The Capacitated Vehicle Routing Problem: Stronger Bounds in Pseudo-Polynomial Time

Adam N. Letchford[*]   Juan-José Salazar González[†]

Abstract

The *Capacitated Vehicle Routing Problem* (CVRP) is a classic combinatorial optimization problem for which many heuristics, relaxations and exact algorithms have been proposed. Since the CVRP is $\mathcal{NP}$-hard in the *strong* sense, a natural research topic is relaxations that can be solved in *pseudo-polynomial* time. We consider several old and new relaxations of this kind, all of which are based on column generation. We also analyze the effect of adding some known inequalities. Computational experiments demonstrate that the best of our new relaxations yields extremely tight lower bounds.

Keywords: vehicle routing, integer programming, column generation.

1 Introduction

Vehicle routing problems form a much-studied family of combinatorial optimization problems, due to their many important practical applications (see, e.g., [11, 18, 33] for surveys). This paper is concerned with the *capacitated* vehicle routing problem (CVRP), which Dantzig and Ramser [4] defined as follows. A fleet of identical vehicles, each with a known (positive integer) capacity Q, is located at a depot. There are n customers that require service. For $i = 1, \ldots, n$, customer i has a known integer demand q_i with $0 < q_i \le Q$. The (positive) cost of travel between any pair of customers, or between any customer and the depot, is also known. The task is to find a minimum-cost collection of vehicle routes, each starting and ending at the depot, such that each customer is visited by exactly one vehicle, and no vehicle visits a set of customers whose total demand exceeds Q.

Several different integer programming formulations have been proposed for the CVRP. These include so-called *two-index* and *three-index vehicle-flow* formulations, *single-commodity*, *two-commodity* and *multi-commodity flow*

---

[*]Department of Management Science, Lancaster University, UK
[†]Department of Mathematics, Universidad de La Laguna, Spain

formulations, and *set partitioning* formulations (see, e.g., [17, 18, 23, 24, 32] for surveys). At present, the most successful exact algorithms for the CVRP are based on set partitioning formulations augmented with various families of cutting planes (see, e.g., [1, 8, 29, 31]).

The CVRP is $\mathcal{NP}$-hard in the *strong* sense, since it contains the well-known Travelling Salesman and Bin Packing problems as special cases. This leads us to ask the following question: what is the strongest linear programming relaxation of the CVRP that can be solved in *pseudo-polynomial* time (i.e., in time bounded by a polynomial in n and Q)? After surveying the known relaxations with this property, we propose several new ones, and prove that the best one dominates all of the others. Computational experiments show that, in practice, this best relaxation yields remarkably tight lower bounds.

The structure of the paper is as follows. The literature is reviewed in Section 2. The new formulations and relaxations are presented in Section 3. Theoretical results, concerned with the strength of the new relaxations and their solvability in pseudo-polynomial time, are given in Section 4. Computational results are given in Section 5, and concluding remarks are made in Section 6.

Throughout the paper, we assume that the travel costs are asymmetric and the fleet is unlimited. Our approach can easily be adapted to the cases with symmetric costs and/or with a limited number of vehicles. We also use the following notation. We have a complete directed graph G with node set $V = \{0, 1, \ldots, n\}$ and arc set A. We write V_c for $V \setminus \{0\}$, the set of customer nodes, and A_c for the arcs $(i, j) \in A$ with $i, j \in V_c$. For any $i, j \in V_c$ with $i \neq j$, we exclude (i, j) and (j, i) from A if $q_i + q_j > Q$. Node 0 represents the depot, and nodes $1, \ldots, n$ represent customers. By convention, $q_0 = 0$. For any $S \subset V$, we use $\delta^+(S)$ (respectively, $\delta^-(S)$) to denote the set of arcs (i, j) with $i \in S$ and $j \in V \setminus S$ (respectively, with $i \in V \setminus S$ and $j \in S$). If $S = \{i\}$, we just write $\delta^+(i)$ and $\delta^-(i)$. Also, for any set of customers $S \subset V_c$, let $q(S) = \sum_{i \in S} q_i$. The cost of traversing arc $(i, j) \in A$ is c_{ij}. Finally, given a vector $x \in [0, 1]^{|A|}$ and a set $A' \subset A$, we write $x(A')$ for $\sum_{a \in A'} x_a$.

2   Literature Review

Now we review some of the existing formulations for the CVRP. For brevity, we cover only *two-index vehicle flow*, *set partitioning* and *multi-commodity flow* formulations. For details on other formulations, see the surveys mentioned in the introduction.

2.1 The two-index vehicle flow formulation

Laporte & Nobert [19] presented what is now called the *two-index vehicle flow* formulation. For all $(i, j) \in A$, define a binary variable x_{ij}, taking the value 1 if and only if some vehicle traverses the arc (i, j). Then the formulation is:

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

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

$$x(\delta^+(S)) \geq \lceil q(S)/Q \rceil \qquad (S \subseteq V_c). \tag{3}$$

$$x_{ij} \in \{0, 1\} \qquad ((i, j) \in A) \tag{4}$$

The constraints (3) are called *rounded capacity* (RC) inequalities [28]. The complexity of separation for them is unknown, but effective heuristics are available [25]. Moreover, separation can be solved efficiently for the following weaker inequalities [28]:

- The *fractional capacity* (FC) inequalities:

$$x(\delta^+(S)) \geq \frac{q(S)}{Q} \qquad (S \subseteq V_c). \tag{5}$$

- The *subtour elimination* (SE) inequalities:

$$x(\delta^+(S)) \geq 1 \qquad (S \subseteq V_c). \tag{6}$$

Many other families of inequalities have been developed for the 2-index formulation (see [23, 25, 28, 32] for surveys). Of interest to us are:

- The *generalized large multistar* (GLM) inequalities (see [13]):

$$x(\delta^+(S)) \geq \frac{1}{Q} \sum_{i \in S} \left(q_i + \sum_{j \in V_c \setminus S} q_j (x_{ij} + x_{ji}) \right) \qquad (S \subseteq V_c). \tag{7}$$

