Computers and Operations Research 108 (2019) 226–237

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

On two new formulations for the fixed charge network design problem with shortest path constraints

Ikram Bouras[a,*], Rosa Figueiredo[b], Michael Poss[a], Fen Zhou[b]

[a] LIRMM, University of Montpellier, 161 rue Ada, Montpellier Cedex 5 34095, France
[b] LIA, University of Avignon, Avignon 84911, France

ARTICLE INFO

Article history:
Received 26 April 2018
Revised 8 February 2019
Accepted 7 April 2019
Available online 16 April 2019

Keywords:
Network design
Bi-level programming
Cutting plane
Branch-and-cut

ABSTRACT

We study the fixed charge network design problem with shortest path constraints which is modeled as a bi-level program. We first review three one-level formulations obtained by applying the complementarity slackness theorem, Bellman's optimality conditions and cycle elimination constraints. We propose two new binary integer programming (BILP) formulations inspired by path and cycle inequalities. The two formulations have exponential numbers of constraints. We incorporate the path and the cycle based formulations in a branch-and-cut algorithm and in another cutting-plane based method. Numerical experiments are performed on real instances, and random data sets generated with different criteria to examine the difficulty of the instances. The results show that the proposed cutting plane algorithms can solve up to 19% more instances than the classic branch-and-bound algorithms.

1. Introduction

The fixed charge network design problem (FCNDP) consists of selecting a subset of edges from a given network, in such a way that a set of commodities can be transported from its origins to its destinations. The objective is to minimize the sum of fixed costs (depending on selected edges) and variable costs (depending on the flow of commodities on the edges). In general, fixed and variable costs can be represented by linear functions, and the arcs are uncapacitated.

There are several variations of FCNDP in the literature, each of which involves a particular objective function and, possibly, additional constraints. Among these variations, we can find the shortest path problem, minimum spanning tree problem, vehicle routing problem, traveling salesman problem and Steiner problem in graph (Ahuja et al., 1993; Bazaraa et al., 2011; Magnanti and Wong, 1984). Numerous applications can be found for network design problems, for instance in transportation systems and telecommunication networks (see Batta and Kwon, 2013; Magnanti and Wong, 1984).

We are interested in a specific variant of the FCNDP, called fixed charge network design problem with shortest path constraints (FCNDP-SPC) (González et al., 2016; LeBlanc and Boyce, 1986), which consists of adding multiple shortest path problems to the original problem.

The FCNDP-SPC involves two distinct agents acting simultaneously rather than sequentially when making decisions. On the upper level, the leader (first agent) is in charge of designing a transportation subnetwork (*i.e.*, choosing a subset of edges to be opened) in order to minimize the sum of fixed and variable costs. In response, on the lower level, the follower (second agent) must choose a set of shortest paths in the subnetwork designed in the upper level, through which the commodities will be sent.

The inclusion of the shortest path requirement makes the problem more difficult to solve exactly. There are few works done for solving exactly the general case of FCNDP-SPC (Amaldi et al., 2011; González et al., 2016; LeBlanc and Boyce, 1986; Marcotte, 1988; Mauttone et al., 2008), and most of the research is dedicated to a particular application. In the literature, the FCNDP-SPC was investigated for the transportation of hazardous materials: the Hazmat transport network design problem (HTNDP) (Bianco et al., 2009; Erkut and Alp, 2007; Erkut and Gzara, 2008; Gzara, 2013; Kara and Verter, 2004; Verter and Kara, 2008). In this application, a given set of hazardous materials is required to be transported from an origin to a destination over a road network. The problem consists of selecting road segments to be opened by the government (the leader) that aim, on the one hand, to minimize the total risk for the population. On the other, the leader assumes that the carriers (the follower) choose the shortest path in the resultant network. This is a particular case of FCNDP-SPC, where the fixed cost of each edge is equal to zero.

* Corresponding author.
E-mail addresses: ikram.bouras@lirmm.fr (I. Bouras), rosa.figueiredo@univ-avignon.fr (R. Figueiredo), michael.poss@lirmm.fr (M. Poss), fen.zhou@univ-avignon.fr (F. Zhou).

https://doi.org/10.1016/j.cor.2019.04.007
0305-0548/

Kara and Verter (2004) were the first to pose the problem as a bi-level program. Then, in (Erkut and Alp, 2007; Erkut and Gzara, 2008; Kara and Verter, 2004; Mauttone et al., 2008) the problem is transformed into a single-level mixed integer programming and the researchers focus on exact methods. More recently, Gzara (2013) studied a combinatorial bi-level formulation for the problem and proposed a cutting plane algorithm.

