPFFP within 10 minutes, and for the E instances only one minute was required. Another observation is that BPFFP and FPFFP perform almost equally well, which can be explained as follows.

Fixing of peak customers has important implications for strengthening the BPF as well as the FPF. In general, for any customer set $S$, let $\pi(S)$ denote the number of customers in $S$ which have been fixed as peaks. Given that the BPF contains a flow of two units from the depot to each peak, the flow which enters $S$ will be at least $2\pi(S)$. Hence, the following inequalities are satisfied after fixing of peak customers:

$$x(\delta^-(S)) \geq 2\pi(S) \qquad \forall S \subseteq V_c \tag{27}$$

If we start from an LP solution to BPF which satisfies (27) and transforms it to an undirected solution to the CVRP as described in Proposition 2, we will clearly obtain an undirected solution which satisfies the following weakened version of (27):

$$x(\delta^-(S)) + x(\delta^+(S)) \geq 2\pi(S) \qquad \forall S \subseteq V_c \tag{28}$$

The left hand side of (28) is the sum of edge variables connecting $S$ and $V \setminus S$ in the resulting undirected 2-index formulation of the CVRP. In comparison, if we define $x(\delta(S)) = x(\delta^-(S)) + x(\delta^+(S))$, the Rounded Capacity Inequalities (RCIs) for the undirected CVRP are the following:

$$x(\delta(S)) \geq 2k(S) \qquad \forall S \subseteq V_c \tag{29}$$

where

$$k(S) = \lceil \sum_{i \in S} q_i / Q \rceil \tag{30}$$

From these observations it follows that for any customer set $S$ for which $\pi(S) \geq k(S)$, the BPFFP satisfies a constraint which dominates the corresponding RCI. Given the importance of the RCIs in the 2-index formulation, this is a strong characteristic of the BPFFP despite it being only polynomial in size. At the same time, it also means that there is less room for further strengthening obtained by the liftings involved in FPF. As such, we consider the implied RCIs in the BPFFP to be the main explanation for the small difference in performance between the BPFFP and the FPFFP.

In order to gain further insight into the effect of peak fixing, we have compared it with exact separation of RCIs in the 2-index formulation of the CVRP. The results are summarized in Table 4. In relation to the results reported under 'Basic model', LP(BPF), LP(BPFFP), and LP(RCI) denote the LP bound obtained with the BPF, the BPFFP, and the 2-index formulation using exact separation of RCIs, respectively. The results reported under 'Full model' are calculated equivalently. The values of LP(RCI) are from [22].

Table 4 shows that, on average, peak fixing leads to a bound improvement which is approximately 57%-84% of the bound improvement obtained by RCIs.

Table 4: Effect of peak fixing relative to exact separation of RCIs. The reported values are minimum, average, and maximum values of (LP(BPFFP)−LP(BPF)) / (LP(RCI)− LP(BPF)) ×100 for 'Basic model' and of (LP(FPFFP)−LP(FPF)) / (LP(RCI)− LP(FPF)) ×100 for 'Full model'.

| | Basic model | | | Full model | | |
|---|---|---|---|---|---|---|
| Instances | min | avg. | max | min | avg. | max |
| A | 43.18 | 84.96 | 109.16 | 34.07 | 83.49 | 111.06 |
| B | 43.92 | 73.37 | 101.02 | 12.22 | 57.87 | 102.34 |
| E | 54.20 | 73.54 | 95.21 | 37.20 | 68.28 | 88.65 |
| P | 11.33 | 71.25 | 131.10 | 0.86 | 74.26 | 147.26 |

This is, in our view, in itself a quite good performance obtained by peak fixing. Moreover, the fact that there are values above 100% confirms the strength of peak fixing and indeed demonstrates the dominance relation as described in relation to (27)–(30).

It is also worth noting that there is more variability in the numbers for the `P` instances than for the other three classes. This is related to the fact that the `P` instances are very varied with respect to the number of vehicles. Indeed, the smallest values among the `P` instances occur on instances with only two vehicles. As $p_n$ is already fixed to one in the BPF, there is only one peak customer which remains to be fixed in the BPFFP and FPFFP in these instances, which significantly limits the effect of peak fixing. Moreover, a small number of peaks leaves many customer sets without peaks for which it is an advantage to have the RCIs.