- The *knapsack large multistar* (KLM) inequalities (see [22]):

$$x(\delta^+(S)) \geq \frac{1}{\beta} \sum_{i \in S} \left(\alpha_i + \sum_{j \in V_c \setminus S} \alpha_j (x_{ij} + x_{ji}) \right) \qquad (S \subseteq V_c), \tag{8}$$

where $\alpha \in \mathbb{R}_+^n$ and $\beta \in \mathbb{R}_+$ are such that the inequality $\alpha^T y \leq \beta$ is valid for the following 0-1 knapsack polytope:

$$\text{conv} \left\{ y \in \{0, 1\}^n : \sum_{i \in V_c} q_i y_i \leq Q \right\}. \tag{9}$$

The GLM inequalities dominate the FC inequalities, and the KLM inequalities include the GLM and SE inequalities as special cases. The separation problem for the GLM inequalities can be solved in polynomial time [25]. The complexity of separation for the KLM inequalities is unknown.

3

2.2 Set partitioning formulations

The *set partitioning* formulation is due to Balinski & Quandt [2]. Let Ω denote the set of possible routes for a single vehicle, and let z_r for each $r \in \Omega$ be a binary variable taking the value 1 if and only if that route is used. Define the constant a_{ir} for each $i \in V_c$ and $r \in \Omega$, taking the value 1 if customer i is served in route r, and 0 otherwise. Finally let c_r denote the cost of route r. Then the formulation is:

$$\min \quad \sum_{r \in \Omega} c_r z_r \tag{10}$$

$$\text{s.t.} \quad \sum_{r \in \Omega} a_{ir} z_r = 1 \qquad (i \in V_c) \tag{11}$$

$$z_r \in \{0, 1\} \qquad (r \in \Omega). \tag{12}$$

Since $|\Omega|$ can be exponential in n, a column-generation technique is necessary to efficiently solve the LP relaxation of this formulation. The column generation requires solving a pricing subproblem which is itself strongly $\mathcal{NP}$-hard. Foster & Ryan [7] noted that pricing becomes easier if one enlarges Ω by permitting routes in which vehicles may visit customers more than once (now called *non-elementary* routes). The pricing subproblem can then be solved in $O(n^2 Q)$ time by dynamic programming. See, e.g., [8, 29, 27] for details.

We will call the formulation with only elementary routes "SPE", and the one with non-elementary routes "nSP". As noted in [8], the x variables of the two-index formulation can be linked to the z variables of nSP via the equations

$$x_{ij} = \sum_{r \in \Omega} b_{ijr} z_r \qquad ((i,j) \in A), \tag{13}$$

where the constant b_{ijr} counts the number of times arc (i,j) is traversed on route r. (Note that b_{ijr} can be larger than 1 when route r is non-elementary.) This enables one to strengthen the LP relaxation of nSP by adding valid inequalities that involve the x variables.

In [23] we proved that the projection of the LP relaxation of SPE into x-space satisfies all KLM inequalities, whereas the projection for nSP satisfies only a subset of them that includes the GLM inequalities but not the SE inequalities. This means that it is worth adding RC, SE or KLM inequalities to nSP, but not FC or GLM inequalities.

2.3 Multi-commodity flow formulations

As mentioned above, several commodity-flow formulations are analysed in [23, 24]. For brevity, we focus on just one multi-commodity flow formulation, called MCF2b in [24]. It involves binary variables f_{ij}^k and g_{ij}^k, indicating whether a vehicle traverses the arc (i,j) on the way to customer k or after

visiting customer k, respectively. It has the objective (1) and the constraints (2), (4) from the two-index vehicle-flow formulation, together with the following constraints:

$$f^k(\delta^+(0)) = f^k(\delta^-(k)) = g^k(\delta^+(k)) = g^k(\delta^-(0)) = 1 \qquad (k \in V_c) \qquad (14)$$

$$f^k(\delta^-(0)) = f^k(\delta^+(k)) = g^k(\delta^-(k)) = g^k(\delta^+(0)) = 0 \qquad (k \in V_c) \qquad (15)$$

$$f^k(\delta^-(l)) = f^k(\delta^+(l)) = g^l(\delta^-(k)) = g^l(\delta^+(k)) \qquad (k, l \in V_c : l \neq k) \qquad (16)$$

$$\sum_{k \in V_c \setminus \{i,j\}} q_k(f_{ij}^k + g_{ij}^k) \leq (Q - q_i - q_j)x_{ij} \qquad ((i,j) \in A) \qquad (17)$$

$$f_{ij}^k + g_{ij}^k \leq x_{ij} \qquad (k \in V_c, (i,j) \in A) \qquad (18)$$

$$f_{ij}^k, g_{ij}^k \in \{0,1\} \qquad (k \in V_c, (i,j) \in A) \qquad (19)$$

It is shown in [24] that the LP relaxation of MCF2b dominates the LP relaxation of all other known polynomial-sized commodity-flow formulations for the CVRP with general demands. (See [12] for alternative formulations for the CVRP with *unit* demands.) It is also shown in [24] that the LP relaxation of MCF2b satisfies all of the SE and GLM inequalities.

One big drawback however is that MCF2b has $O(n^3)$ variables and constraints, which makes even solving the LP relaxation time-consuming.