In this paper, we are also interested at solving the FCNDP-SPC by exact algorithms. The contributions of our work are summarized as follows:

- We propose two new BILP formulations based, respectively on path and cycle valid inequalities used to eliminate the infeasible bi-level solutions. The cycle formulation is inspired from the contribution in (Gzara, 2013). The linear relaxations of the two BILP formulations are compared theoretically.
- In order to solve the studied problem, we propose two ways of integrating the path and cycle formulations in a cutting plane method, using either a branch-and-cut or an iterative cutting-plane strategy. In the second strategy, at each iteration, a partial ILP formulation of FCNDP-SPC is solved exactly, and a set of shortest path inequalities is added while returned solution is unfeasible.
- We strengthen our formulations, through a set of valid inequalities that apply to the case where different commodities have the same origin and the same destination.
- Numerical experiments were done on real data sets from the literature as well as on random instances. The random instances generated in this work are classified into three sets according to their difficult by calculating the angle between the objective function vectors of the first and the second levels.

The paper is organized as follows. In Section 2, we start by presenting the standard bi-level formulation for FCNDP-SPC and three one-level formulations from the literature. We then propose a new path-based BILP formulation to the FCNDP-SPC. In Section 3, we compare theoretically the cycle and the path formulations and we propose a new cycle-based formulation. In Section 4, several algorithms based on the BILP formulations are proposed to solve the studied problem, and a valid inequality is added to strengthen the different formulations. Numerical results are given in Section 5. We finish this paper with a conclusion in Section 6. Detailed statistics on the algorithms are provided in an appendix.

2. Mathematical models

We consider a transportation network, which can be modeled by an undirected graph $G = (V, E)$, where V represents the set of facilities and E represents the connections between them. The connection edges are uncapacitated and undirected. Furthermore, we consider a set K of commodities to be transported over the network (these commodities may represent physical goods as raw material for industry or hazardous material). Each commodity $k \in K$, has a flow ϕ^k to be delivered through a shortest path between its origin $o(k)$ and its destination $d(k)$. Let us define the set of arcs $A = \{(i, j), (j, i) : \{i, j\} \in E\}$. A length c_{ij} and variable costs g^k_{ij}, $k \in K$, are associated to each arc $a = (i, j) \in A$. Also, for each edge $e = \{i, j\} \in E$, a fixed cost f_e is associated and we assume that $c_{ij} = c_{ji}$. The sets of all arcs leaving and arriving at node i are denoted by $\delta^+(i)$ and $\delta^-(i)$, respectively.

The FCNDP-SPC amounts to design a subnetwork (i.e., select a set of edges in E to be opened), and to find for each commodity $k \in K$ a shortest path in the resultant network such that the sum of the fixed and variable costs is minimized.

To formulate the FCNDP-SPC, two types of variables are defined. We use binary variables $y \in \{0, 1\}^{|E|}$ for the network construction such that:

$$y_e = \begin{cases} 1, & \text{if the edge } e \text{ is chosen as a part of the subnetwork,} \\ 0, & \text{otherwise.} \end{cases}$$

Besides, we use variables $x \in \{0, 1\}^{|A| \times K}$ where x^k_{ij} denotes if commodity k is sent ($x^k_{ij} = 1$) through the directed arc $a = (i, j) \in A$ or not ($x^k_{ij} = 0$). In the rest of this section, we will present five different formulations for the FCNDP-SPC problem. The first one (Section 2.1) is a bi-level integer programming model (Kara and Verter, 2004). This formulation is then transformed into two one-level models using optimality conditions of the second level problem (Sections 2.2 and 2.3). Two binary integer programming formulations (BILP) are also presented: the cycle (Section 2.4) and the proposed path (Section 2.5) based formulations.

2.1. Bi-level formulation

In the FCNDP-SPC, each commodity $k \in K$ has to be transported through a shortest path between its origin $o(k)$ and its destination $d(k)$, forcing the addition of shortest path constraints to the general problem. Besides selecting a subset of E with the minimum sum of fixed and variable costs (leader problem), we also need to guarantee that the shortest path is used for each commodity $k \in K$ (follower problem).

The FCNDP-SPC can be modeled as a bi-level mixed integer programming problem Kara and Verter (2004), as follows:

$$\min \quad \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A} \phi^k g^k_{ij} x^k_{ij} \tag{1a}$$

$$\text{s.t.} \quad y_e \in \{0, 1\}, \qquad \forall e \in E, \tag{1b}$$

$$\min \quad \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x^k_{ij} \tag{1c}$$

$$\text{s.t.} \sum_{(i,j) \in \delta^+(i)} x^k_{ij} - \sum_{(i,j) \in \delta^-(i)} x^k_{ji} = b^k_i, \qquad \forall i \in V, \forall k \in K, \tag{1d}$$

$$x^k_{ij} + x^k_{ji} \le y_e, \qquad \forall e = \{i, j\} \in E, \forall k \in K, \tag{1e}$$

$$x^k_{ij} \in \{0, 1\}, \qquad \forall (i, j) \in A, \forall k \in K \tag{1f}$$

where, for $i \in V$ and $k \in K$

$$b^k_i = \begin{cases} 1, & \text{if } i = o(k), \\ -1, & \text{if } i = d(k), \\ 0, & \text{otherwise.} \end{cases}$$

The objective functions of the first and the second levels are presented on (1a) and (1c). In (1d), we have the flow conservation constraints while constraints (1e) do not allow a flow to use arcs whose corresponding edges are closed. Finally, the constraints (1f) and (1b) require the variables x^k_{ij} and y_e to be binary. As constraints (1d) and (1e) are defined by a totally unimodular matrix, the integrality of x can be replaced by a non-negativity constraint.