## 6. Conclusions and perspectives

In this paper we have proposed a polynomial directed formulation of the symmetric CVRP in which difficulties of symmetry are avoided. We have shown that the new formulation produces better lower bounds and leads to significantly smaller computing times than a classical formulation proposed in the literature. Furthermore, given its polynomial size, our model can be formulated with relatively little effort, in particular if using an algebraic modelling language, which in general has many advantages. Altogether, our formulation can be recommended for applications where it is desired to enter a formulation in its entirety into a mixed-integer linear programming solver, avoiding the need for dynamically generating rows or columns during the solution process.

Moreover, we have found that an a priori identification of peak customers leads to a considerable speedup of the solution process. This suggests some interesting perspectives for further research. In particular, it would be possible to design an iterative process where a separate model is used for identifying a new set of peak customers in each iteration, and where each iteration also involves solving the formulation to optimality for the given set of peak customers. Along these lines, we find it natural to consider the use of *local branching* ([23]) and *cut-and-solve* ([24]) for future research.

## References


[1] P. Toth, D. Vigo (Eds.), The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, USA, 2002.

[2] P. Toth, D. Vigo (Eds.), Vehicle Routing: Problems, Methods, and Applications, MOS-SIAM series on optimization, Philadelphia, PA, USA, 2014.

[3] G. Laporte, Y. Nobert, M. Desrochers, Optimal routing under capacity and distance restrictions, Operations Research 33 (5) (1985) 1050–1073.

[4] M. L. Balinski, R. E. Quandt, On an integer program for a delivery problem, Operations Research 12 (2) (1964) 300–304.

[5] R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, R. F. Werneck, Robust branch-and-cut-and-price for the capacitated vehicle routing problem, Mathematical Programming 106 (3) (2006) 491–511.

[6] F. Semet, P. Toth, D. Vigo, Classical exact algorithms for the capacitated vehicle routing problems, in: Toth and Vigo [2], Ch. 2.

[7] M. Poggi, E. Uchoa, New exact algorithms for the capacitated vehicle routing problems, in: Toth and Vigo [2], Ch. 3.

[8] D. Pecin, A. Pessoa, M. Poggi, E. Uchoa, Improved branch-cut-and-price for capacitated vehicle routing, Mathematical Programming Computation 9 (1) (2017) 61–100.

[9] B. Gavish, S. C. Graves, The travelling salesman problem and related problems, Tech. rep., Working paper OR 078-78, Operations Research Center, Massachusetts Institute of Technology (1978).

[10] L. Gouveia, A result on projection for the vehicle routing problem, European Journal of Operational Research 85 (3) (1995) 610–624.

[11] A. N. Letchford, J.-J. Salazar-González, Projection results for vehicle routing, Mathematical Programming 105 (2–3) (2006) 251–274.

[12] R. Fourer, Algebraic modeling languages for optimization, in: S. I. Gass, M. C. Fu (Eds.), Encyclopedia of Operations Research and Management Science, Springer, Boston, MA, USA, 2013, pp. 43–51.

[13] R. Jonker, G. de Leve, J. A. van der Velde, A. Volgenant, Bounding symmetric traveling salesman problems with an asymmetric assignment problem, Operations Research 28 (3) (1980) 623–627.

[14] P. H. van Leeuwen, A. Volgenant, Solving symmetric vehicle routing problems asymmetrically, European Journal of Operational Research 12 (4) (1983) 388–393.

[15] G. Righini, M. Salani, Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints, Discrete Optimization 3 (3) (2006) 255–273.

[16] C. E. Miller, A. W. Tucker, R. A. Zemlin, Integer programming formulation of traveling salesman problems, Journal of the Association for Computing Machinery 7 (4) (1960) 326–329.

[17] M. Desrochers, G. Laporte, Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints, Operations Research Letters 10 (1) (1991) 27–36.

[18] I. Kara, G. Laporte, T. Bektas, A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem, European Journal of Operational Research 158 (3) (2004) 793–795.

[19] J. Lysgaard, A. N. Letchford, R. W. Eglese, A new branch-and-cut algorithm for the capacitated vehicle routing problem, Mathematical Programming 100 (2) (2004) 423–445.

[20] A. N. Letchford, J.-J. Salazar-González, Stronger multi-commodity flow formulations of the Capacitated Vehicle Routing Problem, European Journal of Operational Research 244 (3) (2015) 730–738.

[21] A. N. Letchford, G. Souli, On lifted cover inequalities: A new lifting procedure with unusual properties, Operations Research Letters 47 (2) (2019) 83–87.