MCF2b was strengthened in [24] by adding exponentially many variables, as follows. Let Π be the set of all possible loading patterns of a single vehicle. That is, each member of Π is a subset of V_c whose total demand does not exceed Q. Observe that each member of Π corresponds to an extreme point of the knapsack polytope (9). For each $(i,j) \in A$ and each $P \in \Pi$, let σ_{ij}^P be a binary variable indicating whether a vehicle departs from the depot with loading pattern P and traverses the arc (i,j) at some point. (This implies that $i \in P$ when $i \neq 0$, and that $j \in P$ when $j \neq 0$.) Then we add the following constraints to MCF2b:

$$x_{ij} = \sum_{P \in \Pi: \, i,j \in P} \sigma_{ij}^P \qquad ((i,j) \in A_c) \qquad (20)$$

$$f_{ij}^k + g_{ij}^k = \sum_{P \in \Pi: \, i,j,k \in P} \sigma_{ij}^P \qquad ((i,j) \in A_c, k \in V_c \setminus \{i,j\}). \qquad (21)$$

Accordingly, we called the strengthened formulation "MCF2K". It is shown in [24] that one can price the σ variables in $O(n^3 Q)$ time, by reduction to a series of knapsack problems. It was also shown that the LP relaxation of MCF2K satisfies the KLM inequalities (8), which is not true for MCF2b in general. Unfortunately, solving the LP relaxation of MCF2K is even more time-consuming than solving that of MCF2b.

Finally, we mention that a different way of strengthening MCF2b, using additional *linear ordering* variables, can be found in a recent paper by Leggieri and Haouari [20].

3 New Formulations and Relaxations of the CVRP

In this section, we present new formulations and relaxations of the CVRP. Our starting point, in Subsection 3.1, is a new formulation of the CVRP that we call the *arc-packing* formulation. Some equivalent formulations are discussed in Subsection 3.2. Then, in Subsection 3.3, we present two additional formulations.

3.1 The arc-packing formulation

Before presenting our new formulation, we will need the following lemma:

Lemma 1 *The LP relaxation of* `MCF2b` *is unchanged if we replace the equations (14) with the equations:*

$$x_{ij} = f_{ij}^j \qquad (j \in V_c, \, i \in V \setminus \{j\}) \tag{22}$$

$$x_{ij} = g_{ij}^i \qquad (i \in V_c, \, j \in V \setminus \{i\}). \tag{23}$$

We can also remove constraints (15) by eliminating all f and g variables involved.

Proof. For a given $k \in V_c$, the $n+1$ flow-conservation equations involving the f^k variables are linearly dependent. In particular, any one of them is implied by the other n. This allows us to remove the equations involving the depot in (14) from the system (14)–(16). Now, it is shown in Subsection 3.2 of [24] that the equations (22) and (23) are satisfied by all solutions to the LP relaxation. So, we can add them. Once they are added, the remaining equations in (14) are equivalent to the equations (2). So, they can be deleted. □

Now, from Lemma 1, we can write `MCF2b` with the constraints (2), (4), (15)–(19), (22) and (23). Now, note that the constraints (4), (17)–(19), (22) and (23) decompose into $n(n+1)$ independent blocks, one for each arc $(i,j) \in A$. Thus, it is natural to apply Dantzig-Wolfe decomposition [5], putting those constraints into the subproblems and leaving constraints (2) and (16) in the master problem.

Now consider the constraint block for a fixed arc (i,j). Note that the variables involved in the block are x_{ij} and the f_{ij}^k and g_{ij}^k. Moreover, in any feasible solution to the block, the set of indices k such that $f_{ij}^k + g_{ij}^k = 1$ must correspond to a feasible loading of the vehicle.

To explain this more precisely, we will need a little more notation. For a given arc $(i,j) \in A$ and a given pair (S,T) with $S, T \subseteq V_c$, let us say that (S,T) is *compatible* with (i,j) if it satisfies the following conditions:

- $S, T \subseteq V_c$, $S \cap T = \emptyset$, $S \cup T \neq \emptyset$;

- $q(S) + q(T) \leq Q$;

- if $j = 0$ then $T = \emptyset$, otherwise $j \in T$;

- if $i = 0$ then $S = \emptyset$, otherwise $i \in S$.

Note that, if $j = 0$ then $i \in S \in \Pi$. Similarly, if $i = 0$ then $j \in T \in \Pi$. Otherwise, $S \cup T \in \Pi$ with $i \in S$ and $j \in T$. Let $\mathcal{F}_{ij}$ denote the set of all pairs compatible with (i, j).

The variables in the master problem are then obtained as follows. For all $(i, j) \in A$ and all $(S, T) \in \mathcal{F}_{ij}$, let Φ_{ij}^{ST} be a binary variable, taking the value 1 if and only if a vehicle traverses arc (i, j) having already visited the customers in S (and no others) and being about to visit the customers in T (and no others). The objective function in the master problem is then:

$$\min \sum_{(i,j) \in A} c_{ij} \sum_{(S,T) \in \mathcal{F}_{ij}} \Phi_{ij}^{ST}.$$

The master contains the following constraints, which come from equations (2):

$$\sum_{j \in V \setminus \{i\}} \sum_{(S,T) \in \mathcal{F}_{ij}} \Phi_{ij}^{ST} = \sum_{j \in V \setminus \{i\}} \sum_{(S,T) \in \mathcal{F}_{ji}} \Phi_{ji}^{ST} = 1 \qquad (i \in V_c).$$

It also has the following constraints, which come from equations (16):

$$\sum_{j \in V \setminus \{l\}} \sum_{(S,T) \in \mathcal{F}_{jl} : k \in T} \Phi_{jl}^{ST} = \sum_{j \in V_c \setminus \{l\}} \sum_{(S,T) \in \mathcal{F}_{lj} : k \in T} \Phi_{lj}^{ST} = \sum_{j \in V_c \setminus \{k\}} \sum_{(S,T) \in \mathcal{F}_{jk} : l \in S} \Phi_{jk}^{ST} =$$

$$\sum_{j \in V \setminus \{k\}} \sum_{(S,T) \in \mathcal{F}_{kj} : l \in S} \Phi_{kj}^{ST} \qquad\qquad (k, l \in V_c : l \neq k).$$

$$(24)$$

Finally, it also has the binary conditions:

$$\Phi_{ij}^{ST} \in \{0, 1\} \qquad\qquad ((i, j) \in A,\ (S, T) \in \mathcal{F}_{ij}). \qquad (25)$$

Since the Φ variables are concerned with the "packing" of the vehicle on each arc traversed, we call this new formulation an *arc-packing* formulation and denote it by "AP". Note that AP has $O(n^2)$ constraints. By definition, the LP relaxation of AP dominates that of MCF2b. We will show in the next section that it dominates that of MCF2K as well.