Notice that, solving the follower problem is equivalent to solving $|K|$ shortest path problems independently.

2.2. One-level formulation

The FCNDP-SPC can be formulated as a one-level integer programming problem through replacing the follower problem by optimality conditions (Dempe et al., 2015; Kara and Verter, 2004). This can be done by applying the fundamental theorem of duality and complementarity slackness theorem (Bazaraa et al., 2011), as follows:

$$\min \quad \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A} \phi^k g^k_{ij} x^k_{ij} \tag{2a}$$

$$\text{s.t.} \sum_{(i,j)\in\delta^+(i)} x_{ij}^k - \sum_{(i,j)\in\delta^-(i)} x_{ji}^k = b_i^k \qquad \forall i \in V, \forall k \in K, \quad (2b)$$

$$x_{ij}^k + x_{ji}^k \leq y_e, \qquad \forall e = \{i,j\} \in E, \forall k \in K, \quad (2c)$$

$$\pi_i^k - \pi_j^k - \lambda_{e(a)}^k \leq c_a, \qquad \forall a = (i,j) \in A, \forall k \in K, \quad (2d)$$

$$(y_e - x_{ij}^k - x_{ji}^k)\lambda_e^k = 0, \qquad \forall e = \{i,j\} \in E, \forall k \in K, \quad (2e)$$

$$(c_a - \pi_i^k + \pi_j^k + \lambda_{e(a)}^k)x_{ij}^k = 0, \quad \forall a = (i,j) \in A, \forall k \in K, \quad (2f)$$

$$\lambda_e^k \geq 0, \qquad \forall e = \{i,j\} \in E, \forall k \in K, \quad (2g)$$

$$\pi_i^k \in \mathbb{R}, \qquad \forall i \in V, \forall k \in K, \quad (2h)$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall (i,j) \in A, \forall k \in K, \quad (2i)$$

$$y_e \in \{0,1\}, \qquad \forall e \in E. \quad (2j)$$

Constraints (2b), (2c) and (2i) are the follower's constraints, and (2j) are the leader constraints. Considering an arc $a = (i,j) \in A$, we define the edge associated to a by: $e(a) = \{i,j\}$. Constraints (2d)–(2f) represent the optimality conditions associated to the follower problem which ensures the shortest path requirement.

This new formulation is no more linear since constraints (2e) and (2f) contain product of variables. To bypass this problem, a big-M linearization is applied. After this modification, we can write the model as a one-level mixed integer programming problem, as follows:

$$\min \sum_{e\in E} f_e y_e + \sum_{k\in K} \sum_{(i,j)\in A} \phi^k g_{ij}^k x_{ij}^k \quad (3a)$$

$$\text{s.t. } (2b), (2c), (2d), (2j)$$

$$My_e - Mx_{ij}^k - Mx_{ji}^k + \lambda_e^k \leq M, \qquad \forall e = \{i,j\} \in E, \forall k \in K, \quad (3b)$$

$$Mx_{ij}^k - \pi_i^k + \pi_j^k + \lambda_e^k \leq M - c_a, \quad \forall a = (i,j) \in A, \forall k \in K, \quad (3c)$$

$$\lambda_e^k \geq 0, \qquad \forall e = \{i,j\} \in E, \forall k \in K, \quad (3d)$$

$$\pi_i^k \in \mathbb{R}, \qquad \forall i \in V, \forall k \in K, \quad (3e)$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall (i,j) \in A, \forall k \in K, \quad (3f)$$

Since the integrality of variables x is assumed in the linearization, the constraints (3f) are added to the formulation. The parameter M is a precomputed large number.

2.3. Bellman's model

As we have mentioned before, optimality conditions for the lower level problem are, in fact, the optimality conditions of a set of shortest path problems. Hence, the FCNDP-SPC can be expressed in a more compact way (Mauttone et al., 2008), if we consider the Bellman's optimality conditions for the shortest path problem (Ahuja et al., 1993).

$$\min \sum_{e\in E} f_e y_e + \sum_{k\in K} \sum_{(i,j)\in A} \phi^k g_{ij}^k x_{ij}^k \quad (4a)$$

$$\text{s.t. } (1d), (1e), (1b), (1f)$$

$$\pi_i^k - \pi_j^k \leq M - y_{e(a)}(M - c_a) - 2c_a x_{ji}^k, \quad \forall a = (i,j) \in A, \forall k \in K, \quad (4b)$$

$$\pi_i^k \geq 0, \qquad \forall i \in V, \forall k \in K, \quad (4c)$$

$$\pi_{d(k)}^k = 0, \qquad \forall k \in K. \quad (4d)$$

Non-negative variables π_i^k represent the shortest path distance between the node i and $d(k)$ for each commodity k. Then, $\pi_{d(k)}^k$, $k \in K$, are set to be equal to zero in Constraints (4d). Constraints (4b) are the lifted version of Belman's optimality conditions, that guarantee the shortest path requirement. As in the previous formulation, the parameter M is a precomputed large value. To improve the quality of the formulation, we want to define the smallest possible value of M (in order to strengthen the associated constraint). Let us take the constraint (4b), M can take each value with:

$$\pi_i^k - \pi_j^k \leq M, \qquad \forall a = (i,j) \in A, \forall k \in K,$$

A bound that can be used as a value for M is:

$$M = \sum_{i\in V} \left(\max_{j:(i,j)\in A} c_{ij} \right) - \min_{(i,j)\in A} c_{ij}.$$

This value of M is used in the implementation of all formulations in this paper including big-M constraints.