[22] K. Wenger, Generic cut generation methods for routing problems, Ph.D. thesis, Institute of Computer Science, University of Heidelberg, Heidelberg, Germany (2003).

[23] M. Fischetti, A. Lodi, Local branching, Mathematical Programming 98 (1–3) (2003) 23–47.

[24] S. Climer, W. Zhang, Cut-and-solve: An iterative search strategy for combinatorial optimization problems, Artificial Intelligence 170 (8–9) (2006) 714–738.

## Appendix

In this appendix we provide detailed information on the computational experiments. For each of the four classes `A`, `B`, `E`, and `P` of CVRP instances, a separate table shows the information obtained for the instances in that class.

To explain the contents of the tables, we consider Table 5 which shows the results for the A instances. The first column contains the instance name, where the first letter refers to the instance class, and the two numbers show the number of nodes including the depot, and the number of vehicles, respectively. For example, A-n32-k5 is an instance from class A with 32 nodes (31 customers plus a depot) and 5 vehicles. The second column shows the objective value of an optimal solution.

The remaining columns are divided into three groups representing our three formulations CFF, BPF, and FPF.

Each group contains five columns. Column *LP* contains the objective value of the linear programming relaxation. Column $T^{LP}$ contains the time in seconds required for solving the linear programming relaxation. Column *UB* contains the value of the best upper bound found within the time limit of three hours by that formulation, marked with an asterisk if it is equal to the optimal objective value. Column $T$ reports the time required for solving the problem to optimality, where a '—' represents that the model did not solve the instance to optimality within the three hours time limit. Finally, column *Gap* reports the gap between the lower bound and the upper bound found by that formulation within the three hours time limit, where a '—' represents that the model was solved to optimality. Among all three formulations, the one which solved the problem to optimality in the least amount of time has its $T$ value shown in bold.

In a similar way, Tables 6, 7 and 8 show the information for instance classes B, E, and P, respectively.

Table 5: Detailed results for A instances.

