A disjunctive convex programming approach to the pollution routing problem

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

Abstract

The pollution routing problem (PRP) aims to determine a set of routes and speed over each leg of the routes simultaneously to minimize the total operational and environmental costs. A common approach to solve the PRP exactly is through speed discretization, i.e., assuming that speed over each arc is chosen from a prescribed set of values. In this paper, we keep speed as a continuous decision variable within an interval and propose new formulations for the PRP. In particular, we build two mixed-integer convex optimization models for the PRP, by employing tools from disjunctive convex programming. These are the first arc-based formulations for the PRP with continuous speed. We also derive several families of valid inequalities to further strengthen both models. We test the proposed formulations on benchmark instances, with some instances solved to optimality for the first time. The computational results also show the solutions from speed discretization can always give the same optimal routes for the benchmark instances.

1 Introduction

Transportation accounts for a significant portion of Greenhouse Gas (GHG) emissions, e.g., 27% among all sources in the United States in 2013 (Environmental Protection Agency 2015). Approaches to reduce GHG emissions in transportation include switching to cleaner fuels, improving fuel efficiency, reducing travel demand, etc. Another important approach is to improve operational practices. In classical operational models such as the vehicle routing problem (VRP), a common assumption is that the travel cost and travel time between two vertices is given as input data. The fuel consumption and GHG emissions, however, heavily depend on the routing and scheduling decisions. Many new transportation models have been proposed recently to improve fuel efficiency and reduce the negative environmental impact; see for instance recent surveys on green freight transportation (Demir et al. 2014b, Eglese and Bektaş 2014, Lin et al. 2014).

Speed control plays a significant role on fuel consumption and GHG emissions. For instance, the fuel consumption as well as the GHG emissions per kilometers of a heavy-duty truck at a high speed can be almost twice as much as that at a lower speed (Bektaş and Laporte 2011), and a computational study in Norstad et al. (2011) showed that optimizing speed on a ship route can reduce fuel consumption by as much as 14%. The integration of vehicle speed as a decision variable into traditional transportation models provides opportunities for savings in fuel consumption and reduction in GHG emissions. On the other hand, the dependence of fuel consumption and GHG

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

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

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

emissions on speed is usually nonlinear in practice, imposing great challenges on solving the resulted model exactly. In particular, the objective function and any constraint that involves travel duration in the classical routing models will now be nonlinear in speed. One common exact approach to handle this issue is by speed discretization, i.e., assuming the vehicle speed takes a discrete set of prescribed values within the limits. Then the nonlinear cost for every prescribed speed value can be computed and stored in advance, and the resulting model becomes a mixed-integer linear programming problem. One issue with speed discretization is that the true optimal speed may be excluded in advance from the prescribed set of speed values, if the chosen time step is not small enough. Another potential issue is that speed discretization introduces a large number of additional variables, which renders the model computationally challenge to solve for large-size instances. In this paper, we take a new approach by maintaining speed as a continuous decision variable. We introduce a framework to directly incorporate the nonlinear relationship between cost and speed into the routing problem, by employing tools from disjunctive convex programming. In particular, we focus on the pollution routing problem (PRP) proposed by Bektaş and Laporte (2011). The PRP aims to find a set of vehicle routes and vehicle speeds over the routes to minimize the operational and environmental costs, while respecting constraints on time and vehicle capacities. The PRP is NP-hard, since it generalizes the VRP. Even a medium-size PRP instance is extremely challenging to solve exactly: as shown in Bektaş and Laporte (2011), the mixed-integer linear programming formulation through speed discretization for a 15-city one-vehicle instance cannot be solved by CPLEX 12.1 to optimality within three hours. Our paper makes the following contribution to the literature on models that jointly optimize routing and speed decisions:

- We introduce the first arc-based formulations for the PRP with continuous speed. Our formulations directly model the nonlinear relationship between fuel consumption and vehicle speed, without speed discretization. These formulations provide a benchmark for the maximum cost saving and emission reduction that can be achieved by speed optimization.

- We study the theoretical strengths of the two formulations we propose. In particular, we compare the lower bounds provided by the continuous relaxations of the formulations. We also derive several families of valid inequalities for the PRP to strengthen these formulations.

- We test our formulations on benchmark instances (Bektaş and Laporte 2011, Kramer et al. 2015b). Instances with up to 25 cities can be solved to optimality, with some instances solved to optimality for the first time.

- We analyze the impact of labor cost and departure time at the depot on the optimal routes. We also compare the optimal solutions from our formulations with the optimal solutions through speed discretization. It shows that for the test instances the solutions from speed discretization always give the same optimal routes (although not the same optimal speeds). The comparison provides some empirical justification on speed discretization for the PRP.

The rest of the paper is organized as follows. Section 2 reviews results related to the PRP in the literature. Section 3 gives a complete description of the PRP. Section 4 gives a brief overview of disjunctive convex programming techniques that we employ in formulating the PRP as an MICP. Section 5 presents two MICP formulations for the PRP, and compares the lower bounds given by their continuous relaxations. Several families of valid inequalities are introduced in Section 6. Extensive computational results and analysis are given in Section 7. We conclude and point out several future research directions in Section 8.

2 Literature Review

Speed control has a great economic and environmental impact in transportation, in particular maritime transportation. The daily fuel consumption of a ship increases dramatically with the sailing speed (Notteboom and Vernimmen 2009, Ronen 2011). It is shown in Hvattum et al. (2013), Laporte (2016) that with a fuel price of 450 USD/tonne, a one percent worldwide reduction in fuel consumption would yield an estimated cost reduction of more than 1.2 billion USD, and a reduction of CO_2 emissions of 10.5 million tonnes. The impact of speed control on reducing GHG emissions and fuel consumption in maritime shipping have been studied extensively; See Cariou (2011), Kontovas and Psaraftis (2011), Kim et al. (2012), Lee and Song (2016) and the references therein. One problem closely related to the PRP is the tramp speed optimization problem (SOP) studied by Fagerholt et al. (2010) and Norstad et al. (2011), in which a ship aims to find the optimal sailing speed over each leg of a fixed route to minimize the total fuel consumption. An efficient iterative method, called the speed optimization algorithm (SOA), was developed to solve SOP exactly when the fuel consumption function is a convex function of the ship speed (Norstad et al. 2011, Hvattum et al. 2013). The SOA was modified by Demir et al. (2012) to solve the SOP with additional labor cost, and by Kramer et al. (2015a) to handle the SOP with departure time at the origin as an additional decision variable.

The PRP integrates speed and routing decisions to minimize the total operational and environmental cost. The PRP generalizes several variants of the VRP in the following way: with fixed speed over each arc, the PRP reduces to the energy minimizing vehicle routing problem (EMVRP) (Kara et al. 2007, 2008) with time-windows constraints, and is equivalent to the classical VRP with time windows when vehicle curb weight is much larger than cargo weight (Fukasawa et al. 2015); the PRP also generalizes the fuel consumption VRP proposed in Xiao et al. (2012). The PRP is difficult to solve exactly for large-size instances. In Bektaş and Laporte (2011), the speed over each arc is allowed to take ten prescribed values and the PRP is modeled as a mixed-integer linear program. The resulted model is solved by CPLEX for small-size instances. Heuristic and matheuristic methods were derived to find good solutions for large-size PRP instances. Demir et al. (2012) combined an adaptive large neighborhood search method to find promising routes and the SOA for optimizing speeds over given routes. Kramer et al. (2015b) integrated the SOA and the set partitioning formulation of the VRP into a multi-start iterated local search framework, balancing the search between heuristic and exact methods. Two recent papers developed exact algorithms to solve the PRP with continuous speed. In Dabia et al. (2014), the authors proposed a branch-and-price algorithm for a variant of the PRP, in which the speed is assumed to be the same over each route and the departure time at the depot is allowed to vary. They tested the algorithms on modified Solomon instances and were able to solve most 25-city instances and several 50-city and 100-city instances. Fukasawa et al. (2016) studied a joint routing and speed optimization problem, by allowing the fuel consumption rate to be a general convex function of the speed. They developed a branch-cut-and-price algorithm to solve the problem. Several extensions of the PRP have also been studied in the literature. Demir et al. (2014a) studied a bi-objective PRP with the goal of minimizing both the fuel cost and the total travel duration. Franceschetti et al. (2013) studied the time-dependent PRP using a two-stage planning setting proposed in Jabali et al. (2012) for a time-dependent VRP, in which the whole planning horizon is partitioned into free-flow phase and congestion phase. They proposed a mathematical formulation for the problem by assuming the free-flow speed takes values from a discrete set. More recently, Ehmke et al. (2016) studied the problem of minimizing CO_2 emissions in routing vehicles in urban areas, taking the traffic into account in their model. Koç et al. (2014) introduced the fleet size and mix pollution-routing problem by considering heterogeneous vehicle fleet in the PRP. They proposed a hybrid evolutionary metaheuristic to solve the problem and demonstrated the benefit of using heterogeneous fleet over a homogeneous one. Koç et al. (2016) further studied a model that jointly optimizes the decisions of depot location, fleet composition, and routing to minimize the emission in city logistics. For a comprehensive overview on recent development on routing problems in green freight transportation, please refer to recent surveys (Demir et al. 2014b, Eglese and Bektaş 2014, Lin et al. 2014).

Speed optimization has also been integrated with planning decisions other than routing, mostly in the context of maritime transportation. Alvarez et al. (2010) and Du et al. (2011) studied the berth allocation policy by taking the bunker consumption and vessel speed optimization into account. Wang and Meng (2012b) optimized the speed of containerships for a network consisting of many liner services by solving a mixed-integer nonlinear programming model. Wang and Meng (2012a) studied the liner ship route schedule design problem with sea contingency and uncertain port time to minimize the ship cost and bunker cost. A mixed-integer convex stochastic programming model was developed and solved by a cutting-plane based algorithm. Meng et al. (2015) studied a joint routing and bunkering problem to maximize the total profit for a tramp shipping company, and they developed a branch-and-price algorithm to solve the model. Wang (2016) studied the containership sailing speed optimization problem, in which the goal is to determine the number of ships deployed on each route as well as the sailing speed over each leg of the route to minimize the total ship chartering and fuel costs. Christiansen et al. (2004), Meng et al. (2013), and Lee and Song (2016) have provided a comprehensive survey on ship routing and scheduling problems. Speed optimization has also been considered in airline transportation. Aktürk et al. (2014) studied an airline rescheduling problem from a disruption, by balancing the trade-off between flight delay and fuel consumption.

We finally survey some recent developments for mixed-integer convex programs (MICPs). An MICP is a convex optimization problem where some variables are required to be integral. It can be solved by MICP solvers such as Bonmin (Bonami et al. 2008), DICOPT (Grossmann et al. 2002), and FilMINT (Abhishek et al. 2010). Algorithms for MICPs can be classified into three categories: (1) the nonlinear branch-and-bound (B&B) algorithm, where a relaxed convex optimization problem is solved at each node of the B&B tree (Dakin 1965, Gupta and Ravindran 1985); (2) the outer approximation algorithm and generalized Benders decomposition algorithm, where a mixed integer linear programming relaxation is solved at each iteration (Bonami et al. 2008, Fletcher and Leyffer 1994, Geoffrion 1972); (3) the hybrid algorithm, where a single B&B search tree is maintained and linear/nonlinear programs are solved at each node (Quesada and Grossmann 1992, Abhishek et al. 2010). For an overview of the latest algorithmic developments and available solvers for the MICP, we refer the readers to a recent survey (Belotti et al. 2013) and the book (Lee and Leyffer 2012). We should also note that a class of special-structured MICPs, called the mixed integer second-order cone program (MISOCP), can be handled directly by the state-of-the-art commercial optimization software such as CPLEX (IBM ILOG 2015) and Gurobi (Gurobi Optimization 2015). Recent applications of the MISOCP model include portfolio optimization (Vielma et al. 2008), machine scheduling (Aktürk et al. 2010), the stochastic joint location-inventory problem (Atamtürk et al. 2012), and aircraft rescheduling (Aktürk et al. 2014). We will review more solution techniques for the MISOCP in Section 4 when introducing the techniques in disjunctive convex programming.