3.2 Alternative representations of the arc-packing formulation

A concept that will simplify the proofs in the following section is that of an *explicit master* (see [30]). An explicit master is a formulation that contains

both the original variables and those obtained by Dantzig-Wolfe decomposition. In our case, the explicit master of `AP` can be obtained simply by taking `MCF2b` and adding the constraints (25), along with the following linking constraints:

$$x_{ij} = \sum_{(S,T) \in \mathcal{F}_{ij}} \Phi_{ij}^{ST} \qquad ((i,j) \in A) \qquad (26)$$

$$f_{ij}^k = \sum_{(S,T) \in \mathcal{F}_{ij}:k \in T} \Phi_{ij}^{ST} \qquad ((i,j) \in A, k \in V_c \setminus \{i,j\}) \qquad (27)$$

$$g_{ij}^k = \sum_{(S,T) \in \mathcal{F}_{ij}:k \in S} \Phi_{ij}^{ST} \qquad ((i,j) \in A, k \in V_c \setminus \{i,j\}). \qquad (28)$$

It follows from the main result in [30] that the LP relaxation of the explicit master is of equal strength to that of the standard master.

It is also possible to construct an 'intermediate' formulation, which has x and Φ variables but no f and g variables. This formulation consists of the objective function (1) and the constraints (2), (24), (25) and (26). We will call this third formulation the *semi-explicit* master problem of `AP`. Using exactly the same argument that was used in [30], one can show that the LP relaxation of the semi-explicit master is of equal strength to those of the other two formulations.

3.3 Other formulations

Now, recall the definition of the formulation `nSP` from Subsection 2.2. One can obtain a "hybrid" of `nSP` and `AP` simply by taking the semi-explicit master of `AP` and adding the constraints (12) and (13). We call this formulation "nSP+AP". By definition, the LP relaxation of `nSP+AP` is at least as strong as the LP relaxations of either `nSP` or `AP`. (Of course, this comes at the cost of increasing the number of pricing problems that need to be solved.)

For the sake of completeness, we briefly mention one last formulation. It is obtained by adding the SE inequalities to the explicit master of `nSP`. We call it "nSP+SE". Note that `nSP+SE` has an exponential number of both variables and constraints.

4 Some Theoretical Results

In this section, we prove some theoretical results concerned with the LP relaxations of the formulations presented in the previous section. In Subsection 4.1, we compare the new relaxations with existing relaxations in terms of strength. In Subsection 4.2, we show that the relaxations of `AP`, `nSP+AP` and `nSP+SE` can be solved in pseudo-polynomial time.

4.1 The strength of the new relaxations

As mentioned at the end of Subsection 3.1, the LP relaxation of `AP` is at least as strong as that of `MCF2b`, the strongest known polynomial-sized multicommodity flow formulation. The following proposition shows something stronger.

Proposition 1 *The LP relaxation of `AP` is at least as strong as that of `MCF2K`.*

Proof. As mentioned above, the LP relaxation of the explicit master of `AP` is of equal strength to that of `AP` itself. The only difference between the LP relaxation of the explicit master and that of `MCF2K` is that the former has constraints (26)–(28) (plus non-negativity on the Φ variables), whereas the latter has constraints (20) and (21) (plus non-negativity on the σ variables. Note however that we have the following identities:

$$\sigma_{ij}^P = \sum_{(S,T)\in\mathcal{F}_{ij}:S\cup T=P} \Phi_{ij}^{ST} \qquad ((i,j)\in A, P\in\Pi).$$

Under this mapping, (i) (26) is equivalent to (20), (ii) (27) and (28) imply (21), and (iii) non-negativity on Φ implies non-negativity on σ. $\square$

Proposition 1 has the following corollary.

Corollary 1 *The LP relaxation of `AP` satisfies all KLM inequalities (and therefore all FC, SE and GLM inequalities).*

Proof. It was shown in Section 4 of [24] that the LP relaxation of `MCF2K` satisfies all KLM inequalities. $\square$

We will see in Section 5 that, in fact, the LP relaxation of `AP` is stronger than that of `MCF2K` on many instances. An informal explanation of this fact is that `MCF2K` is obtained by "convexifying" over 0-1 knapsack polytopes of the form:

$$\text{conv}\left\{(f,g)\in\{0,1\}^{2n} : \sum_{k\in V_c} q_k(f_k+g_k)\leq Q\right\},$$

whereas `AP` is obtained by "convexifying" over smaller polytopes of the form

$$\text{conv}\left\{(f,g)\in\{0,1\}^{2n} : \sum_{k\in V_c} q_k(f_k+g_k)\leq Q,\ f_k+g_k\leq 1\ (k\in V_c)\right\}.$$

These latter polytopes can be viewed as 0-1 *multiple-choice* knapsack polytopes (see e.g. [16]).

We remark that, in the unit-demand case, the 0-1 multiple-choice knapsack polytopes have the "integrality property" in the sense of Geoffrion [10]. From this it can be shown that the LP relaxations of MCF2b, MCF2K and AP have equal strength in the unit-demand case.