| | | CFF | | | | | BPF | | | | | FPF | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | Opt | LP | $T^{LP}$ | UB | T | Gap | LP | $T^{LP}$ | UB | T | Gap | LP | $T^{LP}$ | UB | T | Gap |
| A-n32-k5 | 784 | 687.24 | 0.04 | 784* | 495.2 | — | 697.80 | 0.05 | 784* | 386.2 | — | 717.91 | 0.04 | 784* | **10.7** | — |
| A-n33-k5 | 661 | 584.23 | 0.03 | 661* | 696.9 | — | 588.06 | 0.06 | 661* | 171.9 | — | 597.02 | 0.07 | 661* | **39.3** | — |
| A-n33-k6 | 742 | 649.39 | 0.05 | 742* | 2796.9 | — | 654.18 | 0.07 | 742* | 434.1 | — | 669.89 | 0.08 | 742* | **106.9** | — |
| A-n34-k5 | 778 | 650.26 | 0.03 | 778* | 1472.2 | — | 664.56 | 0.05 | 778* | 649.8 | — | 691.45 | 0.06 | 778* | **149.2** | — |
| A-n36-k5 | 799 | 716.11 | 0.05 | 799* | — | 1.7% | 725.12 | 0.08 | 799* | 683.2 | — | 748.65 | 0.08 | 799* | **83.3** | — |
| A-n37-k5 | 669 | 597.49 | 0.05 | 669* | 1956.1 | — | 599.61 | 0.08 | 669* | 539.2 | — | 622.75 | 0.09 | 669* | **40.5** | — |
| A-n37-k6 | 949 | 837.51 | 0.05 | 950 | — | 4.2% | 843.09 | 0.06 | 950 | — | 2.8% | 869.61 | 0.11 | 950 | — | 0.9% |
| A-n38-k5 | 730 | 607.40 | 0.06 | 731 | — | 3.4% | 616.15 | 0.07 | 730* | 4599.3 | — | 645.58 | 0.10 | 730* | **2870.9** | — |
| A-n39-k5 | 822 | 737.62 | 0.08 | 825 | — | 3.7% | 737.62 | 0.12 | 822* | — | 1.8% | 758.39 | 0.11 | 822* | **554.2** | — |
| A-n39-k6 | 831 | 715.14 | 0.07 | 831* | — | 3.4% | 715.67 | 0.08 | 831* | — | 1.0% | 742.46 | 0.17 | 831* | **495.8** | — |
| A-n44-k6 | 937 | 837.25 | 0.12 | 942 | — | 3.6% | 849.53 | 0.14 | 942 | — | 2.0% | 869.54 | 0.17 | 937* | **5097.4** | — |
| A-n45-k6 | 944 | 799.86 | 0.10 | 973 | — | 7.9% | 808.73 | 0.14 | 966 | — | 6.3% | 825.28 | 0.17 | 944* | — | 2.3% |
| A-n45-k7 | 1146 | 1027.43 | 0.14 | 1147 | — | 3.8% | 1035.88 | 0.17 | 1146* | — | 4.1% | 1059.74 | 0.20 | 1146* | — | 0.6% |
| A-n46-k7 | 914 | 810.85 | 0.15 | 914* | — | 1.5% | 813.37 | 0.18 | 917 | — | 2.3% | 844.71 | 0.15 | 914* | **699.1** | — |
| A-n48-k7 | 1073 | 965.57 | 0.15 | 1102 | — | 7.4% | 973.65 | 0.17 | 1086 | — | 5.3% | 986.75 | 0.23 | 1073* | **10083.0** | — |
| A-n53-k7 | 1010 | 880.78 | 0.17 | 1017 | — | 3.4% | 888.99 | 0.26 | 1017 | — | 4.0% | 930.99 | 0.33 | 1010* | — | 1.0% |
| A-n54-k7 | 1167 | 1032.14 | 0.19 | 1178 | — | 7.5% | 1035.27 | 0.30 | 1186 | — | 6.9% | 1064.35 | 0.36 | 1167* | — | 3.3% |
| A-n55-k9 | 1073 | 939.95 | 0.21 | 1074 | — | 5.0% | 940.53 | 0.29 | 1074 | — | 3.8% | 979.71 | 0.32 | 1074 | — | 2.8% |
| A-n60-k9 | 1354 | 1202.71 | 0.20 | 1373 | — | 6.9% | 1208.18 | 0.54 | 1373 | — | 6.4% | 1240.17 | 0.46 | 1373 | — | 5.4% |
| A-n61-k9 | 1034 | 919.06 | 0.23 | 1074 | — | 8.8% | 931.36 | 0.59 | 1056 | — | 6.7% | 943.61 | 0.39 | 1059 | — | 6.1% |
| A-n62-k8 | 1288 | 1141.41 | 0.22 | 1312 | — | 8.2% | 1141.85 | 0.77 | 1323 | — | 8.7% | 1173.53 | 0.48 | 1288* | — | 4.1% |
| A-n63-k10 | 1314 | 1139.01 | 0.23 | 1319 | — | 5.8% | 1142.40 | 0.64 | 1328 | — | 6.3% | 1180.63 | 0.50 | 1319 | — | 5.3% |
| A-n63-k9 | 1616 | 1470.88 | 0.21 | 1638 | — | 6.0% | 1475.84 | 0.69 | 1627 | — | 4.9% | 1519.07 | 0.49 | 1618 | — | 2.3% |
| A-n64-k9 | 1401 | 1234.67 | 0.25 | 1414 | — | 5.5% | 1247.83 | 0.77 | 1422 | — | 5.3% | 1287.88 | 0.60 | 1422 | — | 4.8% |
| A-n65-k9 | 1174 | 1060.75 | 0.23 | 1208 | — | 7.7% | 1060.75 | 0.69 | 1204 | — | 6.4% | 1077.40 | 0.52 | 1211 | — | 6.7% |
| A-n69-k9 | 1159 | 1012.36 | 0.24 | 1167 | — | 5.4% | 1024.03 | 0.79 | 1177 | — | 5.7% | 1043.76 | 0.59 | 1169 | — | 4.2% |
| A-n80-k10 | 1763 | 1591.23 | 0.80 | 1804 | — | 6.9% | 1592.30 | 1.26 | 1843 | — | 8.3% | 1635.14 | 0.96 | 1786 | — | 4.3% |

Table 6: Detailed results for B instances.