3 Problem description

Throughout the paper, we work on the delivery version of the PRP with asymmetric distance matrix and homogeneous vehicles. The problem can be described as follows. Let $G = (V, A)$ be a complete directed graph with $V = \{0, 1, 2, \dots, n\}$, where vertex 0 denotes the depot and $V_0 = \{1, 2, \dots, n\}$

denotes the set of customers or cities (used interchangeably for the rest of the paper). For every $(i, j) \in A$, let d_{ij} be the distance from vertex i to vertex j, and l_{ij} and u_{ij} be the lower and upper speed limits over arc (i, j). Every customer $i \in V_0$ is associated with a demand q_i, a service time τ_i and a time window $[a_i, b_i]$, where a_i and b_i represent the earliest and latest service start time for that customer, respectively (we let b_0 denote the latest time allowed to return to the depot). There are K identical vehicles at the depot, each of which has a curb weight w and a capacity Q. The goal is to select a set of routes and speeds over arcs of the routes to minimize the total operational and environmental costs, respecting the following constraints: (1) each route starts and ends at the depot; (2) the total customer demand on that route does not exceed the vehicle capacity Q; (3) each customer is visited exactly once, and the service must start within its time window (a vehicle will wait if it arrives early); (4) the vehicle speed over arc (i, j) does not violate the speed limits over that arc, i.e., the speed falls within $[l_{ij}, u_{ij}]$; (5) each vehicle needs to return to the depot no later than b_0. The total cost of each route consists of the following two parts.

- The summation of fuel and GHG emission costs over each arc of the route. The fuel cost over an arc is calculated as the product of the fuel unit price, the fuel consumption rate (liter per second) and the travel duration over that arc. The fuel consumption rate is modeled as a function of the vehicle speed and the vehicle weight along with other factors, which will be explained in more detail at the end of this section. Since the GHG emission rate is typically assumed to be proportional to the fuel consumption rate (Bektaş and Laporte 2011), the GHG emission cost over each arc will also be proportional to the fuel cost over that arc.

- The labor cost (driver wages). Each driver is paid by a standard unit salary of p. If a vehicle leaves the depot at time t_1 and returns to the depot at time t_2, then the driver's wage over that route will be $p(t_2 - t_1)$.

In the original PRP, the driver always leaves the depot at time 0 and is paid from time 0 to the time that she returns to the depot, but it is possible to reduce the labor cost by allowing later departure at the depot. To offer more operational flexibility, we also include the departure time at the depot for each vehicle as a decision variable. This more general setting was also considered in Franceschetti et al. (2013), Dabia et al. (2014), Kramer et al. (2015a).

The fuel consumption and GHG emission model There have been many fuel consumption and emission models for trucks in the literature; see the survey by Demir et al. (2014b). The PRP uses the comprehensive modal emissions model developed in Barth et al. (2004), which takes into account many factors including the vehicle type, engine, emission technology, and level of deterioration for calculating the fuel consumption and emissions. The fuel and emission function employed in the PRP is for one specific type of heavy-duty diesel vehicles. In particular, the model assumes that the GHG emission rate is proportional to the fuel consumption rate; the fuel consumption rate over arc (i, j) is calculated as

$$\beta_1 + \beta_2(w + f_{ij})v_{ij} + \beta_3 v_{ij}^3,$$

where v_{ij} is the vehicle speed over arc (i, j), w is the curb weight, f_{ij} is the vehicle's current payload, and β_1, β_2 and β_3 are positive constants calculated from vehicle and environment specific parameters (see Dabia et al. 2014, Demir et al. 2012, Kramer et al. 2015b for a complete list of parameters with their descriptions and typical values). Let c_{fe} be the total unit cost of fuel and GHG emissions. Then the total fuel and emission costs over arc (i, j), $F_{ij}(v_{ij})$, are calculated as

$c_{fe} \cdot (\beta_1 + \beta_2(w + f_{ij})v_{ij} + \beta_3 v_{ij}^3) \cdot d_{ij}/v_{ij}$ when $v_{ij} > 0$. Thus

$$F_{ij}(v_{ij}) = \begin{cases} \alpha_1 \frac{d_{ij}}{v_{ij}} + \alpha_2 d_{ij} + \alpha_3 d_{ij} f_{ij} + \alpha_4 d_{ij} v_{ij}^2, & v_{ij} > 0, \\ 0, & v_{ij} = 0, \end{cases} \tag{3.1}$$

where $\alpha_1 = c_{fe}\beta_1$, $\alpha_2 = c_{fe}\beta_2 w$, $\alpha_3 = c_{fe}\beta_2$ and $\alpha_4 = c_{fe}\beta_3$. Equivalently, given the travel duration $t_{ij} = d_{ij}/v_{ij}$, the total fuel and emission costs over arc (i, j) are

$$H_{ij}(t_{ij}) = \begin{cases} \alpha_1 t_{ij} + \alpha_2 d_{ij} + \alpha_3 d_{ij} f_{ij} + \alpha_4 \frac{d_{ij}^3}{t_{ij}^2}, & t_{ij} > 0, \\ 0, & t_{ij} = 0. \end{cases} \tag{3.2}$$

Both functions F_{ij} and H_{ij} are discontinuous. This discontinuity imposes great challenges on building a tractable exact model for the PRP. We will employ disjunctive convex programming techniques introduced in the next section to overcome this difficulty.

4 Disjunctive convex programming

We introduce some techniques from disjunctive convex programming that will be employed in formulating the PRP. A *disjunctive convex programming problem* is of the form

$$\min \left\{ c^T z \mid z \in \bigcup_{i \in I} K^i \right\}, \tag{4.1}$$

where $K^i = \{z \in \Re^n : G_{ij}(z) \leq 0, j \in J_i\}$ for each $i \in I$, and $G_{ij}(z)$ is a closed convex function for each $j \in J_i$ and $i \in I$. Many design and operation problems arising in engineering can be formulated in the form of (4.1). For example, the PRP contains the following type of constraints: either arc (i, j) is used (so there is a constraint linking the service starting time at customer j and that at customer i), or arc (i, j) is not used (so there is no such constraint). The linear objective function is without loss of generality, since any convex objective function can be replaced by an additional auxiliary variable and modeled as a convex constraint. When G_{ij}'s are affine functions, (4.1) is the well-known disjunctive (linear) programming framework, introduced by Balas (Balas 1979, Balas et al. 1993) and studied extensively in the context of mixed integer linear programming. The case for general convex G_{ij}'s is much more complex.

Despite being a powerful modeling framework, disjunctive convex programming problems cannot be solved directly by state-of-the-art optimization software. It needs to be reformulated into an equivalent form, usually an MINLP, that can be handled by optimization solvers. Two common reformulation approaches are employed: *the big-M reformulation* and *the perspective reformulation.*

The big-M reformulation introduces an auxiliary binary variable x_i for each $i \in I$ to indicate which set K^i is selected. Then (4.1) can be reformulated as follows.

$$\min\ c^T z \tag{4.2a}$$

$$\text{s.t. } G_{ij}(z) \leq M_{ij}(1 - x_i), j \in J_i, i \in I, \tag{4.2b}$$

$$\sum_{i \in I} x_i = 1, \tag{4.2c}$$

$$x_i \in \{0, 1\}, i \in I, \tag{4.2d}$$

where M_{ij} is a large constant such that the constraint $G_{ij}(z) \leq M_{ij}$ is satisfied by any $z \in K^l$ for $l \neq i$, $j \in J_i$ and $i \in I$. For example, we can set $M_{ij} = \max_{l \neq i} \sup\{G_{ij}(z) \mid z \in K^l\}$, if such an

M_{ij} is finite. The big-M reformulation is easy to apply, but it has several drawbacks: such a finite constant M_{ij} may not always exist; even if it does, the introduction of large constants M_{ij}'s into the formulation may lead to weak continuous relaxations as well as numerical instability during computation.

The perspective reformulation, on the other hand, is built upon the fact that (4.1) is equivalent to

$$\min \left\{ c^T z \mid z \in \text{conv}(\bigcup_{i\in I} K^i) \right\},$$

where conv(S) denotes the convex hull of set S. A fundamental result in Ceria and Soares (1999) showed that $\text{conv}(\bigcup_{i\in I} K^i)$ can be represented as the projection of a convex set in a higher dimensional space, using the perspectives of G_{ij}'s. To make the presentation of the result in Ceria and Soares (1999) concise, we assume that K^i is bounded for each $i \in I$, which is the case for the PRP. Let $\overline{S}$ denote the closure of set S and $\text{proj}_z(S)$ denote the projection of set S onto the subspace of z-variables. The perspective of a function G_{ij} is defined as follows.

Definition 1. *(Hiriart-Urruty and Lemaréchal 1996) Given a function $G(z) : \Re^n \to \Re \cup \{+\infty\}$, the perspective of G is defined as a function $\tilde{G} : \Re^n \times \Re \to \Re \cup \{+\infty\}$ given by*

$$\tilde{G}(z,\lambda) = \begin{cases} \lambda G(\frac{z}{\lambda}), & \lambda > 0, \\ +\infty, & \lambda \le 0. \end{cases}$$

If $G(z)$ is convex in $\Re^n$, then its perspective $\tilde{G}(z,\lambda)$ is also convex in $\Re^{n+1}$.

The following theorem is from Ceria and Soares (1999).

Theorem 1. *(Ceria and Soares 1999) Suppose the set $\mathcal{K}$ is in the space of $(z, z^1, \ldots, z^{|I|}, \lambda_1, \ldots, \lambda_{|I|})$ and defined by constraints*

$$\tilde{G}_{ij}(z^i, \lambda_i) \le 0, j \in J_i, i \in I, \tag{4.3a}$$

$$z = \sum_{i\in I} z^i, \tag{4.3b}$$

$$\sum_{i\in I} \lambda_i = 1, \tag{4.3c}$$

$$\lambda_i > 0, i \in I, \tag{4.3d}$$

then conv$(\bigcup_{i\in I} K^i) = \text{proj}_z(\overline{\mathcal{K}})$.

By Theorem 1, problem (4.1) can be reformulated as

$$\begin{aligned} &\min\ c^T z \\ &\text{s.t. } z \in \text{proj}_z(\overline{\mathcal{K}}). \end{aligned}$$

Formulation (4) is called the perspective reformulation of (4.1). It is the tightest convex relaxation of (4.1), and therefore dominates the continuous relaxation of the big-M formulation (4.2). The difficulty of dealing with (4) is that $\text{proj}_z(\overline{\mathcal{K}})$ does not have a simple algebraic representation in general. For some special structured K^i's, however, the set $\text{proj}_z(\overline{\mathcal{K}})$ in (PF2) has a concise closed-form representation, which we illustrate below.

The two-term disjunction We now only consider the special case where $|I| = 2$ and one term of the disjunction is a single point. This type of constraints are sometimes called the indicator constraints (Bonami et al. 2015), since it can be seen as using a binary variable to turn on or off some constraints. The following result is a special case of Theorem 1 and a modified version of Corollary 1 in Günlük and Linderoth (2010).

Observation 1. *Let $x \in \{0,1\}, z \in \Re^n$, $K^1 = \{(x,z) \in \Re^{n+1} \mid x = 0, z_k = 0, k = 1, \dots, n\}$ and $K^2 = \{(x,z) \in \Re^{n+1} \mid x = 1, G_i(z) \leq 0, i \in I, l_k \leq z_k \leq u_k, k = 1, \dots, n\}$. Then conv$(K^1 \cup K^2) = \overline{K}$, where*

$$K = \{(x,z) \mid \tilde{G}_i(z,x) \leq 0, 0 < x \leq 1, l_k x \leq z_k \leq u_k x, k = 1, \dots, n\}.$$

By applying Observation 1 to specific forms of $G_i(z)$'s (which will appear later in the formulation of the PRP), we obtain the following observation.