Now, recall the formulation nSP+AP from Subsection 3.3. By definition, the lower bound associated with nSP+AP dominates those obtained from nSP and AP (and therefore also MCF2b and MCF2K). A natural question is how it compares with the lower bound associated with SPE. The following theorem settles this question:

Theorem 1 *If z^* is a solution to the LP relaxation of SPE, then there exists a solution $(x^*, f^*, g^*, z^*, \Phi^*)$ to the LP relaxation of nSP+AP that has the same cost.*

Proof. We showed in Theorem 5 of [24] how to construct the triple (x^*, f^*, g^*), so it suffices to show how to construct Φ^*. For any $(i, j) \in A$, $r \in \Omega$ and $(S, T) \in \mathcal{F}_{ij}$, let t_{ijr}^{ST} be a binary constant which takes the value 1 if and only if a vehicle following the elementary route r departs from the depot, visits the customers in S, traverses the arc (i, j), visits the customers in T, and finally returns to the depot. The desired vector Φ^* is then created by setting Φ_{ij}^{ST} to the value $\sum_{r \in \Omega} t_{ijr}^{ST} z_r^*$ for all $(i, j) \in A$ and $(S, T) \in \mathcal{F}_{ij}$. $\square$

Theorem 1 implies that the lower bound from nSP+AP is dominated by the lower bound from SPE. We stress however that the former bound can be computed in pseudo-polynomial time, whereas the latter bound cannot (unless $\mathcal{P} = \mathcal{NP}$).

As for the formulation nSP+SE, the fact that AP satisfies all SE inequalities implies that the bound from nSP+AP dominates the one from nSP+SE.

To aid the reader, we display in Figure 1 a hierarchy of formulations. An arrow from one class to another means that the lower bound from the latter is at least as strong as the one for the former. By "RC", we mean the 2-index formulation (1)-(4), which is based on the RC inequalities (3). By "FC" and "GLM", we mean the formulations obtained from the 2-index formulation by replacing the RC inequalities with the FC inequalities (5) and GLM inequalities (7), respectively. Finally, "FC+SE" and "GLM+SE" indicate that the SE inequalities (6) have been added.

4.2   Solving the new relaxations

Now we consider the time taken to solve the new LP relaxations.

Theorem 2 *The LP relaxation of AP can be solved in pseudo-polynomial time.*
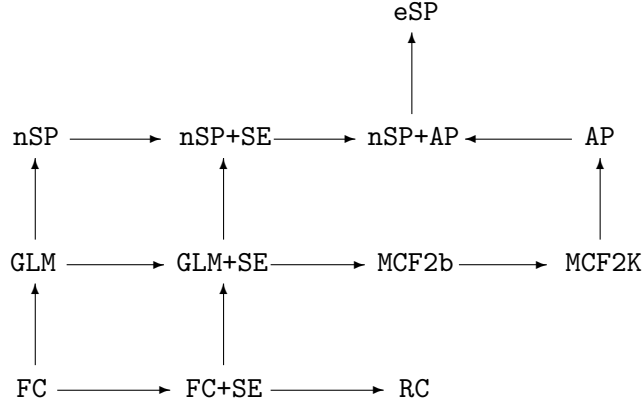
Figure 1: Hierarchy of CVRP formulations.

Proof. It suffices to show that the LP relaxation of the associated *explicit master* (see Subsection 3.2) can be solved in pseudo-polynomial time. Due to the polynomial equivalence between separation and optimization [15], along with LP duality, it follows that this can be done if and only if the pricing problem for the Φ variables can be solved in pseudo-polynomial time.

Now observe that the only constraints in the explicit master that involve the Φ variables (apart from the binary conditions) are equations (26)–(28). Let α, β and γ denote the dual prices of these equations, respectively. (One can think of β_{ij}^k and γ_{ij}^k as representing estimates of the potential benefit of visiting customer k after traversing the arc (i, j), or before traversing it, respectively.) The reduced cost of a column Φ_{ij}^{ST} is

$$-\alpha_{ij} - \sum_{k \in T} \beta_{ij}^k - \sum_{k \in S} \gamma_{ij}^k.$$

For a fixed arc (i, j), the problem of finding sets S and T leading to the minimum reduced cost can be modelled by introducing binary variables u_k and v_k for all $k \in V_c \setminus \{i, j\}$, with the meaning $u_k = 1$ iff $k \in S$ and $v_k = 1$ iff $k \in T$, respectively. We then have to solve:

$$\max \sum_{k \in V_c \setminus \{i,j\}} (\beta_{ij}^k v_k + \gamma_{ij}^k u_k)$$

$$\text{s.t.} \sum_{k \in V_c \setminus \{i,j\}} q_k (u_k + v_k) \le Q - q_i - q_j$$

$$u_k + v_k \le 1 \qquad (k \in V_c \setminus \{i, j\})$$

$$u_k, v_k \in \{0, 1\} \qquad (k \in V_c \setminus \{i, j\}).$$

This is a 0-1 multiple-choice knapsack problem and it aims at determining $S \setminus \{i\}$ and $T \setminus \{j\}$. It can be reduced to a standard 0-1 knapsack problem

11

(KP) by setting the item set to $V_c \setminus \{i,j\}$, the capacity of the knapsack to $Q - q_i - q_j$, the weight of item k to q_k, and the profit of item k to the maximum of β_{ij}^k and γ_{ij}^k. This 0-1 KP can be solved in $O(nQ)$ time via dynamic programming [3]. A column of negative cost exists, for the given arc (i,j), if and only if the profit of the 0-1 KP solution plus $\alpha_{ij} + \beta_{ij}^j + \gamma_{ij}^i$ is positive. $\qquad\square$

We now make some remarks about Theorem 2.

- If one uses dynamic programming to price the Φ variables, it takes $O(n^3 Q)$ time, since a 0-1 KP must be solved for each arc. By comparison, pricing the z variables in nSP takes $O(n^2 Q)$ time (see Subsection 2.2).

- In practice, one can usually solve the 0-1 KPs much more quickly using a branch-and-bound algorithm, such as the one in Martello & Toth [26].

- The LP relaxation of the *semi-explicit* master of AP can be solved in a similar way. Let $(i,j) \in A$ be given, let (ρ, τ, σ) be the dual vectors for the three families of equations (24), and let α be as before. To price the Φ variables, it suffices to set

$$
\begin{aligned}
\beta_{ij}^k &= \rho_{kj} - \rho_{ki} + \tau_{ki} \\
\gamma_{ij}^k &= \sigma_{jk} - \sigma_{ik} - \tau_{jk},
\end{aligned}
$$

and then proceed as in the proof of Theorem 2.