| Instance | Opt | CFF LP | CFF $T^{LP}$ | CFF UB | CFF T | CFF Gap | BPF LP | BPF $T^{LP}$ | BPF UB | BPF T | BPF Gap | FPF LP | FPF $T^{LP}$ | FPF UB | FPF T | FPF Gap |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B-n31-k5 | 672 | 573.21 | 0.02 | 672* | 2517.8 | — | 626.04 | 0.04 | 672* | 258.8 | — | 646.78 | 0.04 | 672* | **151.8** | — |
| B-n34-k5 | 788 | 690.61 | 0.04 | 788* | — | 2.0% | 695.40 | 0.08 | 788* | 8739.6 | — | 746.83 | 0.07 | 788* | **81.6** | — |
| B-n35-k5 | 955 | 817.43 | 0.02 | 955* | — | 4.4% | 824.10 | 0.09 | 955* | — | 2.9% | 879.95 | 0.06 | 955* | **8.9** | — |
| B-n38-k6 | 805 | 679.31 | 0.05 | 806 | — | 8.9% | 685.01 | 0.11 | 806 | — | 8.5% | 730.51 | 0.08 | 805* | **168.9** | — |
| B-n39-k5 | 549 | 452.97 | 0.05 | 549* | 2885.6 | — | 464.68 | 0.09 | 549* | 250.6 | — | 518.98 | 0.08 | 549* | **15.2** | — |
| B-n41-k6 | 829 | 702.23 | 0.06 | 830 | — | 4.3% | 704.41 | 0.09 | 829* | — | 3.4% | 777.25 | 0.11 | 829* | **339.1** | — |
| B-n43-k6 | 742 | 630.05 | 0.08 | 748 | — | 7.4% | 633.43 | 0.15 | 742* | — | 5.9% | 688.64 | 0.15 | 742* | **833.2** | — |
| B-n44-k7 | 909 | 795.08 | 0.08 | 909* | — | 2.1% | 795.08 | 0.19 | 924 | — | 7.7% | 841.49 | 0.18 | 909* | **96.6** | — |
| B-n45-k5 | 751 | 602.76 | 0.09 | 751* | — | 8.0% | 605.65 | 0.17 | 751* | — | 7.6% | 660.30 | 0.23 | 751* | **1379.8** | — |
| B-n45-k6 | 678 | 565.28 | 0.09 | 684 | — | 5.8% | 565.31 | 0.16 | 678* | — | 4.3% | 597.09 | 0.17 | 680 | — | 1.0% |
| B-n50-k7 | 741 | 633.73 | 0.13 | 741* | — | 2.8% | 646.99 | 0.35 | 741* | — | 1.6% | 661.29 | 0.21 | 741* | **4300.3** | — |
| B-n50-k8 | 1312 | 1160.71 | 0.17 | 1318 | — | 4.3% | 1162.61 | 0.41 | 1329 | — | 5.1% | 1225.67 | 0.33 | 1316 | — | 2.0% |
| B-n51-k7 | 1032 | 889.47 | 0.11 | 1051 | — | 8.7% | 891.84 | 0.35 | 1034 | — | 6.7% | 913.08 | 0.30 | 1032* | — | 4.2% |
| B-n52-k7 | 747 | 602.49 | 0.16 | 747* | — | 10.0% | 609.20 | 0.39 | 747* | — | 9.4% | 655.88 | 0.31 | 747* | — | 7.7% |
| B-n56-k7 | 707 | 578.85 | 0.19 | 710 | — | 1.7% | 580.39 | 0.36 | 719 | — | 12.8% | 619.48 | 0.38 | 717 | — | 11.0% |
| B-n57-k7 | 1153 | 1021.06 | 0.19 | 1159 | — | 4.6% | 1027.20 | 0.57 | 1186 | — | 6.1% | 1113.31 | 0.42 | 1166 | — | 2.6% |
| B-n57-k9 | 1598 | 1467.47 | 0.19 | 1612 | — | 3.7% | 1471.28 | 0.49 | 1611 | — | 3.8% | 1507.78 | 0.36 | 1598* | — | 1.7% |
| B-n63-k10 | 1496 | 1360.94 | 0.24 | 1538 | — | 7.0% | 1369.67 | 0.71 | 1547 | — | 7.3% | 1409.72 | 0.51 | 1558 | — | 5.7% |
| B-n64-k9 | 861 | 733.08 | 0.24 | 892 | — | 7.9% | 755.65 | 0.77 | 878 | — | 6.1% | 778.85 | 0.56 | 861* | — | 2.7% |
| B-n66-k9 | 1316 | 1162.89 | 0.25 | 1349 | — | 8.3% | 1164.57 | 0.87 | 1348 | — | 7.9% | 1230.16 | 0.58 | 1327 | — | 2.5% |
| B-n67-k10 | 1032 | 940.50 | 0.25 | 1041 | — | 5.2% | 941.57 | 0.71 | 1076 | — | 8.5% | 972.09 | 0.59 | 1041 | — | 2.8% |
| B-n68-k9 | 1272 | 1120.73 | 0.25 | 1289 | — | 8.1% | 1136.72 | 0.67 | 1289 | — | 7.7% | 1177.73 | 0.65 | 1290 | — | 4.5% |
| B-n78-k10 | 1221 | 1072.46 | 0.34 | 1260 | — | 10.2% | 1073.81 | 2.71 | 1252 | — | 9.6% | 1117.64 | 0.89 | 1247 | — | 6.9% |