Observation 2. *Suppose that $K^1 = \{(x, z_1, z_2, z_3) \in \Re^4 \mid x = z_1 = z_2 = z_3 = 0\}$ and $K^2 = \{(x, z_1, z_2, z_3) \in \Re^4_+ \mid x = 1, \frac{d}{z_1} - z_2 \leq 0, z_1^2 - z_3 \leq 0, l_k \leq z_k \leq u_k, k = 1,2,3\}$. Then*

$$\text{conv}(K^1 \cup K^2) = \{(x, z_1, z_2, z_3) \in \Re^4_+ \mid z_1 z_2 \geq d x^2, z_3 x \geq z_1^2, 0 \leq x \leq 1, l_k x \leq z_k \leq u_k x, k = 1,2,3\}. \quad (4.4)$$

Besides being convex, the set in Observation 2 is *second-order-cone representable.*

Definition 2. *A second-order cone constraint for z is of the form $Az - b \in \mathcal{L}^m$, where $A \in \Re^{m \times n}$, $b \in \Re^m$, and $\mathcal{L}^m$ is the second-order cone in $\Re^m$, i.e.,*

$$\mathcal{L}^m = \left\{ y \in \Re^m \mid y_m \geq \sqrt{\sum_{j=1}^{m-1} y_j^2} \right\}.$$

Definition 3. *A set $S \subseteq \Re^n$ is second-order-cone representable (SOCP-representable), if there exist some integers r and N, $A_j \in \Re^{m_j \times (n+r)}$ and $b_j \in \Re^{m_j}$ for $j = 1, \dots, N$ such that*

$$S = \text{proj}_z \left\{ (z,u) \in \Re^{n+r} \mid A_j \begin{pmatrix} z \\ u \end{pmatrix} - b_j \in \mathcal{L}^{m_j}, j = 1, \dots, N \right\}.$$

It is clear that set conv$(K^1 \cup K^2)$ in Observation 2 is SOCP-representable. In general, conv$(K^1 \cup K^2)$ is SOCP-representable if K^1 is a single point and K^2 is SOCP-representable Günlük and Linderoth (2010). We should note that in the PRP, the disjunctive constraint set always contains two terms with one term being a single point.

5 The arc-based formulations for the PRP with continuous speed

We first introduce a nonconvex formulation for the PRP, and then transform it into a disjunctive convex programming problem, and finally use the techniques introduced in Section 4 to reformulate the problem into MICPs.

5.1 A naive MINLP formulation

We first define variables that will be used in all formulations below:

- the binary variable x_{ij} which indicates whether arc (i,j) is used in any route, for any $(i,j) \in A$;

- the continuous variable f_{ij} which indicates the amount of flow (payload) over arc (i,j), for any $(i,j) \in A$;
- the continuous variable y_i which represents the service start time at customer i, for any $i \in V_0$,
- the continuous variable s_i which represents the return time to the depot on a route with customer i being the last visited customer (s_i equals 0 if customer i is not the last visited customer on a route), for any $i \in V_0$;
- the continuous variable r_i which represents departure time from the depot on a route with customer i being the first visited customer (r_i equals 0 if customer i is not the first visited customer on a route), for any $i \in V_0$.

The variables s_i and r_i are introduced to calculate the labor cost over each route. We next introduce the first formulation for the PRP, which we call *the naive formulation.* The naive formulation is easy to interpret, yet nonconvex. It includes another decision variable t_{ij}, the travel duration over arc (i,j), for any $(i,j) \in A$. The formulation is described as follows.

$$\min \sum_{(i,j)\in A} H_{ij}(t_{ij}) + p(\sum_{i\in V_0} s_i - \sum_{i\in V_0} r_i) \tag{5.1a}$$

$$\text{s.t.} \sum_{j\in\delta^+(i)} x_{ij} = 1, \ \forall i \in V_0, \tag{5.1b}$$

$$\sum_{j\in\delta^-(i)} x_{ji} = 1, \ \forall i \in V_0, \tag{5.1c}$$

$$\sum_{j\in\delta^+(0)} x_{0j} = K, \tag{5.1d}$$

$$\sum_{j\in\delta^-(i)} f_{ji} - \sum_{j\in\delta^+(i)} f_{ij} = q_i, \ \forall i \in V_0, \tag{5.1e}$$

$$q_j x_{ij} \le f_{ij} \le (Q - q_i)x_{ij}, \ \forall (i,j) \in A, \tag{5.1f}$$

$$x_{ij} \in \{0,1\}, f_{ij} \ge 0, \ \forall (i,j) \in A, \tag{5.1g}$$

$$y_j - y_i - \tau_i - t_{ij} \ge -M_{ij}(1 - x_{ij}), \ \forall i \in V_0, j \in V_0, \tag{5.1h}$$

$$a_i \le y_i \le b_i, \ \forall i \in V_0, \tag{5.1i}$$

$$s_i - y_i - \tau_i - t_{i0} \ge -M_{i0}(1 - x_{i0}), \ \forall i \in V_0, \tag{5.1j}$$

$$0 \le s_i \le b_0, \ \forall i \in V_0, \tag{5.1k}$$

$$y_i - r_i - t_{0i} \ge -M_{0i}(1 - x_{0i}), \ \forall i \in V_0, \tag{5.1l}$$

$$0 \le r_i \le U_i x_{0i}, \ \forall i \in V_0, \tag{5.1m}$$

$$\frac{d_{ij}}{u_{ij}} x_{ij} \le t_{ij} \le \frac{d_{ij}}{l_{ij}} x_{ij}, \ \forall (i,j) \in A, \tag{5.1n}$$

where $\delta^+(i)$ ($\delta^-(i)$) denotes the set of arcs leaving (entering) vertex i, and M_{ij}'s and U_i's are some large constants which we will discuss at the end of this paragraph. Constraints (5.1b) to (5.1g) ensure that each customer is visited once and the solution given by x variables is a collection of routes respecting the vehicle capacity constraints. Constraints (5.1h) and (5.1i) with x_{ij} being binary guarantee the feasibility of routes with respect to time considerations. Constraints (5.1j) and (5.1k) ensure the return time to the depot for each route to be calculated correctly, as constraints (5.1l) and (5.1m) do for the departure time from the depot. Constraints (5.1n) impose the speed limits.

The objective function consists of the total fuel and emission costs, $\sum_{(i,j)\in A} H_{ij}(t_{ij})$, and the total driver wages, $p(\sum_{i\in V_0} s_i - \sum_{i\in V_0} r_i)$, over all routes. To ensure that the constraints do not exclude any feasible solution, the values of M_{ij}'s and U_i's can be set as follows: $M_{ij} = \max\{b_i+\tau_i+\frac{d_{ij}}{l_{ij}}-a_j, 0\}$ for $i, j \in V_0$; $M_{i0} = b_i + \tau_i + \frac{d_{i0}}{l_{i0}}$, $M_{0i} = \max\{\frac{d_{0i}}{l_{0i}} - a_i, 0\}$, and $U_i = \max\{b_i - \frac{d_{0i}}{u_{0i}}, 0\}$ for $i \in V_0$.

Observe that constraints (5.1b) to (5.1g) are exactly the same set of constraints in the two-index one-commodity flow formulation for the classical capacitated vehicle routing problem (CVRP). Define

$$S_{\text{cvrp}} = \{(x, f) \in \Re^{2|A|} |\ (5.1\text{b}) - \ (5.1\text{g})\},$$

which denotes the feasible set of the CVRP. Any valid inequality for the feasible set of the CVRP is also valid for the PRP.

With t_{ij} fixed, constraints (5.1h) to (5.1i) also appear in the two-index one-commodity flow formulation for the VRP with time windows. Define

$$S_{\text{time}} = \{(x, t, y, s, r) \in \Re^{2|A|+3|V|-2} |\ (5.1\text{h}) - \ (5.1\text{m})\},$$

which includes all the time-related constraints in the PRP. Any valid inequality for the VRP with time windows can be modified for the PRP by treating t_{ij}'s as variables instead of parameters. We will present valid inequalities for the PRP in more detail in Section 6.

The main drawback of the naive formulation is that it cannot be handled by existing optimization software directly, since its objective function is discontinuous and nonconvex. In the rest of this section, we will transform the naive formulation into disjunctive convex programming problems, and then reformulate them into MICPs.

The first disjunctive constraints of the PRP are as follows. Notice that in the naive formulation, $f_{ij} = t_{ij} = 0$ when $x_{ij} = 0$. So, by introducing a new set of decision variables h_{ij} to represent $\frac{1}{t_{ij}^2}$, we can rewrite the fuel consumption function in the objective as $H_{ij}(t_{ij}) = \alpha_1 t_{ij} + \alpha_2 d_{ij} x_{ij} + \alpha_3 d_{ij} f_{ij} + \alpha_4 d_{ij}^3 h_{ij}$ with $(x_{ij}, t_{ij}, h_{ij}) \in K_{ij}^1 \bigcup K_{ij}^2$ and

$$K_{ij}^1 = \{(x_{ij}, t_{ij}, h_{ij}) \mid x_{ij} = t_{ij} = h_{ij} = 0\}, \tag{5.2a}$$

$$K_{ij}^2 = \left\{(x_{ij}, t_{ij}, h_{ij}) \mid x_{ij} = 1, l_{ij}^t \le t_{ij} \le u_{ij}^t, l_{ij}^h \le h_{ij} \le u_{ij}^h, h_{ij} \ge \frac{1}{t_{ij}^2}\right\}, \tag{5.2b}$$

where $l_{ij}^t = \frac{d_{ij}}{u_{ij}}$, $u_{ij}^t = \frac{d_{ij}}{l_{ij}}$, $l_{ij}^h = \frac{l_{ij}^2}{d_{ij}^2}$, and $u_{ij}^h = \frac{u_{ij}^2}{d_{ij}^2}$. Then the PRP can be reformulated as the following disjunctive convex programming problem.

$$\min \alpha_1 \sum_{(i,j)\in A} t_{ij} + \alpha_2 \sum_{(i,j)\in A} d_{ij} x_{ij} + \alpha_3 \sum_{(i,j)\in A} d_{ij} f_{ij} + \alpha_4 \sum_{(i,j)\in A} d_{ij}^3 h_{ij} + p(\sum_{i\in V_0} s_i - \sum_{i\in V_0} r_i) \tag{5.3a}$$

$$\text{s.t. } (x, f) \in S_{\text{cvrp}}, \tag{5.3b}$$

$$(x, t, y, s, r) \in S_{\text{time}}, \tag{5.3c}$$

$$(x_{ij}, t_{ij}, h_{ij}) \in K_{ij}^1 \bigcup K_{ij}^2, \forall (i, j) \in A. \tag{5.3d}$$

It can be easily verified that the constraint $h_{ij} \ge \frac{1}{t_{ij}^2}$ in K_{ij}^2 will always be tight for any optimal solution of the reformulation. Thus (5.3) is an equivalent formulation of the PRP.