Finally, we consider the other two formulations that were presented in Subsection 3.3.

Proposition 2 *The LP relaxation of nSP+AP can be solved in pseudo-polynomial time.*

Proof. This follows from the fact that the pricing subproblems for the z and Φ variables can be solved in $O(n^2 Q)$ time and $O(n^3 Q)$ time, respectively. $\square$

Proposition 3 *The LP relaxation of nSP+SE can be solved in pseudo-polynomial time.*

Proof. Since the pricing subproblem for nSP can be solved in pseudo-polynomial time, we can separate over the associated polyhedron (in (x,z)-space) in pseudo-polynomial time. Together with the fact that the separation problem for the SE inequalities can be solved in polynomial time, this implies that the separation problem for the LP relaxation of nSP+SE can also be solved in pseudo-polynomial time. This in turn implies the result. $\square$

5 Computational Experiments

In this section, we present the results of some computational experiments. We describe the test instances in Subsection 5.1. In Subsection 5.2, we present results for five relaxations that can be solved in pseudo-polynomial time. Then, in Subsection 5.3, we examine what happens when certain cutting planes (RC inequalities) are added to the relaxations.

All experiments were conducted on a Dell Precision T5400 desktop computer, with a 3.16 GHz Intel Xeon X5460 processor, running Windows 10 64-bit. Our code, which involves both column and row generation, was written in Microsoft Visual C, using IBM ILOG Cplex 12.6 as LP solver. Primal and dual simplex were used to re-optimize after adding columns and rows, respectively.

5.1 Test instances

We used similar instances to those used in [24]. These instances have $n = 16$, which is small by today's standards (see [31, 34]), but big enough to establish dominance relationships between the various bounds. There are two kinds of instances: asymmetric and symmetric. In the asymmetric instances, the costs c_{ij} were random integers uniformly distributed between l and 500. In the symmetric instances, the costs were obtained by computing the Euclidean distance between locations randomly distributed in a square of width 500, and then rounding to the nearest integer. (The depot was located in the center of the square.) We created instances with general demands (random integers in the range $[25, 33]$) and instances with only unit demands. For the instances with general demands, we considered $Q \in \{100, 150, 200\}$. For the instances with unit demands, we considered $Q \in \{4, 6, 8\}$. This led to twelve families of instances, and for each family we generated 20 instances.

These 240 instances are available in the VRP-REP repository. We remark that the instances with $Q = 100$ are surprisingly hard to solve for their size, due no doubt to the fact that the "bin packing" aspect of the CVRP plays a key role in such instances.

5.2 Results with five relaxations

We began by considering five LP relaxations that can be solved in pseudo-polynomial time. These are the relaxations of the following formulations:

- `nSP`: the set partitioning formulation (10)–(12), with non-elementary routes permitted.

- `nSP+SE`: the same, but with SE inequalities (6) added (see Subsection 3.3).

- `MCF2K`: the best multi-commodity flow formulation from [24].

Relaxation	Variables	Constraints
nSP	x_{ij} , z_r	(2), (11), (13)
nSP+SE	x_{ij} , z_r	(2), (11), (13), (6)
MFC2K	x_{ij} , f_{ij}^k , g_{ij}^k, σ_{ij}^P	(2), (14)–(21)
AP	x_{ij} , Φ_{ij}^{ST}	(2), (24), (26)
nSP+AP	x_{ij} , z_r , Φ_{ij}^{ST}	(2), (11), (13), (24), (26)

Table 1: Variables and constraints involved in the first five relaxations.

- **AP**: the arc-packing formulation.

- **nSP+AP**: the formulation combining both set partitioning and arc-packing variables described in Subsection 3.3.

Table 1 shows the variables and constraints that are present in these relaxations (where non-negativity on all variables is assumed). For ease of implementation, we include the x variables in all relaxations (that is, we use explicit or semi-explicit master LPs). For the same reason, we include (2) in all relaxations, even if they are redundant in some cases. We also include in all relaxations the RC inequality on the set V_c, i.e., the inequality $x(\delta^+(0)) \geq \lceil q(V_c)/Q \rceil$. This inequality lower bounds the number of vehicles leaving the depot. Although it was not included in the relaxations analyzed in [24], we found that it tends to slightly improve the lower bounds from all five relaxations.

We used a time limit of 2 hours for each instance and each relaxation. Table 2 displays, for each set of 20 instances and each of the five relaxations, the average ratio between the resulting lower bound and the optimum, expressed as a percentage. The letters 'A' and 'S' in the first column indicate asymmetric and symmetric instances, respectively. The letters 'G' and 'U' refer to general and unit demands. (For one of the A-G-200 instances, we were unable to solve the last relaxation within the time limit. For this reason, we give an interval instead of a precise figure. (The lower and upper limits were obtained by computing near-optimal dual and primal solutions, respectively, whenever the relaxation could not be solved exactly within the 2 hours.)

We see that all five relaxations are rather strong. Interestingly, however, **AP** gives a significantly stronger lower bound than **nSP** for symmetri instances, but the reverse is true for asymmetric instances (with the exception of A-U-8). We do not have an explanation for this phenomenon.

Table 3 displays, for the same set of instances, the average amount of time taken (in seconds) to solve the five relaxations. (An entry of "T.L." indicates that the given relaxation could not be solved exactly for all 20 instances within the time limit.) We see that the last three relaxations took

Type	nSP	nSP+SE	MCF2K	AP	nSP+AP
A-G-100	99.65	99.65	98.86	99.38	99.66
A-G-150	97.84	97.94	96.23	96.45	98.27
A-G-200	97.83	98.30	97.58	97.60	[98.37, 98.41]
A-U-4	98.78	98.78	98.08	98.08	98.88
A-U-6	98.12	98.40	97.69	97.69	98.61
A-U-8	98.07	98.52	98.51	98.51	98.85
S-G-100	95.16	97.30	95.10	97.87	97.87
S-G-150	90.94	96.89	96.36	97.18	97.26
S-G-200	88.46	96.86	97.12	97.41	97.48
S-U-4	92.78	96.87	97.32	97.32	97.44
S-U-6	88.79	96.98	97.15	97.15	97.26
S-U-8	86.28	97.49	98.10	98.10	98.14

Table 2: Average percentage ratios for five relaxations that can be solved in pseudo-polynomial time.

considerably longer to solve than the first two. We tried some simple tricks to speed up the computation (such as pricing heuristics, dual stabilisation, and the use of an interior-point LP solver), but they did not lead to a significant improvement. An examination of detailed output from the LP solver indicated that the long computing times were almost entirely due to massive primal degeneracy in the master LP. (Indeed, for most instances, over 95% of the pivots were degenerate.) Possible ways of addressing this issue are discussed in the conclusions.