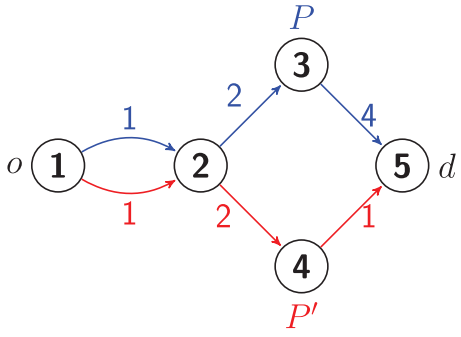


Fig. 1. A restricted graph containing two alternative paths for a commodity to be sent from 1 to 5.

2.4. BILP formulation based on cycle constraints

A one-level cycle based BILP formulation is proposed in (Gzara, 2013) for the FCNDP-SPC. Before presenting this formulation, we first introduce some notations. Let Φ be the set of pairs (x, y) satisfying the constraints of the first and the second level problems of the bi-level formulation:

$$\Phi = \{(x,y) : (1b), (1d), (1e), (1f)\}.$$

Let $\bar{y}$ be a decision of the leader and define the restricted graph $G(\bar{y}) = (V, E(\bar{y}))$ with $E(\bar{y}) = \{e \in E : \bar{y}_e = 1\}$. The feasible region of the follower, denoted by $\Phi(\bar{y})$, is given by the set of all paths on $G(\bar{y})$, i.e.,

$$\Phi(\bar{y}) = \{x : (1d), (1f), x_{ij}^k + x_{ji}^k \leq \bar{y}_e, \forall e = \{i,j\} \in E, \forall k \in K\}.$$

The followers' reaction set $\Omega(\bar{y})$ is defined as the set of shortest paths, for all commodities in K, when the leader decision is $\bar{y}$:

$$\Omega(\bar{y}) = \arg\min_{x\in\Phi(\bar{y})} \{\sum_{k\in K} \sum_{(i,j)\in A} c_{ij} x_{ij}^k\}.$$

We consider the fixed charge network design problem (FCNDP) defined as:

$$\min \sum_{e\in E} f_e y_e + \sum_{k\in K} \sum_{(i,j)\in A} \phi^k g_{ij}^k x_{ij}^k$$

$$\text{s.t. } (x,y) \in \Phi.$$

Given a feasible solution $(\bar{x}, \bar{y}) \in \Phi$ of FCNDP, if $\bar{x} \notin \Omega(\bar{y})$ then, clearly, it is not feasible for the FCNDP-SPC. As a consequence, there exists at least one commodity $k \in K$ with alternative paths P and P' from $o(k)$ to $d(k)$ in the restricted graph $G(\bar{y})$ and these alternative paths have unequal costs (wrt the lengths c_{ij}). The alternative paths P and P' form at least one cycle defined by sub-paths $p \subseteq P$ and $p' \subseteq P'$ such that $c(p) > c(p')$.

In this case, the commodity k will use the cheapest sub-path p'. The example in Fig. 1 presents the case when a commodity with

$o = 1$ and $d = 5$ has two paths in the restricted graph with different costs. The two paths form the cycle $(2-3-5-4-2)$. In this example, the commodity takes the shortest sub-path p'.

Let $\mathcal{P}(k)$ denote the set of all paths in the original graph $G = (V, E)$ for the commodity k, i.e. all paths from $o(k)$ to $d(k)$. Let $|p|$ represent the number of arcs in a given path p. Using additional binary variables z_p^k, for each $k \in K$ and for each $p \in \mathcal{P}(k)$, the FCNDP-SPC is formulated in (Gzara, 2013) as follows.

$$\min \quad \sum_{e\in E} f_e y_e + \sum_{k\in K}\sum_{(i,j)\in A} \phi^k g_{ij}^k x_{ij}^k \tag{5a}$$

$$\text{s.t.} \qquad (1b), (1d), (1e), (1f) \tag{5b}$$

$$\sum_{(i,j)\in p} x_{ij}^k \le |p| - 1 + z_p^k, \qquad \forall k \in K, \forall p \in \mathcal{P}(k), \tag{5c}$$

$$z_p^k \le x_{ij}^k, \qquad \forall k \in K, \forall p \in \mathcal{P}(k), \forall (i,j) \in p, \tag{5d}$$

$$\sum_{e\in p'} y_e \le |p'| - z_p^k, \quad \forall k \in K, \forall p, p' \in \mathcal{P}(k), \text{s.t. } c(p') < c(p) \tag{5e}$$

$$z_p^k \in \{0,1\}, \qquad \forall k \in K, \forall p \in \mathcal{P}(k). \tag{5f}$$