5.2 The big-M formulation

Usually it is easier to reformulate a disjunctive convex program through the big-M approach. However, the big-M formulation (4.2) in Section 4 cannot be directly applied to model the set $K^1_{ij} \cup K^2_{ij}$ in (5.3d). The reason is as follows: let $G(t_{ij}, h_{ij}) = \frac{1}{t^2_{ij}} - h_{ij}$ be the function defining the convex constraint $h_{ij} \geq \frac{1}{t^2_{ij}}$ in K^2_{ij}. In the big-M formulation, the logical constraint, $x_{ij} = 1 \Rightarrow G(t_{ij}, h_{ij}) \leq 0$, will be reformulated as $G(t_{ij}, h_{ij}) \leq M(1 - x_{ij})$, if there exists a constant M such that $M \geq \sup\{G(t_{ij}, h_{ij}) \mid (x_{ij}, t_{ij}, h_{ij}) \in K^1_{ij}\}$. Such an M, however, does not exist, since $G(t_{ij}, h_{ij})$ is unbounded from above with t_{ij} and h_{ij} equal to 0. One way to circumvent this issue is to introduce an auxiliary variable θ_{ij} for each $(i, j) \in A$ and define two new convex sets

$$D^1_{ij} = \left\{(x_{ij}, t_{ij}, h_{ij}, \theta_{ij}) \mid x_{ij} = h_{ij} = \theta_{ij} = 0, l^t_{ij} \leq t_{ij} \leq u^t_{ij}\right\},$$

$$D^2_{ij} = \left\{(x_{ij}, t_{ij}, h_{ij}, \theta_{ij}) \mid x_{ij} = 1, \frac{1}{t^2_{ij}} - h_{ij} \leq 0, \theta_{ij} = t_{ij}, l^t_{ij} \leq t_{ij} \leq u^t_{ij}, l^h_{ij} \leq h_{ij} \leq u^h_{ij}\right\}.$$

It can be verified that (5.3) is equivalent to the problem below.

$$\min\ \alpha_1 \sum_{(i,j)\in A} \theta_{ij} + \alpha_2 \sum_{(i,j)\in A} d_{ij}x_{ij} + \alpha_3 \sum_{(i,j)\in A} d_{ij}f_{ij} + \alpha_4 \sum_{(i,j)\in A} d^3_{ij}h_{ij} + p(\sum_{i\in V_0} s_i - \sum_{i\in V_0} r_i) \quad (5.4a)$$

$$\text{s.t. } (x, f) \in S_{\text{cvrp}}, \quad (5.4b)$$

$$(x, t, y, s, r) \in S_{\text{time}}, \quad (5.4c)$$

$$(x_{ij}, t_{ij}, h_{ij}, \theta_{ij}) \in D^1_{ij} \cup D^2_{ij}, \forall (i, j) \in A. \quad (5.4d)$$

Then problem (5.4) can be reformulated into an equivalent MICP through a big-M approach, as shown in (4.2).

$$\min\ \alpha_1 \sum_{(i,j)\in A} \theta_{ij} + \alpha_2 \sum_{(i,j)\in A} d_{ij}x_{ij} + \alpha_3 \sum_{(i,j)\in A} d_{ij}f_{ij} + \alpha_4 \sum_{(i,j)\in A} d^3_{ij}h_{ij} + p(\sum_{i\in V_0} s_i - \sum_{i\in V_0} r_i)$$

$$\text{s.t. } (x, f) \in S_{\text{cvrp}}, \quad (5.5a)$$

$$(x, t, y, s, r) \in S_{\text{time}}, \quad (5.5b)$$

$$\frac{1}{t^2_{ij}} - h_{ij} \leq \frac{1}{(l^t_{ij})^2}(1 - x_{ij}),\ \forall (i, j) \in A, \quad (5.5c)$$

$$l^t_{ij}(1 - x_{ij}) \leq t_{ij} - \theta_{ij} \leq u^t_{ij}(1 - x_{ij}),\ \forall (i, j) \in A, \quad (5.5d)$$

$$l^t_{ij} \leq t_{ij} \leq u^t_{ij},\ \forall (i, j) \in A, \quad (5.5e)$$

$$l^h_{ij}x_{ij} \leq h_{ij} \leq u^h_{ij}x_{ij},\ \forall (i, j) \in A, \quad (5.5f)$$

$$l^t_{ij}x_{ij} \leq \theta_{ij} \leq u^t_{ij}x_{ij}, \forall (i, j) \in A. \quad (5.5g)$$

Formulation (5.5) can be handled directly by any MICP or MINLP solver. Another way to solve this formulation is through a two-stage generalized Benders' decomposition algorithm. The first-stage problem is a mixed integer linear program with variables x and f.

$$\min\ \alpha_2 \sum_{(i,j)\in A} d_{ij}x_{ij} + \alpha_3 \sum_{(i,j)\in A} d_{ij}f_{ij} + \mathcal{Q}(x) \quad (5.6a)$$

$$\text{s.t. } (x, f) \in S_{\text{cvrp}}. \quad (5.6b)$$

With fixed x and f, the second-stage problem is a convex optimization problem with variables t, y, s, r, h and θ.

$$\begin{aligned} \mathcal{Q}(x) = \ & \min \ \alpha_1 \sum_{(i,j)\in A} \theta_{ij} + \alpha_4 \sum_{(i,j)\in A} d_{ij}^3 h_{ij} + p \sum_{i\in V_0} (s_i - r_i) \\ & \text{s.t. } (5.1\text{h}) - (5.1\text{m}), \\ & \qquad (5.5\text{c}) - \ (5.5\text{g}). \end{aligned} \tag{5.7}$$

The first-stage master problem (5.6) without the term $\mathcal{Q}(x)$ in the objective is exactly the EMVRP studied in Fukasawa et al. (2015), Kara et al. (2007, 2008). The second-stage problem can be decomposed into K separate speed optimization problems, each of which allows varying departure time at the depot. This speed optimization problem has been studied in a recent paper (Kramer et al. 2015a). If we set $r_i = 0$ for each $i \in V_0$, then the speed optimization problem reduces to the one studied in Demir et al. (2012), Kramer et al. (2015b).

5.3 A mixed integer second-order cone programming model

In practice, the fuel consumption or emission rate is modeled as a function of the vehicle's instantaneous speed, instead of the travel duration over certain arc. To make our formulation amenable to other transportation models involving fuel consumption and GHG emissions, we now include vehicle speed v_{ij} over arc (i,j) explicitly as decision variables. The travel duration on arc (i,j) is a piecewise nonlinear function of v_{ij}: it equals to d_{ij}/v_{ij} if arc (i,j) is used in any route and 0 otherwise. We keep the travel duration t_{ij}'s in the decision variables to keep the objective function linear. By introducing a new set of decision variables π_{ij} to represent the quadratic term v_{ij}^2, the PRP can be formulated as follows.

$$\min \alpha_1 \sum_{(i,j)\in A} t_{ij} + \alpha_2 \sum_{(i,j)\in A} d_{ij}x_{ij} + \alpha_3 \sum_{(i,j)\in A} d_{ij}f_{ij} + \alpha_4 \sum_{(i,j)\in A} d_{ij}\pi_{ij} + p(\sum_{i\in V_0} s_i - \sum_{i\in V_0} r_i) \tag{5.8a}$$

$$\text{s.t. } (x, f) \in S_{\text{cvrp}}, \tag{5.8b}$$

$$(x, t, y, s, r) \in S_{\text{time}}, \tag{5.8c}$$

$$(x_{ij}, v_{ij}, t_{ij}, \pi_{ij}) \in L_{ij}^1 \cup L_{ij}^2, \ \forall (i,j) \in A, \tag{5.8d}$$

where

$$L_{ij}^1 = \{(x_{ij}, v_{ij}, t_{ij}, \pi_{ij}) \mid x_{ij} = v_{ij} = t_{ij} = \pi_{ij} = 0\},$$

$$L_{ij}^2 = \{(x_{ij}, v_{ij}, t_{ij}, \pi_{ij}) \mid x_{ij} = 1, t_{ij} \geq \frac{d_{ij}}{v_{ij}}, \pi_{ij} \geq v_{ij}^2, l_{ij}^v \leq v_{ij} \leq u_{ij}^v, l_{ij}^t \leq t_{ij} \leq u_{ij}^t, l_{ij}^\pi \leq \pi_{ij} \leq u_{ij}^\pi\},$$

where $l_{ij}^v = l_{ij}$, $u_{ij}^v = u_{ij}$, $l_{ij}^\pi = l_{ij}^2$ and $u_{ij}^\pi = u_{ij}^2$.

Proposition 1. *Formulation* (5.8) *is a valid formulation for the PRP.*

Proof. We only need to show that any optimal solution $(x, f, v, t, y, s, r, \pi)$ for formulation (5.8) satisfies that $t_{ij} = \frac{d_{ij}}{v_{ij}}$ and $\pi_{ij} = v_{ij}^2$ when $x_{ij} = 1$ for each $(i,j) \in A$. First, we must have $\pi_{ij} = v_{ij}^2$ to achieve optimality. Second, if $t_{ij} > \frac{d_{ij}}{v_{ij}}$ for some $(i,j) \in A$, then replacing t_{ij} with $\frac{d_{ij}}{v_{ij}}$ in the solution $(x, f, v, t, y, s, r, \pi)$ will give a feasible solution of (5.8) with a smaller objective, contradicting to the assumption that the original solution is optimal. □

To reformulate (5.8) as an MICP, we notice that the constraint $(x_{ij}, v_{ij}, t_{ij}, \pi_{ij}) \in (L^1_{ij} \cup L^2_{ij}) \cap (\{0,1\} \times \Re^3_+)$ in (5.8) is equivalent to that $(x_{ij}, v_{ij}, t_{ij}, \pi_{ij}) \in \text{conv}(L^1_{ij} \cup L^2_{ij}) \cap (\{0,1\} \times \Re^3_+)$. The description of $\text{conv}(L^1_{ij} \cup L^2_{ij})$ can be obtained through the perspective formulation in Proposition 2, which leads to the following MICP formulation for the PRP.

$$\min\ \alpha_1 \sum_{(i,j)\in A} t_{ij} + \alpha_2 \sum_{(i,j)\in A} d_{ij}x_{ij} + \alpha_3 \sum_{(i,j)\in A} d_{ij}f_{ij} + \alpha_4 \sum_{(i,j)\in A} d_{ij}\pi_{ij} + p(\sum_{i\in V_0} s_i - \sum_{i\in V_0} r_i) \quad (5.9a)$$

$$\text{s.t. } (x,f) \in S_{\text{cvrp}}, \quad (5.9b)$$

$$(x,t,y,s,r) \in S_{\text{time}}, \quad (5.9c)$$

$$v_{ij}t_{ij} \geq d_{ij}x_{ij}^2,\ \forall (i,j) \in A, \quad (5.9d)$$

$$\pi_{ij}x_{ij} \geq v_{ij}^2,\ \forall (i,j) \in A, \quad (5.9e)$$

$$l^v_{ij}x_{ij} \leq v_{ij} \leq u^v_{ij}x_{ij},\ \forall (i,j) \in A, \quad (5.9f)$$

$$l^t_{ij}x_{ij} \leq t_{ij} \leq u^t_{ij}x_{ij},\ \forall (i,j) \in A, \quad (5.9g)$$

$$l^\pi_{ij}x_{ij} \leq \pi_{ij} \leq u^\pi_{ij}x_{ij},\ \forall (i,j) \in A. \quad (5.9h)$$

Note that constraints (5.9d) can be reformulated as $(v_{ij} - t_{ij}, 2\sqrt{d_{ij}}x_{ij}, v_{ij} + t_{ij}) \in \mathcal{L}^3$ and constraints (5.9e) can be reformulated as $(\pi_{ij} - x_{ij}, 2v_{ij}, \pi_{ij} + x_{ij}) \in \mathcal{L}^3$ with $x_{ij}, v_{ij}, t_{ij}, \pi_{ij} \geq 0$. Therefore, (5.9) is an MISOCP.

The following result shows that the MISOCP formulation (5.9) is stronger than the big-M formulation (5.5).

Proposition 2. *Let z_{bigM} and z_{socp} be the optimal objective values of the continuous relaxations of the big-M formulation* (5.5) *and the MISOCP formulation* (5.9)*, respectively, we have:*

$$z_{socp} \geq z_{bigM}.$$

Proof. Given an optimal solution $(\hat{x}, \hat{f}, \hat{y}, \hat{s}, \hat{r}, \hat{v}, \hat{t}, \hat{\pi})$ of the continuous relaxation of the MISOCP formulation (5.9), we construct a solution $(\bar{x}, \bar{f}, \bar{y}, \bar{s}, \bar{r}, \bar{t}, \bar{h}, \bar{\theta})$ of the continuous relaxation of the big-M formulation (5.5) with the same objective value as follows: $\bar{x} = \hat{x}, \bar{f} = \hat{f}, \bar{y} = \hat{y}, \bar{s} = \hat{s}, \bar{r} = \hat{r}$, and:

- when $\hat{x}_{ij} = 0$, $\bar{h}_{ij} = \bar{\theta}_{ij} = 0$, $\bar{t}_{ij} = u^t_{ij}$;
- when $\hat{x}_{ij} > 0$, $\bar{h}_{ij} = \hat{\pi}_{ij}/d^2_{ij}$, $\bar{t}_{ij} = \hat{t}_{ij}/\hat{x}_{ij}$, $\bar{\theta}_{ij} = \hat{t}_{ij}$.