Table 7: Detailed results for E instances.

| | | CFF | | | | | BPF | | | | | FPF | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | Opt | LP | $T^{LP}$ | UB | T | Gap | LP | $T^{LP}$ | UB | T | Gap | LP | $T^{LP}$ | UB | T | Gap |
| E-n22-k4 | 375 | 330.563 | 0.02 | 375* | 2.5 | — | 338.16 | 0.01 | 375* | 1.9 | — | 359.45 | 0.01 | 375* | **1.2** | — |
| E-n23-k3 | 569 | 489.8 | 0.01 | 569* | **0.3** | — | 531.17 | 0.01 | 569* | 0.6 | — | 541.68 | 0.01 | 569* | 0.4 | — |
| E-n30-k3 | 534 | 433.6 | 0.02 | 534* | — | 2.4% | 435.85 | 0.03 | 534* | 440.4 | — | 463.46 | 0.03 | 534* | **150.1** | — |
| E-n33-k4 | 835 | 772.2 | 0.04 | 835* | 582.0 | — | 792.64 | 0.05 | 835* | 37.4 | — | 811.40 | 0.06 | 835* | **6.2** | — |
| E-n51-k5 | 521 | 483.6 | 0.17 | 521* | 668.2 | — | 491.08 | 0.23 | 521* | 587.1 | — | 499.25 | 0.27 | 521* | **182.7** | — |
| E-n76-k7 | 682 | 612.3 | 0.29 | 689 | — | 4.0% | 620.18 | 0.91 | 691 | — | 4.4% | 636.43 | 0.96 | 689 | — | 3.6% |
| E-n76-k8 | 735 | 662.2 | 0.26 | 738 | — | 4.2% | 666.62 | 0.90 | 748 | — | 5.1% | 679.53 | 0.85 | 743 | — | 3.4% |
| E-n76-k10 | 835 | 750.6 | 0.29 | 846 | — | 6.7% | 752.84 | 0.89 | 869 | — | 8.4% | 761.28 | 0.85 | 856 | — | 6.8% |
| E-n76-k14 | 1021 | 924.1 | 0.33 | 1062 | — | 9.6% | 925.99 | 0.96 | 1056 | — | 7.9% | 929.19 | 0.82 | 1059 | — | 7.9% |
| E-n101-k8 | 815 | 744.1 | 1.12 | 826 | — | 4.9% | 752.30 | 1.77 | 828 | — | 4.8% | 770.04 | 1.53 | 829 | — | 4.8% |
| E-n101-k14 | 1067 | 982.5 | 1.3 | 1103 | — | 8.2% | 982.71 | 1.93 | 1081 | — | 6.0% | 991.95 | 1.59 | 1117 | — | 8.3% |

Table 8: Detailed results for P instances.