The binary variable z_p^k is equal to 1 if p is the path used by commodity k, and 0 otherwise. Hence, a constraint in (5c) forces z_p^k to take value 1 if $\sum_{(i,j)\in p} x_{ij}^k = |p|$. Likewise, a constraint in (5d) imposes $z_p^k = 0$ whenever there exists an arc $(i,j) \in p$ such that $x_{ij}^k = 0$. Then, constraints (5e) eliminate any solution $(\bar{x}, \bar{y})$ that has alternative sub-paths with unequal costs. Notice that this formulation has a number of constraints and variables which depend on the number of paths for each commodity $k \in K$.

2.5. BILP formulation based on path constraints

In this subsection, we propose an alternative path based formulation to avoid the additional variables z_p^k, for each $k \in K$ and each $p \in \mathcal{P}(k)$. We replace the set of constraints (5c)–(5e) by one set of constraints in charge of avoiding the commodities to use any path p whenever a path p' with $c(p') < c(p)$ is opened by the leader. The FCNDP-SPC is modeled as follows.

$$\min \quad \sum_{e\in E} f_e y_e + \sum_{k\in K}\sum_{(i,j)\in A} \phi^k g_{ij}^k x_{ij}^k \tag{6a}$$

$$\text{s.t.} \quad (1d), (1e)$$

$$\sum_{(i,j)\in A} c_{ij} x_{ij}^k \le c(p) + (|p| - \sum_{e\in p} y_e) M, \forall k \in K, \forall p \in \mathcal{P}(k), \tag{6b}$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall (i,j) \in A, \forall k \in K, \tag{6c}$$

$$y_e \in \{0,1\}, \qquad \forall e \in E. \tag{6d}$$

Parameter M is a precomputed large value (see Section 2.3). The above formulation contains a polynomial number of variables but a number of constraints that depends on the number of paths for each commodity $k \in K$.

3. Theoretical comparison and improvements

This section is devoted to compare the formulations (5a)–(5f) and (6a)–(6d) of FCNDP-SPC. As shown in the previous section, we have presented a cycle-based and a path-based formulations for the FCNDP-SPC. To compare the two BILP formulations, we will study the relation between their sets of feasible points. First, let us rewrite the set of inequalities (5c)–(5e) as a single inequality that does not involve the variable z_p^k.

Theorem 1. *For each $p, p' \in \mathcal{P}(k)$, such that $o(p) = o(p')$, $d(p) = d(p')$ and $c(p') < c(p)$, we can rewrite (5c)–(5e) as one set of inequalities:*

$$\sum_{(i,j)\in p} x_{ij}^k - |p| + 1 \le |p'| - \sum_{e\in p'} y_e, \qquad \forall p, p' \in \mathcal{P}(k), \forall k \in K. \tag{7}$$

Proof. Let $(P1)$ and $(P2)$ be the sets of points defined respectively by (5c)–(5e) and (7) associated with $p, p' \in \mathcal{P}(k)$, i.e.,

$$(P1)\begin{cases} \sum_{(i,j)\in p} x_{ij}^k \le |p| - 1 + z_p^k, & \\ z_p^k \le x_{ij}^k, & \forall (i,j) \in A, \forall k \in K, \\ \sum_{e\in p'} y_e \le |p'| - z_p^k, & \\ 0 \le x_{ij}^k \le 1, & \forall (i,j) \in A, \forall k \in K, \\ 0 \le y_e \le 1, & \forall e \in E, \forall k \in K, \\ 0 \le z_p^k \le 1. & \end{cases}$$

and

$$(P2)\begin{cases} \sum_{(i,j)\in p} x_{ij}^k - |p| + 1 \le |p'| - \sum_{e\in p'} y_e, & \\ 0 \le x_{ij}^k \le 1, & \forall (i,j) \in A, \forall k \in K, \\ 0 \le y_e \le 1, & \forall e \in E. \end{cases}$$

The inequalities are equivalent if and only if $(P1) = (P2)$. By using the Fourier–Motzkin elimination method, we can eliminate variables z_p^k from $(P1)$, and show that $(P1)$ is equivalent to the following system of inequalities:

$$\sum_{(i,j)\in p} x_{ij}^k - |p| + 1 \le |p'| - \sum_{(i,j)\in p'} y_{ij}, \tag{8a}$$

$$\sum_{(i,j)\in p} x_{ij}^k - |p| + 1 \le x_{ij}^k, \qquad \forall (i,j) \in p, \tag{8b}$$

$$0 \le x_{ij}^k \le 1, \qquad \forall (i,j) \in A, \forall k \in K, \tag{8c}$$

$$0 \le y_e \le 1, \qquad \forall e \in E. \tag{8d}$$

We can easily check that the inequality (8b) is trivial. Therefore, the set of constraints of $(P1)$ is equivalent to the set of points defining $(P2)$ and $(P2) = (P1)$. □

Based on the result of Theorem 1, we investigate if a relation between the polytopes of the linear relaxation of formulations (5) and (6) can be established.