It is not hard then to check that all constraints in (5.5) are satisfied by $(\bar{x}, \bar{f}, \bar{y}, \bar{s}, \bar{r}, \bar{t}, \bar{h}, \bar{\theta})$, and $(\bar{x}, \bar{f}, \bar{y}, \bar{s}, \bar{r}, \bar{t}, \bar{h}, \bar{\theta})$ and $(\hat{x}, \hat{f}, \hat{y}, \hat{s}, \hat{r}, \hat{v}, \hat{t}, \hat{\pi})$ yield the same objective value in (5.5) and (5.9), respectively. □

6 Valid inequalities for the PRP

We present several families of valid inequalities for the PRP to strengthen the formulations proposed in Section 5. First of all, since the feasible solutions of the PRP satisfy the constraints in the two-index one-commodity flow formulation of the CVRP, any valid inequality for that formulation of the CVRP is also valid for the PRP, including all the cuts in the CVRPSEP package (Lysgaard 2003). Note that CVRPSEP package contains cuts for *undirected graphs*. To make it work, we first transform the fractional solution obtained from our formulation into a solution in an undirected graph and then perform the separation. Specifically, let $\overline{G} = (V, E)$ be the undirected version of graph G. We transform any fractional solution $(\hat{x}_{ij})_{(i,j)\in A}$ obtained from our formulation into a solution $(\bar{x}_{ij})_{(i,j)\in E}$ in $\overline{G}$, by setting $\bar{x}_{ij} = \hat{x}_{ij} + \hat{x}_{ji}$, for each $(i, j) \in E$. Suppose $\bar{x}$ violates a cut $\sum_{(i,j)\in E} \pi_{ij} x_{ij} \geq \pi_0$, we then add a cut $\sum_{(i,j)\in A} \pi_{ij}(x_{ij} + x_{ji}) \geq \pi_0$ into our formulation.

The new valid inequalities for the PRP are motivated by the lifted t-bound constraints and strengthened Miller-Tucker-Zemlin subtour elimination constraints proposed in Desrochers and Laporte (1991) for the traveling salesman problem with time windows (TSPTW). Define S_{prp} to be the set of feasible solutions of the MISOCP formulation of the PRP. We first introduce the base inequalities motivated by the lifted t-bound inequalities for TSPTW.

Proposition 3. *The following inequalities are valid for S_{prp}.*

$$y_i \geq a_i + \sum_{j\in\delta^-(i)} \max\{0, a_j + \tau_j + t_{ji} - a_i\} x_{ji}, \qquad \forall i \in V_0, \tag{6.1a}$$

$$y_i + \sum_{j\in\delta^+(i)} \max\{0, b_i + \tau_i + t_{ij} - b_j\} x_{ij} \leq b_i, \qquad \forall i \in V_0. \tag{6.1b}$$

Proof. We first prove the validity of constraint (6.1a). Given a feasible route, suppose that $x_{j^*i} = 1$ and $x_{ji} = 0$ for any $j \in \delta^-(i) \setminus \{j^*\}$. Since $y_i \geq a_i + (a_{j^*} + \tau_{j^*} + t_{j^*i} - a_i)$ and $y_i \geq a_i$, then $y_i \geq a_i + \max\{0, a_j + \tau_j + t_{j^*i} - a_i\} x_{j^*i}$. Since $\sum_{j\in\delta^-(i)} x_{ji} = 1$ for any feasible solution in S_{prp}, $y_i \geq a_i + \sum_{j\in\delta^-(i)} \max\{0, a_j + \tau_j + t_{ji} - a_i\} x_{ji}$. To prove the validity of constraint (6.1b), we assume that $x_{ij^*} = 1$ and $x_{ij} = 0$ for any $j \in \delta^+(i) \setminus \{j^*\}$ in a feasible route. Since $y_i \leq b_i$ and $y_i + (b_i + \tau_i + t_{ij^*} - b_{j^*}) \leq b_i$, then $y_i + \max\{0, b_i + \tau_i + t_{ij^*} - b_{j^*}\} x_{ij^*} \leq b_i$ and (6.1b) follows. □

Note that constraints (6.1a) and (6.1b) are nonlinear. We have two ways to linearize the constraints.

Proposition 4. *The following modified lifted t-bound inequalities are valid for S_{prp}.*

$$y_i \geq a_i + \sum_{j\in\delta^-(i)} \max\{0, a_j + \tau_j + l^t_{ij} - a_i\} x_{ji}, \qquad \forall i \in V_0, \tag{6.2a}$$

$$y_i + \sum_{j\in\delta^+(i)} \max\{0, b_i + \tau_i + l^t_{ij} - b_j\} x_{ij} \leq b_i, \qquad \forall i \in V_0. \tag{6.2b}$$

Proposition 5. *The following linearized lifted t-bound inequalities are valid for S_{prp}.*

$$y_i \geq a_i + \sum_{j\in\delta^-(i): a_j+\tau_j+u^t_{ij} > a_i} [(a_j + \tau_j - a_i) x_{ji} + t_{ji}], \qquad \forall i \in V_0, \tag{6.3a}$$

$$y_i + \sum_{j\in\delta^+(i): b_i+\tau_i+u^t_{ij} > b_j} [(b_i + \tau_i - b_j) x_{ij} + t_{ij}] \leq b_i, \qquad \forall i \in V_0. \tag{6.3b}$$

Proof. If $a_j + \tau_j + u^t_{ij} \leq a_i$, then $\max\{0, a_j + \tau_j + t_{ji} - a_i\} = 0$ for any optimal solution. Constraints (6.1a) can be first relaxed to

$$y_i \geq a_i + \sum_{j\in\delta^-(i): a_j+\tau_j+u^t_{ij} > a_i} (a_j + \tau_j + t_{ji} - a_i) x_{ji}.$$

Observe that $t_{ji} x_{ji} = t_{ji}$ for $x_{ji} \in \{0, 1\}$. Then we obtain constraints (6.3a). Constraints (6.3b) follow similarly. □

We now introduce another family of nonlinear base inequalities for the PRP, motivated by the strengthened Miller-Tucker-Zemlin subtour elimination constraints introduced in Desrochers and Laporte (1991).

Proposition 6. *For any $i, j \in V_0$, the following inequalities are valid for S_{prp}.*

$$y_i + \tau_i + t_{ij} - (1 - x_{ij})M_{ij} + (M_{ij} - \tau_i - \max\{\tau_j + t_{ji}, a_i - a_j\})x_{ji} \le y_j, \tag{6.4}$$

where M_{ij} is chosen the same way as in (5.1)*.*

Proof. We first prove the validity of the following constraints.

$$y_i + \tau_i + t_{ij} - (1 - x_{ij})M_{ij} + (M_{ij} - \tau_i - t_{ij} - \max\{\tau_j + t_{ji}, a_i - a_j\})x_{ji} \le y_j. \tag{6.5}$$

If $x_{ij} = 1$ and $x_{ji} = 0$, constraint (6.5) holds since $y_i + \tau_i + t_{ij} \le y_j$. If $x_{ij} = x_{ji} = 0$, then constraint (6.5) holds due to the choice of M_{ij}. If $x_{ij} = 0$ and $x_{ji} = 1$, then the vehicle goes from customer j to customer i. If the vehicle waits at customer i, then $y_i = a_i$ and $y_i - (a_i - a_j) \le y_j$; if the vehicle starts service at customer i right away, then $y_i = y_j + \tau_j + t_{ji}$, so $y_i - (\tau_j + t_{ji}) \le y_j$. Therefore, $y_i - \max\{\tau_j + t_{ji}, a_i - a_j\} \le y_j$. Constraints (6.5) can be strengthened to (6.4) since $t_{ij}x_{ji} = 0$ for any $i, j \in V_0$ in a feasible solution of the PRP. □

There are two ways to linearize (6.4).

Proposition 7. *We can relax* (6.4) *to the following valid inequalities.*

$$y_i + \tau_i + t_{ij} - (1 - x_{ij})M_{ij} + (M_{ij} - \tau_i - \max\{\tau_j + u^t_{ji}, a_i - a_j\})x_{ji} \le y_j, \tag{6.6}$$

for $i, j \in V_0$.

Proposition 8. *For $i, j \in V_0$,*

- *if $a_i - a_j - \tau_j \le l^t_{ji}$, then the following inequality is valid for S_{prp},*

$$y_i + \tau_i + t_{ij} - (1 - x_{ij})M_{ij} + (M_{ij} - \tau_i - \tau_j)x_{ji} - t_{ji} \le y_j; \tag{6.7}$$

- *if $a_i - a_j - \tau_j \ge u^t_{ji}$, then the following inequality is valid for S_{prp},*

$$y_i + \tau_i + t_{ij} - (1 - x_{ij})M_{ij} + (M_{ij} - \tau_i - a_i + a_j)x_{ji} \le y_j; \tag{6.8}$$

- *if $a_i - a_j - \tau_j \in (l^t_{ji}, u^t_{ji})$, then the following inequality is valid for S_{prp},*

$$y_i + \tau_i + t_{ij} - (1 - x_{ij})M_{ij} + (M_{ij} - \tau_i - \tau_j - a_i + a_j)x_{ji} - t_{ji} \le y_j. \tag{6.9}$$

Proof. Note that when $a_i - a_j - \tau_j \in (l^t_{ji}, u^t_{ji})$, $\tau_j + t_{ji} + a_i - a_j \ge \max\{\tau_j + t_{ji}, a_i - a_j\}$. The valid inequalities (6.7) – (6.9) all follow from (6.4). □

Finally, it is not difficult to verify that the following set of inequalities involving s_i's, which we called the *s-bound inequalities*, are also valid for S_{prp}.

$$\sum_{j \in V_0} s_j \ge y_i + \tau_i + \frac{d_{i0}}{u_{i0}}, \qquad \forall i \in V_0, \tag{6.10a}$$

$$\sum_{i \in V_0} s_i \ge \sum_{i \in V_0} (a_i + \tau_i + \frac{d_{i0}}{u_{i0}})x_{i0}, \tag{6.10b}$$

$$\sum_{i \in V_0} s_i - \sum_{i \in V_0} r_i \ge 0. \tag{6.10c}$$

7 Computational Experiments

We solve the three formulations (the big-M formulation (5.5) and the MISOCP formulation (5.9)) on three sets of instances in the literature. These instances are generated based on the geographical locations of United Kingdom cities, so we call these instances UK-A series, UK-B series, and UK-C series. Each series contain nine groups of instances with 10 to 200 cities, 20 instances in each group, and have a planning horizon of 32,400 seconds (9 hours). The UK-A series are created by Bektaş and Laporte (2011), and available at `http://www.apollo.management.soton.ac.uk/prplib.htm`. It has the widest time windows, with the width being around 20,000 seconds for each customer in the instances. The UK-B and UK-C series are created in Kramer et al. (2015b), and available at `http://w1.cirrelt.ca/~vidalt/resources/PRP-Instances-Krameretal.zip`. These instances have the same parameters as the UK-A series, except with narrower time windows: the widths of the time windows in the UK-B series are uniformly distributed between 2,000 and 5,000 seconds, and the widths of the time windows in the UK-C series are uniformly distributed between 2,000 and 15,000 seconds. The parameters in the objective functions of all formulations are from Demir et al. (2012), Kramer et al. (2015b):

- $\alpha_1 = 1.42469471 \times 10^{-3}$ (£/s),
- $\alpha_2 = 7.47047305 \times 10^{-5}$ (£/m),
- $\alpha_3 = 1.17645245 \times 10^{-8}$ (£/m/kg),
- $\alpha_4 = 1.97712815 \times 10^{-7}$ (£s^2/m^3),
- $p = 2.22222222 \times 10^{-3}$ (£/s).

In the rest of the paper, we use "UK-S-n-m" to refer to a particular instance, where S denotes the series of the instances, n is the total number of customers and m is the index of the instance. For example, UK-B-10-2 refers to the second 10-customer instance in the UK-B series.

The MISOCP formulation is implemented in C++ and solved by the MIQCP solver of CPLEX 12.6. We use the LP relaxation strategy (by setting the CPLEX parameter MIQCPStrat to 2), as we observe that this strategy gives a better performance than the QP relaxation strategy as well as the default strategy of CPLEX on our test instances. We add several families of cuts valid for the CVRP (rounded capacity inequalities, framed capacity inequalities, and strengthened comb inequalities) by calling a separation routine implemented in the package CVRPSEP (Lysgaard 2003). We also include all the cuts introduced in Section 6 as constraints for the MISOCP formulation.

Since the big-M formulation cannot be solved by CPLEX, they are implemented in GAMS and solved by the MICP solver COIN-OR Bonmin 1.7 with IPOPT 3.11 as the nonlinear programming solver and CBC as the mixed integer linear programming solver. We use the option `B-QG` in Bonmin (the branch-and-cut algorithm developed in Quesada and Grossmann (1992)), as it gives better performances than other algorithms for the PRP instances.