5.3   Results with cutting planes

One can of course obtain stronger relaxations by adding cutting planes. Since the RC inequalities (3) are important cutting planes in practice (see [8, 25]), we experimented with five relaxations: the LP relaxation of the 2-index formulation (1)–(4), which we call "RC", and the relaxations obtained by adding RC inequalities to nSP, MCF2K, AP and nSP+AP. We call the latter four relaxations "nSP+RC", "MCF2K+RC" and so on.

We remark that the separation problem for the RC inequalities has recently been proved to be $\mathcal{NP}$-hard, and is conjectured to be $\mathcal{NP}$-hard in the strong sense [6]. If that is correct, then the five relaxations mentioned cannot be solved in pseudo-polynomial time. Fortunately, since our instances have $n = 16$, we were able to solve the separation problem exactly, by brute-force enumeration.

Table 4 displays the average ratio between the lower bounds and the optimum, again expressed as a percentage. (Again, for one set of instances,

Type	nSP	nSP+SE	MCF2K	AP	nSP+AP
A-G-100	0.02	0.02	4.45	4.99	14.61
A-G-150	0.04	0.04	13.51	36.69	326.04
A-G-200	0.07	0.07	85.54	121.04	T.L.
A-U-4	0.02	0.01	6.45	22.05	88.74
A-U-6	0.03	0.02	17.30	98.29	421.73
A-U-8	0.05	0.04	269.75	412.26	T.L.
S-G-100	0.01	0.02	4.37	3.91	8.49
S-G-150	0.01	0.06	14.20	26.19	223.96
S-G-200	0.02	0.05	62.95	69.19	554.45
S-U-4	0.00	0.02	7.88	14.39	53.98
S-U-6	0.00	0.03	25.80	49.06	258.71
S-U-8	0.01	0.03	163.08	69.83	1281.69

Table 3: Average times (in seconds) for five relaxations that can be solved in pseudo-polynomial time.

this time A-U-8, one of the figures is given as an interval.) We see that all five relaxations are very strong. As before, `AP` performs better than `nSP` for symmetric instances, but worse for asymmetric instances. Also, a comparison of Table 4 with Table 2 shows that `nSP+AP` consistently gives better bounds than `RC` on symmetric instances, but worse bounds on asymmetric ones. Again, we are not sure why.

Finally, Table 5 displays the running times. One result that may seem surprising is that `nSP+AP+RC` sometimes takes less time to be solved than the weaker `nSP+AP`. The explanation is that, when solving `nSP+AP+RC`, we gave cutting priority over pricing. In other words, in each major iteration, pricing was invoked only when no more violated RC inequalities could be found. This strategy seems to have pushed the x variables to "promising" values early on, leading to a substantial decrease in the number of pricing calls.

6 Conclusion

In this paper, we have considered five CVRP relaxations that can be solved in pseudo-polynomial time, two of which are completely new. The strongest one yields remarkably tight lower bounds, especially in the case of symmetric CVRP instances. On the other hand, the time taken to solve the new relaxations was excessive.

We can think of several interesting topics for future research. The most pressing one is whether the new relaxations can somehow be solved more

Type	RC	nSP+RC	MCF2K+RC	AP+RC	nSP+AP+RC
A-G-100	92.61	99.79	99.06	99.46	99.79
A-G-150	94.12	98.23	96.66	96.79	98.46
A-G-200	98.05	98.91	98.58	98.59	98.99
A-U-4	96.89	99.56	99.01	99.01	99.59
A-U-6	98.75	99.32	99.15	99.15	99.44
A-U-8	98.62	98.85	98.95	98.95	[99.05,99.07]
S-G-100	98.70	99.39	99.08	99.59	99.59
S-G-150	99.62	99.67	99.67	99.70	99.72
S-G-200	99.64	99.70	99.70	99.72	99.72
S-U-4	99.98	100.00	100.00	100.00	100.00
S-U-6	99.95	99.95	99.96	99.96	99.96
S-U-8	100.00	100.00	100.00	100.00	100.00

Table 4: Average percentage ratios with exact separation of RC inequalities.

Type	RC	nSP+RC	MCF2K+RC	AP+RC	nSP+AP+RC
A-G-100	0.00	0.20	39.76	6.65	13.75
A-G-150	0.00	1.42	82.02	43.97	357.49
A-G-200	0.00	0.47	53.48	109.90	T.L.
A-U-4	0.00	0.18	23.27	26.61	79.55
A-U-6	0.00	0.20	34.93	96.68	528.61
A-U-8	0.00	0.37	106.40	227.22	T.L.
S-G-100	0.01	0.37	46.75	10.46	22.46
S-G-150	0.01	2.92	70.50	33.57	210.12
S-G-200	0.00	1.34	50.60	66.79	445.19
S-U-4	0.01	0.27	5.90	25.98	68.03
S-U-6	0.01	0.30	25.75	61.45	237.61
S-U-8	0.00	1.25	63.98	108.26	1078.48

Table 5: Average times (in seconds) with exact separation of RC inequalities.

quickly, so that they can be used to compute strong bounds for instances of larger size (such as those described in [34]). Perhaps one of the recently-developed techniques for alleviating primal degeneracy, surveyed in [9], could be useful. Alternatively, one could apply Lagrangian relaxation instead of Dantzig-Wolfe decomposition to MCF2b, and then solve the Lagrangian dual via a method with proven fast convergence (see, e.g., [14, 21]).