Theorem 2. *Let us define:*

$$\mathcal{P}1 = \begin{cases} (x_{ij}^k y_e) \in [0,1], \\ x \in \Phi(y), \\ \sum_{(i,j)\in p} x_{ij}^k - |p| + 1 \le |p'| - \sum_{e\in p'} y_e, & \forall p, p' \in \mathcal{P}(k). \end{cases}$$

and

$$\mathcal{P}2 = \begin{cases} (x_{ij}^k, y_e) \in [0,1], \\ x \in \Phi(y), \\ \sum_{(i,j)\in E} c_{ij} x_{ij}^k \le C(p') + (|p'| - \sum_{(i,j)\in p'} y_{ij}) M, & \forall p' \in \mathcal{P}(k). \end{cases}$$

then we have $\mathcal{P}1 \not\subseteq \mathcal{P}2$ and $\mathcal{P}2 \not\subseteq \mathcal{P}1$.

Proof. We define an instance of the FCNDP-SPC problem with one commodity that is sent from an origin $s \in V$ to a destination node $t \in V$ via two alternative paths p and p' which form a cycle.

To show $\mathcal{P}1 \not\subseteq \mathcal{P}2$ (resp. $\mathcal{P}2 \not\subseteq \mathcal{P}1$), it is sufficient to find a fractional vector which is included in $\mathcal{P}1 \backslash \mathcal{P}2$ (resp. $\mathcal{P}2 \backslash \mathcal{P}1$).

Table 1
Number of constraints and binary variables of each formulation.

Formulations	Number of binary variables	Number of constraints
One-level formulation	$K\|A\| + \|E\|$	$K(\|V\| + 2\|A\| + 2\|E\|)$
Bellman formulation	$K\|A\| + \|E\|$	$K(\|V\| + \|A\| + \|E\|)$
Cycle-based formulation	$K(\|A\| + \|\mathcal{P}(k)\|) + \|E\|$	$K(\|V\| + \|E\| + 3\|\mathcal{P}(k)\|)$
Path-based formulation	$K\|A\| + \|E\|$	$K(\|V\| + \|E\| + \|\mathcal{P}(k)\|)$
New cycle-based formulation	$K\|A\| + \|E\|$	$K(\|V\| + \|E\| + \|\mathcal{P}(k)\|)$

- We define the fractional vector:

  $y^*_e = 1 - \epsilon \quad \forall e \in E$
  $x^*_{ij} = 1 - \epsilon \quad \forall (i,j) \in p$
  $x^*_{ij} = \epsilon \quad \forall (i,j) \in p'$
  $|p| = 2 |p'| = 2$
  $c(p') = c' \leq c = c(p)$

  The path constraint (6b) for this vector is:

  $(1-\epsilon)c + \epsilon c' \leq c' + \epsilon M$

  Hence, the constraint is satisfied for many values of ϵ, M (we can take for instance: $\epsilon = \frac{1}{4}$ and $M = 4c$). However, the cycle constraint in $\mathcal{P}1$ for this vector becomes:

  $$\frac{10}{4} \leq 2$$

  then, $(y*, x*) \notin \mathcal{P}1$.
  Hence, $(x^*, y^*) \in \mathcal{P}2 \backslash \mathcal{P}1$.

- We define the fractional vector:

  $y^*_e = 1 - \epsilon \quad \forall e \in E$
  $x^*_{ij} = 1 - \epsilon \quad \forall (i,j) \in p$
  $x^*_{ij} = \epsilon \quad \forall (i,j) \in p'$
  $|p'| = 1$
  $c(p') = c' = 1$
  $M = c(p) = c = 4$

  We can easily check that $(x^*, y^*) \in \mathcal{P}1$, for all $\epsilon > \frac{1}{1+|p|}$. Furthermore, choosing $\epsilon = \frac{2}{5}$ and $|p| = 4$, the path constraint cannot be satisfied.
  Hence, we obtain: $(x^*, y^*) \in \mathcal{P}1 \backslash \mathcal{P}2$.

□

Theorem 2 proves that the two BILP formulations (5) and (6) are not comparable.

In Theorem 1, we proved that the inequalities (7) can eliminate any path violating the shortest path requirement, thus, the FCNDP-SPC can be formulated as a new cycle-based formulation:

$$\min \quad \sum_{e \in E} f_e y_e + \sum_{k \in K} \sum_{(i,j) \in A} \phi^k g^k_{ij} x^k_{ij}, \tag{9a}$$

$$\text{s.t.} \quad \sum_{(i,j) \in \delta^+(i)} x^k_{ij} - \sum_{(i,j) \in \delta^-(i)} x^k_{ji} = b^k_i, \qquad \forall i \in V, \forall k \in K, \tag{9b}$$

$$x^k_{ij} + x^k_{ji} \leq y_e, \qquad \forall e \in E, \forall k \in K, \tag{9c}$$

$$\sum_{(i,j) \in p} x^k_{ij} + \sum_{e \in p'} y_e \leq |p| + |p'| - 1,$$

$$\forall k \in K, \forall p, p' \in \mathcal{P}(k) : c(p') < c(p), \tag{9d}$$

$$x^k_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A, \forall k \in K, \tag{9e}$$

$$y_e \in \{0,1\}, \qquad \forall e \in E. \tag{9f}$$

4. Proposed algorithms for FCNDP-SPC

In Sections 2 and 3, we presented five different formulations to this problem. Table 1 compares them according to the number of variables and constraints.