| | | CFF | | | | | BF | | | | | FF | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | Opt | LP | $T^{LP}$ | UB | T | Gap | LP | $T^{LP}$ | UB | T | Gap | LP | $T^{LP}$ | UB | T | Gap |
| P-n16-k8 | 450 | 423.71 | 0.01 | 450* | 1.0 | — | 423.71 | 0.01 | 450* | **0.5** | — | 423.71 | 0.005 | 450* | 0.6 | — |
| P-n19-k2 | 212 | 184.28 | 0.01 | 212* | 2.1 | — | 193.46 | 0.01 | 212* | **0.6** | — | 194.92 | 0.005 | 212* | 1.4 | — |
| P-n20-k2 | 216 | 194.51 | 0.01 | 216* | 1.7 | — | 200.75 | 0.01 | 216* | 0.8 | — | 205.98 | 0.005 | 216* | **0.7** | — |
| P-n21-k2 | 211 | 193.93 | 0.01 | 211* | 0.8 | — | 200.39 | 0.01 | 211* | 0.6 | — | 206.90 | 0.009 | 211* | **0.4** | — |
| P-n22-k2 | 216 | 198.17 | 0.01 | 216* | 1.4 | — | 204.14 | 0.01 | 216* | **0.8** | — | 209.32 | 0.006 | 216* | 1.1 | — |
| P-n22-k8 | 603 | 535.56 | 0.01 | 603* | 28.6 | — | 541.20 | 0.01 | 603* | 12.0 | — | 549.90 | 0.015 | 603* | **1.4** | — |
| P-n23-k8 | 529 | 491.89 | 0.01 | 529* | 46.1 | — | 494.94 | 0.02 | 529* | **10.2** | — | 495.79 | 0.031 | 529* | 12.0 | — |
| P-n40-k5 | 458 | 422.19 | 0.09 | 458* | 80.5 | — | 433.79 | 0.10 | 458* | 43.0 | — | 440.85 | 0.104 | 458* | **11.7** | — |
| P-n45-k5 | 510 | 471.18 | 0.14 | 510* | 2296.9 | — | 480.65 | 0.14 | 510* | 313.5 | — | 485.82 | 0.210 | 510* | **124.1** | — |
| P-n50-k7 | 554 | 509.93 | 0.15 | 554* | — | 1.2% | 517.52 | 0.22 | 554* | 1816.2 | — | 518.77 | 0.203 | 554* | **1372.4** | — |
| P-n50-k8 | 631 | 567.72 | 0.16 | 634 | — | 6.2% | 569.10 | 0.23 | 662 | — | 9.5% | 570.06 | 0.246 | 656 | — | 8.4% |
| P-n50-k10 | 696 | 636.38 | 0.15 | 697 | — | 3.9% | 641.52 | 0.23 | 697 | — | 2.5% | 641.94 | 0.249 | 697 | — | 2.6% |
| P-n51-k10 | 741 | 680.93 | 0.16 | 755 | — | 5.4% | 684.62 | 0.25 | 741* | — | 2.5% | 687.15 | 0.342 | 749 | — | 3.9% |
| P-n55-k7 | 568 | 519.00 | 0.18 | 574 | — | 4.6% | 522.14 | 0.28 | 568* | — | 3.1% | 523.98 | 0.344 | 575 | — | 3.0% |
| P-n55-k10 | 694 | 637.21 | 0.17 | 698 | — | 5.0% | 637.43 | 0.28 | 698 | — | 4.5% | 639.53 | 0.335 | 696 | — | 2.6% |
| P-n55-k15 | 989 | 872.73 | 0.17 | 1007 | — | 10.0% | 874.00 | 0.27 | 1007 | — | 9.2% | 874.52 | 0.352 | 1007 | — | 9.4% |
| P-n60-k10 | 744 | 674.91 | 0.20 | 750 | — | 3.8% | 682.87 | 0.53 | 747 | — | 2.8% | 686.72 | 0.436 | 744* | — | 2.4% |
| P-n60-k15 | 968 | 895.11 | 0.21 | 976 | — | 4.8% | 897.63 | 0.52 | 982 | — | 4.9% | 899.05 | 0.423 | 973 | — | 2.8% |
| P-n65-k10 | 792 | 724.68 | 0.22 | 799 | — | 4.0% | 730.76 | 0.57 | 792* | — | 2.7% | 735.67 | 0.512 | 808 | — | 4.5% |
| P-n70-k10 | 827 | 748.07 | 0.26 | 869 | — | 9.5% | 751.50 | 0.71 | 861 | — | 7.9% | 758.71 | 0.633 | 849 | — | 6.5% |
| P-n76-k4 | 593 | 541.62 | 0.30 | 593* | — | 1.0% | 542.91 | 0.83 | 593* | — | 0.7% | 566.62 | 0.796 | 595 | — | 1.1% |
| P-n76-k5 | 627 | 566.88 | 0.28 | 691 | — | 12.0% | 568.55 | 0.87 | 667 | — | 8.4% | 589.36 | 0.746 | 628 | — | 2.5% |
| P-n101-k4 | 681 | 628.50 | 1.24 | 681* | — | 1.1% | 634.59 | 1.91 | 681* | — | 1.1% | 660.46 | 1.465 | 681* | **10784.9** | — |