All our computational experiments are performed on a workstation with an Intel 3.40 GHz processor and 7.7 Gb memory running the Ubuntu Linux 12.04 operating system. For each instance, we impose a time limit of one hour on the computing time.

7.1 Preprocessing

Tightening the speed lower limit Since the fuel cost function $F_{ij}(v_{ij})$ in (3.1) attains the minimum at $v_F = (\frac{\alpha_1}{2\alpha_4})^{\frac{1}{3}}$ and is non-increasing when $v_{ij} \le v_F$, a vehicle will never travel below v_F

in an optimal solution of any PRP instance (otherwise it will travel at v_F instead and wait, with less fuel and emission costs and the same labor cost). Note that v_F is the same for each arc and can be computed in advance. Therefore, for each $(i, j) \in A$ such that $l_{ij} < v_F$, we set $l_{ij} = v_F$.

Tightening the time windows of each instance We tighten the time windows following a similar procedure for the TSPTW, described in Ascheuer et al. (2001). A slight difference is that for the PRP we cannot assume that the service time for each customer is 0, since the service time cannot be incorporated into the travel duration. For each instance, we repeat the following tightening steps until no more changes can be made:

- $a_k \leftarrow \max\{a_k, \min_{i\in\delta^-(k)}\{a_i + \tau_i + \frac{d_{ik}}{u_{ik}}\}\}$ for $k \in V$,
- $a_k \leftarrow \max\{a_k, \min_{j\in\delta^+(k)}\{a_j - \frac{d_{kj}}{l_{kj}} - \tau_k\}\}$ for $k \in V$,
- $b_k \leftarrow \min\{b_k, \max\{a_k, \max_{i\in\delta^-(k)}\{b_i + \tau_i + \frac{d_{ik}}{l_{ik}}\}\}\}$ for $k \in V$,
- $b_k \leftarrow \min\{b_k, \max_{j\in\delta^+(k)}\{b_j - \frac{d_{kj}}{u_{kj}} - \tau_k\}\}$ for $k \in V$.

Strengthening the big-M constants In the MISOCP formulation (5.9), $t_{ij} = 0$ when $x_{ij} = 0$, so the big-M constants can be strengthened to $M_{ij} = \max\{0, b_i + \tau_i - a_j\}$ and $M_{i,0} = b_i + \tau_i$. This strengthening is not applicable to the big-M formulation (5.5).

7.2 Comparison between the big-M formulation (5.5) and the MISOCP formulation (5.9) for the original PRP

In the original PRP proposed in Bektaş and Laporte (2011), the departure time at the depot for each route is always 0. Thus we first compare the performances of three formulations for the original PRP, by setting the departure time at the depot $r_i = 0$ for $i \in V_0$.

Table 1 illustrates a detailed comparison between the performances of the MISOCP formulation and the big-M formulation on UK-B-10 instances. In the table, columns **Time** and **Nodes** illustrate the total time (in seconds) it took to solve the instance and the total number of nodes explored in the branch-and-bound tree, respectively; column **R-Gap-B** represents the integrality gap given by the continuous relaxation of each formulation; column **R-Gap-A** represents the integrality gap at the root node after adding CPLEX generated cuts, cuts from CVRPSEP package, and additional valid inequalities that we derive in Section 6.

The UK-B instances with 10 customers can be solved in less than 100 seconds using the MISOCP formulation and the big-M formulation. Note that the continuous relaxations of the two formulations are quite weak, despite the fact that the continuous relaxation of the MISOCP formulation is stronger than that of the big-M formulation. CPLEX, however, can strengthen the MISOCP formulation significantly through preprocessing and adding cuts. This can also be seen from the computational results even without using cuts from the CVRPSEP package or the additional valid inequalities derived in Section 6. Bonmin with option `B-QG`, on the other hand, does not add any cuts at the root node, so each entry in column R-Gap-A is the same as the corresponding entry in column R-Gap-B. We also observe that the UK-A and UK-C instances with 10 customers can only be solved to optimality within the time limit using the MISOCP formulation. Overall, the largest instances that can be solved to optimality using the big-M formulation are 10-customer instances, while the MISOCP formulation can be solved to optimality on instances with up to 25 customers, as illustrated in the next section.

Table 1: Computational results of the MISOCP formulation (5.9) and the big-M formulation (5.5) with fixed departure time at the depot for UK-B 10-customer instances

Instance	K	MISOCP (5.9)		MISOCP (5.9) with cuts				Big-M (5.5)		
		Time	Nodes	Time	Nodes	R-Gap-B	R-Gap-A	Time	Nodes	R-Gap
UK-B-10-1	2	0.2	9	0.2	1	59.9%	0.4%	10.3	368	68.4%
UK-B-10-2	2	0.2	0	0.0	0	47.3%	0.0%	10.4	490	62.8%
UK-B-10-3	3	0.1	0	0.1	1	39.5%	0.1%	7.2	290	52.6%
UK-B-10-4	2	0.3	14	0.3	4	57.7%	2.7%	18.8	1709	69.0%
UK-B-10-5	2	0.2	1	0.2	1	59.8%	0.3%	11.5	752	68.0%
UK-B-10-6	3	0.2	1	0.1	0	37.2%	0.0%	7.7	653	52.0%
UK-B-10-7	3	0.6	79	0.3	34	46.3%	7.3%	73.3	16K	57.3%
UK-B-10-8	2	0.2	1	0.1	1	48.2%	1.2%	12.9	1104	62.7%
UK-B-10-9	2	0.2	1	0.1	1	56.9%	0.1%	7.1	362	67.3%
UK-B-10-10	2	0.2	1	0.2	1	54.4%	0.6%	8.7	453	66.7%
UK-B-10-11	3	0.2	8	0.0	0	39.0%	0.0%	15.1	2174	54.0%
UK-B-10-12	2	0.2	4	0.1	1	58.7%	0.1%	9.3	375	67.5%
UK-B-10-13	3	0.3	25	0.3	1	46.4%	1.7%	8.2	700	57.8%
UK-B-10-14	2	0.4	6	0.2	1	50.5%	0.1%	11.7	330	65.3%
UK-B-10-15	2	0.0	0	0.0	0	57.2%	0.0%	7.2	237	66.8%
UK-B-10-16	2	0.4	41	0.3	7	55.7%	2.0%	12.3	502	66.3%
UK-B-10-17	2	0.4	5	0.2	1	55.4%	0.2%	10.5	863	67.7%
UK-B-10-18	2	0.3	6	0.1	1	60.5%	0.7%	9.2	595	69.0%
UK-B-10-19	3	0.2	0	0.0	0	53.6%	0.0%	11.9	1254	62.7%
UK-B-10-20	2	0.3	3	0.2	1	55.5%	2.2%	11.9	170	65.0%

7.3 Comparison between the MISOCP formulation (5.9) and the MIP formulation with speed discretization for the original PRP

Since the MISOCP formulation (5.9) is much stronger and yields a better computational performance than the big-M formulation (5.5), we solve several variants of the PRP based on this formulation in the rest of the paper. In Table 2, we present the results of the MISOCP formulation on UK-B instances for the original PRP. Results on UK-A and UK-C instances for the original PRP are shown in Tables 4 and 5 in the Appendix. In all these tables, column **Time (s)/O-Gap** shows the total time (in seconds) it takes to solve the instance or the final optimality gap if the time limit is reached (shown with a % sign), column **R-Gap** shows the integrality gap at the root node, column **Nodes** shows the total number of nodes explored in the branch-and-bound tree, and column **Obj** shows the optimal value or the best solution found if the time limit is reached. In Table 2, we also show the computational performance of the MIP formulation based on speed discretization proposed in Bektaş and Laporte (2011). Column **Obj1** shows the optimal objective value of the MIP formulation with speed discretization, and column **Obj2** shows the optimal objective value calculated by speed optimization using the optimal route obtained by the MIP formulation.

We see from Table 2 that all the UK-B instances with 25 customers are solved to optimality within the time limit. Tables 4 and 5 in the Appendix show that all the 10-customer UK-A and UK-C instances can be solved to optimality within the one-hour time limit, and some 20-customer UK-C instances can be solved to optimality as well. To the best of our knowledge, it is the first time that proven optimal solutions are reported for these instances when the speed over each arc is considered as a continuous variable.

Comparing the computational performances between the MISOCP formulation (5.9) and the MIP formulation Bektaş and Laporte (2011), we see that in all these instances, although the optimal objective values given by the MIP formulation (shown in column **Obj1**) are slightly different from

Table 2: Computational results for the MISOCP formulation (5.9) and the MIP formulation in Bektaş and Laporte (2011) for UK instance family B

Instance	K	MISOCP (5.9)				MIP Bektaş and Laporte (2011)				
		Obj	Time	Nodes	R-Gap	Obj1	Obj2	Time	Nodes	R-Gap
UK-B-25-1	5	452.09	29.4	965	4.8%	452.09	452.09	5.9	1661	7.1%
UK-B-25-2	5	497.98	710.2	20K	5.6%	498.07	497.98	163.4	28457	8.2%
UK-B-25-3	5	367.15	39.8	1026	4.8%	367.37	367.15	42.6	4359	11.5%
UK-B-25-4	5	438.75	36.4	1666	9.8%	439.13	438.75	15.5	2949	12.5%
UK-B-25-5	5	462.64	36.5	1311	9.1%	462.79	462.64	34.9	5949	12.5%
UK-B-25-6	5	480.88	69.1	3279	5.1%	480.90	480.88	42.5	7186	6.9%
UK-B-25-7	5	484.39	149.6	6706	5.5%	484.51	484.39	87.3	14130	7.9%
UK-B-25-8	5	512.94	22.7	873	3.9%	513.12	512.94	16.8	3240	6.9%
UK-B-25-9	5	416.17	315.7	9478	8.4%	416.30	416.17	252.4	32364	10.9%
UK-B-25-10	5	495.64	310.9	7989	6.1%	495.69	495.64	75.5	12593	9.7%
UK-B-25-11	5	511.44	466.1	10K	5.9%	511.46	511.44	102.2	21142	9.5%
UK-B-25-12	5	600.81	9.7	178	3.6%	600.81	600.81	14.5	1360	6.6%
UK-B-25-13	5	389.01	670.9	31K	8.4%	389.12	389.01	357.2	71101	12.3%
UK-B-25-14	5	548.96	4.7	57	2.7%	549.12	548.96	2.4	725	7.6%
UK-B-25-15	5	553.37	7.6	225	2.8%	553.65	553.37	1.6	312	5.3%
UK-B-25-16	5	520.36	35.2	1360	4.7%	520.59	520.36	37.3	6307	9.9%
UK-B-25-17	5	624.55	13.9	515	3.2%	624.90	624.55	15.2	1724	5.3%
UK-B-25-18	5	540.33	4.5	97	2.6%	540.58	540.33	1.1	243	4.5%
UK-B-25-19	5	610.11	9.5	378	3.8%	610.41	610.11	15.6	1647	9.6%
UK-B-25-20	5	521.02	264.4	8119	6.8%	521.22	521.02	116.3	18779	10.8%

the true optimal ones (shown in column **Obj**), the optimal routes from the MIP formulation with speed discretization are the same optimal routes from the MISOCP formulation (without discretizing the speed). In addition, the MIP formulation takes less time to solve than the MISOCP formulation (5.9). This suggests that the MIP formulation based on speed discretization is a very good heuristic approach, when the speed over each arc is considered as a continuous variable. The proposed MISOCP formulation (and the big-M based MICP formulation) can serve as a theoretical benchmark for this heuristic approach.

We also observe that the length of the time windows is a good indicator for the difficulty of the instance, in addition to the number of customers: the wider the time window is, the longer it takes to solve the instance. With the same number of customers, the UK-B instances yield the smallest average solution time, the smallest average number of nodes explored, and the smallest root gap among all three series. On the other hand, the UK-A instances have the widest time windows, and is challenging to solve even with 20 customers. This is demonstrated not only with our MICP formulations—in the recent branch-and-price algorithm developed in Dabia et al. (2014), the *root LPs* of 9 out of 20 UK-A instances with 15 cities cannot be solved in 2 hours, assuming the same speed over each arc of a route.