Other interesting topics are the derivation of strong valid inequalities that involve the new arc packing variables, the derivation of specialised branching rules that involve the new variables, and explaining why some formulations work better for symmetric instances, whereas some work better for asymmetric ones. Finally, it may be that arc packing formulations can be useful for generalisations of the CVRP in which the cost of traversing each arc depends on the commodities that the vehicle is carrying at the time.

References

[1] Baldacci, R., Christofides, N., Mingozzi, A.: An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Math. Program. **115**, 351–385 (2008)

[2] Balinski, M.L., Quandt, R.E.: On an integer program for a delivery problem. Oper. Res. **12**, 300–304 (1964)

[3] Bellman, R.E.: *Dynamic Programming.* Princeton, NJ: Princeton University Press (1957)

[4] Dantzig, G.B., Ramser, R.H.: The truck dispatching problem. Mgmt. Sci. **6**, 80–91 (1959)

[5] Dantzig, G.B., Wolfe, P.: Decomposition principle for linear programs. Oper. Res. **8**, 101–111 (1960)

[6] Diarrassouba, I.: On the complexity of the separation problem for rounded capacity inequalities. Discr. Optim. **25**, 86–104 (2017)

[7] Foster, B.A., Ryan, D.M.: An integer programming approach to the vehicle scheduling problem. Oper. Res. Quart. **27**, 367–384 (1976)

[8] Fukasawa, R., Lysgaard, J., de Aragão, M.P., Reis, M., Uchoa, E., Werneck, R.F.: Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Math. Program. **106**, 491–511 (2006)

[9] Gauthier, J.B., Desrosiers, J., Lübbecke, M.E.: Tools for primal degenerate linear programs: IPS, DCA, and PE. EURO J. Transp. Logist. **5**, 161–204 (2016)

[10] Geoffrion, A.M.: Lagrangean relaxation for integer programming. Math. Program. Study **2**, 82–114 (1974)

[11] Golden, B.L., Raghavan, S., Wasil, E.A. (eds.): *The Vehicle Routing Problem: Latest Advances and New Challenges.* New York: Springer (2008)

[12] Godinho, M.T., Gouveia, L., Magnanti, T.L.: Combined route capacity and route length models for unit demand vehicle routing problems. Discr. Optim. **5**, 350–372 (2008)

[13] Gouveia, L: A result on projection for the vehicle routing problem. Eur. J. Oper. Res. **85**, 610–624 (1995)

[14] Guignard, M.: Lagrangean relaxation. Trabajos de Operativa (TOP) **11**, 151–228 (2003)

[15] Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**, 169–197 (1981)

[16] Kellerer, H., Pferschy, U., Pisinger, D.: The multiple-choice knapsack problem. Chapter 11 in: *Knapsack Problems.* Heidelberg: Springer (2004)

[17] Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. Eur. J. Oper. Res. **59**, 345–358 (1992)

[18] Laporte, G.: Fifty years of vehicle routing. Transp. Sci. **43**, 408–416 (2009)

[19] Laporte, G., Nobert, Y.: A branch and bound algorithm for the capacitated vehicle routing problem. OR Spektrum **5**, 77–85 (1983)

[20] Leggieri, V., Haouari, M.: Lifted polynomial size formulations for the homogeneous and heterogeneous vehicle routing problems. Eur. J. Oper. Res. **263**, 755–767 (2017)

[21] Lemaréchal, C.: The omnipresence of Lagrange. Ann. Oper. Res. **153**, 9–27 (2007)

[22] Letchford, A.N., Eglese, R.W., Lysgaard, J.: Multistars, partial multistars and the capacitated vehicle routing problem. Math. Program. **94**, 21–40 (2002)

[23] Letchford, A.N., Salazar-González, J.J.: Projection results for vehicle routing. Math. Program. **105**, 251–274 (2006)

[24] Letchford, A.N., Salazar-González, J.J.: Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. Eur. J. Oper. Res. **244**, 730–738 (2015)

[25] Lysgaard, J., Letchford, A.N., Eglese, R.W.: A new branch-and-cut algorithm for the capacitated vehicle routing problem. Math. Program. **100**, 423–445 (2004)

[26] Martello, S., Toth, P.: A new algorithm for the 0–1 knapsack problem. Mgmt. Sci. **34**, 633–644 (1988)

[27] Martinelli, R., Pecin, D., Poggi, M.: Efficient elementary and restricted non-elementary route pricing. Eur. J. Oper. Res. **239**, 102–111 (2014)

[28] Naddef, D., Rinaldi, G.: Branch-and-cut algorithms for the capacitated VRP. In Toth, P., Vigo, D. (eds.): *The Vehicle Routing Problem.* Philadelphia, PA: SIAM (2002)

[29] Pecin, D., Pessoa, A., Poggi, M., Uchoa, E.: Improved branch-cut-and-price for capacitated vehicle routing. Math. Program. Comput. **9**, 61–100 (2017).

[30] Poggi, M., Uchoa, E.: Integer program reformulation for robust branch-and-cut-and-price algorithms. In Proc. of Mathematical Programming in Rio: A Conference in Honour of Nelson Maculan, pp. 56–61 (2003)

[31] Poggi, M., Uchoa, E.: New exact algorithms for the capacitated vehicle routing problem. In [33], Chapter 3, pp. 59–86.

[32] Semet, F., Toth, P., Vigo, D.: Classical exact algorithms for the capacitated vehicle routing problem. In [33], Chapter 2, pp. 37–58.

[33] Toth, P., Vigo, D. (eds.): *Vehicle Routing: Problems, Methods and Applications.* Philadelphia, PA: MOS-SIAM Series on Optimization (2014)

[34] Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A.: New benchmark instances for the capacitated vehicle routing problem. Eur. J. Oper. Res. **257**, 845–858 (2017)