Formulations (3) and (4) are compact ones. On the other hand, formulations (5), (6) and (9) may have exponential numbers of constraints. Formulation (5) may also have an exponential number of variables.

In this section, we focus on presenting different exact methods to solve the FCNDP-SPC.

4.1. Compact formulations

The first way to solve the FCNDP-SPC is to feed the models (3) and (4) to Gurobi solver with default parameters. The two formulations are solved by branch-and-bound algorithms, let **B&B1** and **B&B2** denote the branch-and-bound algorithms for the formulations (3) and (4), respectively.

4.2. Iterative cutting plane algorithms

Our iterative cutting plane algorithms are based on formulations (5),(6) and (9). Algorithm 1 gives the pseudo-code of these algorithms.

Algorithm 1 Cutting plane algorithms.

Step 0: $(BILP)^0$ is the fixed charge network design problem defined by the set of constraints (1b),(1d),(1e), and (1f).$l = 0$.
Step 1: Solve $(BILP)^l$ to obtain the solution (x^l, y^l). Let N^l be the network induced by y^l, and $(P^k)^l$ be the obtained path for the commodity $k \in K$. We denote by $c^l(P^k)$ the total cost of $(P^k)^l$.
Step 2: On N^l, find $(P'^k)^l$ the shortest path for each commodity k with cost $c^l(P'^k)$.
Step 3: If $c^l(P^k) = c^l(P'^k)$ for each commodity k, then N^l defines an optimal solution for FCNDP-SPC, Stop.
Step 4: For the commodities with $c^l(P^k) \neq c^l(P'^k)$, generate a set of shortest path constraints to eliminate the path $(P^k)^l$. Let S be the set of constraints generated.Append the constraints generated S to $(BILP)^{l+1}$.Do $l = l + 1$, and go to step 1.

The set S of inequalities in **Step 4** can be generated in three different ways giving us three different iterative cutting plane methods **CP1, CP2**, and **CP3**:

- **CP1**: Generate the inequality (6b).
- **CP2**: Find the cycles formed by $(P'^k)^l$ and $(P^k)^l$. Generate the set of valid inequalities (5c)–(5e).
- **CP3**: Find the cycles formed by $(P'^k)^l$ and $(P^k)^l$. Generate the inequality (7).

4.3. Branch-and-cut algorithm

The FCNDP-SPC can also be solved to optimality by using branch-and-cut algorithms based on formulations (6) and (9).

The principle of the algorithm used here is to solve a fixed charge network design problem (without the shortest path constraints) by a branch-and-bound procedure and to add valid inequalities to each integral node violating shortest path constraints. The algorithm stops when there is no more node to evaluate, i.e.,

all the need path constraints were generated. The difference of this algorithm over the iterative cutting plane algorithms lies in solving a unique mixed integer problem by considering all the shortest path constraints. Next, we detail each component of the Branch-and-cut procedure.

The initial model: The initial model consists in minimizing the sum of the fixed and variable costs under the flow conservation constraints (1d), the constraints (1e) forcing the flow to use only the opened edges and the binary requirement constraints of x (1f) and of y (1b).

Separation problem: Since the initial problem does not contain all the constraints of the FCNDP-SPC, an integer solution obtained can be infeasible. For this reason, for each integer node on the branch-and-bound tree, we introduce a cut generation procedure to eliminate the infeasible paths.

We first check if an integer solution $(\bar{x}, \bar{y})$ is feasible for FCNDP-SPC by solving a set of shortest path problems. For each commodity k, in the subnetwork defined by $\bar{y}$ we verify if the path P' defined by $\bar{x}$ is the shortest path from $o(k)$ to $d(k)$ in $\bar{y}$. If $(\bar{x}, \bar{y})$ is infeasible, i.e., there exists at least one path P with $c(P) < c(P')$, then a set of shortest path constraints is added. Two variants of the algorithm were developed. In **B&C1** shortest path constraints correspond to (6b) while for **B&C2** they correspond to (7).

4.4. Valid inequalities

The different formulations of the FCNDP-SPC can be weak. One way to strengthen the models is to introduce a set of valid inequalities. From the definition of the problem, we can remark that, if there exist several different commodities with the same origin and destination, then their paths in the optimal solution have the same cost.

The following proposition shows that a valid inequality can be generated in this particular case.

Proposition 1. *Consider two commodities k^1, k^2 such that $o(k^1) = o(k^2)$ and $d(k^1) = d(k^2)$, then the constraint:*

$$\sum_{(i,j)\in A} c_{ij} x_{ij}^{k^1} = \sum_{(i,j)\in A} c_{ij} x_{ij}^{k^2} \tag{10}$$

is valid for FCNDP − SPC.