7.4 The impact of labor cost and flexible departure time at the depot

In this section, we investigate how the optimal solution of the PRP and the computational performance of the MISOCP formulation are affected by the following two factors: driver wages in the objective function, and a flexible departure time at the depot. A higher vehicle speed (higher than v_F) will lead to more fuel consumption and GHG emissions, and on the other hand help satisfy some customer's time-window constraint, giving rise to a shorter route, less travel duration, and less driver wages. Thus an optimal solution of the PRP should strike the balance between the two parts of costs. It has been demonstrated in Bektaş and Laporte (2011) that the PRP with fixed departure time at the depot becomes much easier to solve when driver wages are ignored. So we solve the MISOCP formulation of the PRP under three different settings: driver wages are ignored (by setting $p = 0$) and the vehicle departure time at the depot is fixed to 0; driver wages are included in the cost and the vehicle departure time at the depot is fixed to 0 (the original PRP); driver wages is included in the cost and the vehicle departure time from the depot is considered as a variable. Table 3 reports the computational results using the MISOCP formulation on the 10-customer instances using these three settings.

From Table 3, we can see that the total cost reduction by allowing flexible departure time at the depot is quite significant. We also observe that the inclusion of driver wages and flexible departure time at the depot significantly increase the difficulty of the instances, especially for the UK-A instances which have wider time windows.

We now look more closely at the optimal solutions for one instance under these three settings. We illustrate each optimal solution in the following way: if the service start time at customer j is the earliest possible time a_j, we denote it by $|j$; if the service start time at customer j is the latest possible time b_j, we denote it by $j|$; if the vehicle waits at customer j before starting service, we denote it by $(w)j$; if the vehicle waits at customer j and the service start time is exactly a_j, we denote it by $(w)|j$. In addition, we compute two special speeds in advance (Kramer et al. 2015b): $v_F = (\frac{\alpha_1}{2\alpha_4})^{\frac{1}{3}}$, which minimizes the fuel and emission costs over an arc, and $v_{FD} = (\frac{\alpha_1+p}{2\alpha_4})^{\frac{1}{3}}$, which minimizes the sum of fuel and emission costs and driver wages over an arc. Below are the optimal solutions under three different settings for instance UK-10-B-1 with 3 vehicles.

1. Driver wages are not considered and the departure time at the depot is fixed to 0.
   - Route 1: $0 \xrightarrow{v_F} (w)5 \xrightarrow{v_F} (w)2 \xrightarrow{v_F} 0$
   - Route 2: $0 \xrightarrow{v_F} (w)6 \xrightarrow{v_F} (w)|4 \xrightarrow{21.2709} 9 \xrightarrow{21.2709} 7| \xrightarrow{v_F} 0$
   - Route 3: $0 \xrightarrow{v_F} (w)8 \xrightarrow{v_F} (w)3 \xrightarrow{v_F} (w)10 \xrightarrow{v_F} (w)1 \xrightarrow{v_F} 0$
2. Driver wages are considered and the departure time at the depot is fixed to 0.
   - Route 1: $0 \xrightarrow{v_F} (w)3 \xrightarrow{v_F} (w)|10 \xrightarrow{19.0939} 1 \xrightarrow{19.0939} 5 \xrightarrow{19.0939} |2 \xrightarrow{v_{FD}} 0$
   - Route 2: $0 \xrightarrow{v_F} (w)6 \xrightarrow{v_F} (w)|4 \xrightarrow{21.2708} 9 \xrightarrow{21.2708} 7| \xrightarrow{v_{FD}} 0$
   - Route 3: $0 \xrightarrow{v_F} (w)|8 \xrightarrow{v_{FD}} 0$
3. Driver wages are considered and the departure time at the depot is flexible.
   - Route 1: $0 \xrightarrow{v_{FD}} 3| \xrightarrow{18.5418} |10 \xrightarrow{v_{FD}} 1 \xrightarrow{v_{FD}} 0$
   - Route 2: $0 \xrightarrow{v_{FD}} 5 \xrightarrow{v_{FD}} 2 \xrightarrow{v_{FD}} 0$
   - Route 3: $0 \xrightarrow{v_{FD}} 8| \xrightarrow{19.9581} 6| \xrightarrow{v_F} (w)|4 \xrightarrow{21.2709} 9 \xrightarrow{21.2709} 7| \xrightarrow{v_{FD}} 0$

Under the three different settings, we observe that not only the optimal speeds are different, but also the routes themselves. Note that the optimal speed over many arcs is either v_F or v_{FD}. This could be a useful property to explore for developing efficient specialized algorithms for the PRP (Fukasawa et al. 2016).

Table 3: Computational performance of the MISOCP formulation (5.9) with no driver wages and fixed departure time ("NDW+FDT"), with driver wages and fixed departure time ("DW+FDT"), and with driver wages and flexible departure time ("DW+VDT")

Instance	K	NDW+FDT			DW+FDT			DW+VDT		
		Obj†	Time	Nodes	Obj	Time	Nodes	Obj	T/Gap	Nodes
UK10-A-1	2	207.533	0.7	9	170.641	1354.4	297K	168.82	35.1%	525K
UK10-A-2	2	234.929	0.5	2	204.877	813.7	200K	203.824	29.8%	569K
UK10-A-3	3	268.346	0.1	0	202.562	1708.3	299K	198.351	30.6%	729K
UK10-A-4	2	229.81	0.9	61	189.884	844.9	187K	186.594	31.0%	454K
UK10-A-5	2	221.681	0.9	18	175.59	2649.2	573K	172.894	36.4%	535K
UK10-A-6	2	253.442	2.3	468	214.481	1472.8	302K	209.363	28.3%	604K
UK10-A-7	2	220.13	0.6	8	190.144	882.5	236K	190.044	29.0%	548K
UK10-A-8	2	260.814	0.2	1	222.168	564.3	130K	222.168	23.8%	571K
UK10-A-9	2	217.626	0.9	59	174.541	352.0	83K	174.541	27.4%	615K
UK10-A-10	2	236.358	0.2	0	189.816	211.1	67K	189.78	26.3%	708K
Avg. cost reduction						17.7%			>0.9%	
UK10-B-1	2	252.744	0.1	0	246.445	0.2	1	217.959	0.3	1
UK10-B-2	2	307.093	0.1	1	303.732	0.0	0	266.738	0.2	1
UK10-B-3	3	311.718	0.0	0	301.891	0.0	0	261.496	0.2	1
UK10-B-4	2	282.917	0.2	8	273.905	0.2	4	256.779	0.2	2
UK10-B-5	2	263.246	0.1	1	255.074	0.1	1	254.529	0.1	1
UK10-B-6	3	340.941	0.0	0	332.340	0.0	0	287.072	0.1	0
UK10-B-7	3	323.944	0.1	7	314.641	0.0	0	239.138	0.5	30
UK10-B-8	2	342.274	0.1	1	339.364	0.1	1	309.095	0.1	1
UK10-B-9	2	267.852	0.0	0	261.101	0.1	1	250.725	0.2	1
UK10-B-10	2	291.173	0.2	1	285.202	0.1	1	261.668	0.2	1
Avg. cost reduction						3.5%			9.2%	
UK10-C-1	2	229.976	0.3	1	210.183	1.6	213	171.956	51.2	13K
UK10-C-2	2	279.439	8.6	1929	271.93	72.4	11K	230.525	85.5	14K
UK10-C-3	2	249.733	0.6	6	229.178	2.5	455	204.972	24.7	4499
UK10-C-4	2	246.149	0.6	19	230.52	5.2	862	187.883	13.2	3670
UK10-C-5	2	235.864	0.6	62	205.491	7.6	2164	184.098	89.1	24K
UK10-C-6	2	271.635	0.6	53	255.819	17.5	3500	236.541	49.9	8423
UK10-C-7	2	230.294	0.5	11	217.788	14.3	2858	190.044	41.0	7834
UK10-C-8	2	262.231	0.8	75	251.294	4.7	1073	244.998	24.3	4212
UK10-C-9	2	212.367	0.3	0	186.041	5.8	1100	174.541	23.1	5875
UK10-C-10	2	239.78	0.3	4	231.619	3.8	748	201.44	33.0	6096
Avg. cost reduction						7.0%			11.5%	

'†': This objective includes the fuel and emission costs and the driver wages calculated based on the optimal solution

7.5 Discussion on continuous speeds

We would like to discuss the practical relevance of continuous speeds as decision variables in the PRP. This is more related to the PRP model itself instead of the methodology we propose in this paper. On the one hand, it will be difficult to dictate a speed of 76.4 km/h in some practical problems. A model with continuous speeds nonetheless provides an estimate on the maximum cost savings by operating on flexible speeds. On the other hand, in applications where there is more freedom on speed control, such as aircraft cruise control (Aktürk et al. 2014), maritime transportation (Norstad et al. 2011, Meng et al. 2013), and future applications of autonomous vehicles on highways, a model with continuous speeds seems more reasonable. Then the approach with speed discretization serves as a heuristic for the model.

8 Conclusions and future research

In this paper, we studied the pollution routing problem, an important variant of the vehicle routing problem aiming at minimizing the total cost of fuel consumptions, GHG emissions and driver wages. The introduction of vehicle speeds as additional decision variables in the PRP significantly complicated the traditional routing models. By keeping the speed decision as a continuous variable, we proposed the first arc-based MICP formulations for the PRP, motivated by recent advances in disjunctive convex programming. We performed extensive computational experiments to evaluate the effectiveness of these formulations, among which the MISOCP formulation can solve PRP instances with up to 25 customers. Instances with wide time windows are in general much more difficult to solve.

We have identified several research directions that are worth exploring in the future. First, larger-size instances with wide time windows are still challenging to solve because of weak relaxation bounds. Recently significant advances have been made on deriving cutting planes by exploring the underlying conic structure of the problem. The strength of the MISOCP formulation could be improved if these cutting planes can be incorporated. Second, we observe that for many instances, the optimal routes under three settings (no driver wages and fixed depot departure time, with driver wages and fixed depot departure time, and with driver wages and flexible depot departure time) are identical. This interesting observation motivates a further investigation on conditions under which the optimal routes remain the same with driver wages and varying departure time at the depot. If such a general condition exists, then one could solve the more challenging PRP variant with driver wages and varying depot departure time through the following scheme: solve the PRP without driver wages and depot departure time fixed at 0, which is much easier, and obtain the optimal routes; then solve a speed and departure time optimization problem over the given routes.

References

K. Abhishek, S. Leyffer, and J. T. Linderoth. FilMINT: An outer approximation-based solver for convex mixed-integer nonlinear programs. *INFORMS Journal on Computing*, 22(4):555–567, 2010.

M. S. Aktürk, A. Atamtürk, and S. Gürel. Parallel machine match-up scheduling with manufacturing cost considerations. *Journal of Scheduling*, 13(1):95–110, 2010.

M. S. Aktürk, A. Atamtürk, and S. Gürel. Aircraft rescheduling with cruise speed control. *Operations Research*, 62(4):829–845, 2014.

J. F. Alvarez, T. Longva, and E. S. Engebrethsen. A methodology to assess vessel berthing and speed optimization policies. *Maritime Economics & Logistics*, 12(4):327–346, 2010.

N. Ascheuer, M. Fischetti, and M. Grötschel. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, 90(3):475–506, 2001.

A. Atamtürk, G. Berenguer, and Z.-J. Shen. A conic integer programming approach to stochastic joint location-inventory problems. *Operations Research*, 60(2):366–381, 2012.

E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical programming*, 58(1-3):295–324, 1993.

M. Barth, G. Scora, and T. Younglove. Modal emissions model for heavy-duty diesel vehicles. *Transportation Research Record: Journal of the Transportation Research Board*, 1880(1):10–20, 2004.

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

P. Belotti, C. Kirches, S. Leyffer, J. T. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.

P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

P. Bonami, A. Lodi, A. Tramontani, and S. Wiese. On mathematical programming with indicator constraints. *Mathematical Programming*, 151(1):191–223, 2015.

P. Cariou. Is slow steaming a sustainable means of reducing co 2 emissions from container shipping? *Transportation Research Part D: Transport and Environment*, 16(3):260–264, 2011.

S. Ceria and J. Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86(3):595–614, 1999.

M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation science*, 38(1):1–18, 2004.

S. Dabia, E. Demir, and T. Van Woensel. An exact approach for the pollution-routing problem. Technical report, School of Industrial Engineering, Eindhoven University of Technology, 2014.

R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8 (3):250–255, 1965.

E. Demir, T. Bektaş, and G. Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359, 2012.

E. Demir, T. Bektaş, and G. Laporte. The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3):464–478, 2014a.

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

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

Y. Du, Q. Chen, X. Quan, L. Long, and R. Y. Fung. Berth allocation considering fuel consumption and vessel emissions. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1021–1037, 2011.

R. Eglese and T. Bektaş. Green vehicle routing. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS/SIAM Series on Optimization, pages 437–458. SIAM, 2nd edition, 2014.

J. F. Ehmke, A. M. Campbell, and B. W. Thomas. Vehicle routing to minimize time-dependent expected emissions in urban areas. *European Journal of Operational Research*, 251(2):478–494, 2016.

Environmental Protection Agency. Inventory of U.S. greenhouse gas emissions and sinks: 1990-2013. `http://www.epa.gov/climatechange/Downloads/ghgemissions/US-GHG-Inventory-2015-Main-Text.pdf`, 2015.

K. Fagerholt, G. Laporte, and I. Norstad. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3):523–529, 2010.

R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1-3):327–349, 1994.

A. Franceschetti, D. Honhon, T. Van Woensel, T. Bektaş, and G. Laporte. The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293, 2013.

R. Fukasawa, Q. He, and Y. Song. A branch-cut-and-price algorithm for the energy minimization vehicle routing problem. *Transportation Science*, 2015. `http://pubsonline.informs.org/doi/abs/10.1287/trsc.2015.0593`.

R. Fukasawa, Q. He, F. Santos, and Y. Song. A joint routing and speed optimization problem. *arXiv preprint arXiv:1602.08508*, 2016.

A. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4): 237–260, 1972.

I. E. Grossmann, J. Viswanathan, A. Vecchietti, R. Raman, and E. Kalvelagen. GAMS/DICOPT: A discrete continuous optimization package. *GAMS Corporation Inc.*, 2002.

O. Günlük and J. T. Linderoth. Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical programming*, 124(1-2):183–205, 2010.

O. K. Gupta and A. Ravindran. Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31(12):1533–1546, 1985.

Gurobi Optimization. The gurobi optimizer. `https://www.gurobi.com`, 2015.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I: Part 1: Fundamentals*. Springer Science & Business Media, 1996.

L. M. Hvattum, I. Norstad, K. Fagerholt, and G. Laporte. Analysis of an exact algorithm for the vessel speed optimization problem. *Networks*, 62(2):132–135, 2013.

IBM ILOG. CPLEX high-performance mathematical programming engine. `http://www-01.ibm.com/software/in/integration/optimization/cplex/`, 2015.

O. Jabali, T. Van Woensel, and A. de Kok. Analysis of travel times and co2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074, 2012.

İ. Kara, B. Y. Kara, and M. K. Yetiş. Energy minimizing vehicle routing problem. In A. Dress, Y. Xu, and B. Zhu, editors, *Proceedings of the First International Conference on Combinatorial Optimization and Applications*, volume 4616 of *Lecture Notes in Computer Science*, pages 62–71. Springer Berlin Heidelberg, Xi'an, China, 2007.

İ. Kara, B. Y. Kara, and M. K. Yetiş. Cumulative vehicle routing problems. In T. Carić and H. Gold, editors, *Vehicle Routing Problem*, pages 85–98. In-Teh, 2008.

H.-J. Kim, Y.-T. Chang, K.-T. Kim, and H.-J. Kim. An epsilon-optimal algorithm considering greenhouse gas emissions for the management of a ships bunker fuel. *Transportation Research Part D: Transport and Environment*, 17(2):97–103, 2012.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, 2014.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B: Methodological*, 84:81–102, 2016.

C. Kontovas and H. N. Psaraftis. Reduction of emissions along the maritime intermodal container chain: operational models and policies. *Maritime Policy & Management*, 38(4):451–469, 2011.

R. Kramer, N. Maculan, A. Subramanian, and T. Vidal. A speed and departure time optimization algorithm for the pollution-routing problem. *European Journal of Operational Research*, 247(3):782–787, 2015a.

R. Kramer, A. Subramanian, T. Vidal, and L. d. A. F. Cabral. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539, 2015b.

G. Laporte. Scheduling issues in vehicle routing. *Annals of Operations Research*, 236(2):463–474, 2016.

C.-Y. Lee and D.-P. Song. Ocean container transport in global supply chains: Overview and research opportunities. *Transportation Research Part B: Methodological*, 2016.

J. Lee and S. Leyffer, editors. *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*. Springer-Verlag New York, 2012.

C. Lin, K. Choy, G. T. Ho, S. Chung, and H. Lam. Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, 41(4):1118–1138, 2014.

J. Lysgaard. CVRPSEP: A package of separation routines for the capacitated vehicle routing problem. `http://www.hha.dk/~lys/CVRPSEP.htm`, 2003.

Q. Meng, S. Wang, H. Andersson, and K. Thun. Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2):265–280, 2013.

Q. Meng, S. Wang, and C.-Y. Lee. A tailored branch-and-price approach for a joint tramp ship routing and bunkering problem. *Transportation Research Part B: Methodological*, 72:1–19, 2015.

I. Norstad, K. Fagerholt, and G. Laporte. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19(5):853–865, 2011.

T. E. Notteboom and B. Vernimmen. The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, 17(5):325–337, 2009.

I. Quesada and I. E. Grossmann. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers & Chemical Engineering*, 16(10):937–947, 1992.

D. Ronen. The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society*, 62(1):211–216, 2011.

J. P. Vielma, S. Ahmed, and G. L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

S. Wang. Fundamental properties and pseudo-polynomial-time algorithm for network containership sailing speed optimization. *European Journal of Operational Research*, 250(1):46–55, 2016.

S. Wang and Q. Meng. Liner ship route schedule design with sea contingency time and port time uncertainty. *Transportation Research Part B: Methodological*, 46(5):615–633, 2012a.

S. Wang and Q. Meng. Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):701–714, 2012b.

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

A Computational results of the MISOCP formulation for UK-A and UK-C instances

Table 4: Computational results for MISOCP formulation (5.9) for UK instance family A

Instances	K	Obj	Time/ O-Gap	Nodes	R-Gap	Instances	K	Obj	Time/ O-Gap	Nodes	R-Gap
UK-A-10-1	2	170.641	1354.4	297K	30.2%	UK-A-10-11	2	262.079	437.7	94K	28.7%
UK-A-10-2	2	204.877	813.7	200K	28.3%	UK-A-10-12	2	183.189	2412.8	429K	29.3%
UK-A-10-3	2	202.562	1708.3	299K	32.6%	UK-A-10-13	2	195.97	6.8%	655K	28.9%
UK-A-10-4	2	189.884	844.9	187K	29.8%	UK-A-10-14	2	163.167	424.1	143K	27.8%
UK-A-10-5	2	175.59	2649.2	573K	28.0%	UK-A-10-15	2	127.103	777.4	230K	27.4%
UK-A-10-6	2	214.481	1472.8	302K	28.7%	UK-A-10-16	2	186.633	2562.7	490K	30.0%
UK-A-10-7	2	190.144	882.5	236K	29.1%	UK-A-10-17	2	159.034	90.8	36K	25.2%
UK-A-10-8	2	222.168	564.3	130K	27.9%	UK-A-10-18	2	162.086	1073.3	266K	29.5%
UK-A-10-9	2	174.541	352.0	83K	24.9%	UK-A-10-19	2	169.464	479.5	113K	28.0%
UK-A-10-10	2	189.816	211.1	67K	27.7%	UK-A-10-20	2	168.798	3243.8	625K	32.2%
UK-A-20-1	5	352.447	22.9%	>103K	26.4%	UK-A-20-11	5	401.027	22.1%	>81K	27.2%
UK-A-20-2	5	365.767	20.7%	>90K	24.4%	UK-A-20-12	5	345.137	19.6%	>106K	26.9%
UK-A-20-3	5	230.486	23.6%	>96K	28.4%	UK-A-20-13	5	340.69	25.7%	>75K	29.7%
UK-A-20-4	5	347.043	21.2%	>109K	25.0%	UK-A-20-14	5	420.64	23.3%	>75K	27.7%
UK-A-20-5	5	329.626	24.3%	>113K	29.3%	UK-A-20-15	5	347.281	22.1%	>100K	26.4%
UK-A-20-6	5	367.732	25.0%	>60K	27.7%	UK-A-20-16	5	352.689	18.3%	>131K	23.4%
UK-A-20-7	5	258.747	23.3%	>78K	26.2%	UK-A-20-17	5	413.325	20.6%	>103K	22.5%
UK-A-20-8	5	303.174	23.0%	>101K	27.2%	UK-A-20-18	5	365.82	17.3%	>121K	22.9%
UK-A-20-9	5	362.562	19.5%	>112K	23.7%	UK-A-20-19	5	337.664	22.2%	>99K	26.1%
UK-A-20-10	5	317.786	26.3%	>93K	30.1%	UK-A-20-20	5	369.829	21.3%	>123K	25.8%

Table 5: Computational results for MISOCP formulation (5.9) for UK instance family C

Instances	K	Obj	Time/ O-Gap	Nodes	R-Gap	Instances	K	Obj	Time/ O-Gap	Nodes	R-Gap
UK-C-10-1	2	210.183	1.6	213	8.3%	UK-C-10-11	2	298.199	11.7	2487	20.1%
UK-C-10-2	2	271.93	72.4	11K	26.8%	UK-C-10-12	2	206.583	6.3	1318	18.2%
UK-C-10-3	2	229.178	2.5	455	18.5%	UK-C-10-13	2	211.751	27.8	5588	20.6%
UK-C-10-4	2	230.52	5.2	862	19.8%	UK-C-10-14	2	209.067	22.6	5676	21.6%
UK-C-10-5	2	205.491	7.6	2164	16.7%	UK-C-10-15	2	176.555	5.4	856	20.1%
UK-C-10-6	2	255.819	17.5	3500	19.0%	UK-C-10-16	2	229.151	52.2	9540	21.2%
UK-C-10-7	2	217.788	14.3	2858	20.4%	UK-C-10-17	2	219.196	12.6	3037	26.3%
UK-C-10-8	2	251.294	4.7	1073	13.2%	UK-C-10-18	2	195.044	6.8	1801	20.8%
UK-C-10-9	2	186.041	5.8	1100	19.5%	UK-C-10-19	2	218.194	53.6	7970	30.1%
UK-C-10-10	2	231.619	3.8	748	20.4%	UK-C-10-20	2	189.56	19.2	3782	25.5%
UK-C-20-1	5	432.821	6.8%	118K	21.1%	UK-C-20-11	5	488.069	5.0%	97K	19.0%
UK-C-20-2	5	450.355	12.7%	89K	21.4%	UK-C-20-12	5	420.773	7.8%	101K	15.0%
UK-C-20-3	5	287.037	9.9%	123K	17.9%	UK-C-20-13	5	417.646	8.7%	93K	19.6%
UK-C-20-4	5	434.233	7.3%	86K	17.1%	UK-C-20-14	5	488.946	3.8%	87K	16.4%
UK-C-20-5	5	382.462	10.9%	85K	20.0%	UK-C-20-15	5	420.328	10.3%	87K	22.0%
UK-C-20-6	5	444.355	14.9%	86K	23.0%	UK-C-20-16	5	445.489	0.5%	91K	12.5%
UK-C-20-7	5	321.666	17.3%	82K	24.6%	UK-C-20-17	5	502.792	10.6%	73K	15.1%
UK-C-20-8	5	410.347	1224.7	58K	18.2%	UK-C-20-18	5	459.562	7.7%	92K	19.0%
UK-C-20-9	5	421.385	3347.8	111K	12.0%	UK-C-20-19	5	445.842	12.5%	69K	20.0%
UK-C-20-10	5	390.676	7.2%	88K	18.9%	UK-C-20-20	5	460.849	1436.7	60K	15.5%