Proof. Straightforward. □

5. Numerical results

In this section, we present computational experiments carried out with all the methods described in the previous sections. All algorithms are implemented in Julia 0.5.0, and the problems are solved using Gurobi 6.5.2 (with four threads). Simulations were performed on an Intel(R)core TM i7-3520M CPU@2.90 GHz × 4 computer with 8 GB of RAM. Numerical experiments were performed on two sets of data. The first one concerns 405 random instances generated for this work, while the second one consists of real instances from different city transportation networks (Ravenna, Italy Bonvicini and Spadoni, 2008, and Albany, NY,USA Toumazis et al., 2013). Next, we describe each data set and discuss the computational results obtained on each one.

We summarize all the exact algorithms used to solve the FCNDP-SPC:

B&B1 : Branch-and-bound algorithm for the one-level formulation (3).
B&B2 : Branch-and-bound algorithm for the Bellman model (4).
CP1 : Cutting plane algorithm using the inequalities (6b).
CP2 : Cutting plane algorithm using the inequalities (5c)–(5e).
CP3 : Cutting plane algorithm using the inequalities (7).
B&C1 : Branch-and-cut algorithm using inequalities (6b) in the cut generation.
B&C2 : Branch-and-cut algorithm using inequalities (7) in the cut generation.

5.1. Random instances

The different methods are tested on 405 instances generated randomly with different values for the angles α between the variable cost vector "g" and the length vector associated to the edges "c" in the network $G(V, E)$. We consider three different categories for the value of α:

- $0^\circ \le \alpha \le 10^\circ$
- $40^\circ \le \alpha \le 50^\circ$
- $80^\circ \le \alpha \le 90^\circ$

The purpose of distinguishing these three scenarios is to examine whether the difficulty of the instance is related to the angle between g and c, i.e., the direction of the objective function of the upper and the lower levels. For instance, we can expect that if both levels go in the same direction, i.e., the two vectors are very close to each other, the instance is easy.

The instances are generated varying also the number of nodes in the graph $n \in \{10, 20, 30\}$, the graph density $d \in \{0.3, 0.5, 0.7\}$, and the number of different commodities to be transported $K \in \{\frac{n}{2}, n, \frac{3n}{2}\}$. For each combination of (n, d, K), 5 instances are generated.

Similar random instances have also been generated for FCNDP-SPC by Mauttone et al. (2008) and used in González et al. (2016). Their instances were not used here for two reasons. First, these instances belong to a particular case of the problem where all the variable costs are equal for all the commodities. Second, all instances are included in the first scenario of α, where α is close to zero. Hence, the random instances of Mauttone et al. (2008) are considered very easy to solve.

The obtained results are shown in Tables 2–4, where, we report in each row the average CPU time in seconds spent by each algorithm on each group of 5 instances with the same (n, d, k). The symbol "-" means that the algorithm was not able to find the optimal solution in the time limit of 3600 s for all the considered instances, also, an additional number (.) is added to represent the number of instances solved in each group of 5 instances. In addition, the average CPU time and the total number of solved instances are given for each number of nodes n. Overall, the iterative cutting plane algorithms **CP1** and **CP3** outperform the other algorithms and they solve more instances. Furthermore, algorithms **B&C1** and **B&C2** are more efficient than **B&B1** and **B&B2**, although the difference is less marked. We can also see from Tables 2–4 that the instances with $0^\circ \le \alpha \le 10^\circ$ are easy to solve than the other instances.

We provide next more detailed statistics to compare the different formulations and algorithms. These are reported in Fig. 2, while the full details are reported to Tables A.9– A.12 of Appendix A.

In order to compare the different one-level formulations presented in this work, in Fig. 2a, we compare the average of the gap between the solution of the linear relaxation and the optimal solution. We can remark that the linear relaxations of the BILP formulations (5) and (9) are stronger than those of formulations (3) and (4). For instance, in the first case of α the gap of (5) (resp. (9)) is twice (resp. four times) smaller than the gap of the formulations (3) and (4). Although the formulation (9) has smaller gap for our random instances, there are instances where the linear relaxation of the formulation (5) is stronger (see Theorem 2).

The computational experiments show that, overall, the iterative cutting plane algorithm **CP3** is faster than the branch-and-cut

Table 2
CPU time in seconds for instances with $0^\circ \leq \alpha \leq 10^\circ$.

n - d - K	B&B1	B&B2	CP1	CP2	CP3	B&C1	B&C2
10-0.3-5	0.04	0.01	0.01	0.01	0.01	0.08	0.13
10-0.3-10	0.09	0.04	0.02	0.02	0.02	0.06	0.13
10-0.3-15	0.26	0.06	0.05	0.07	0.05	0.20	0.28
10-0.5-5	0.07	0.03	0.03	0.05	0.04	0.04	0.09
10-0.5-10	0.43	0.12	0.14	0.25	0.19	0.22	0.35
10-0.5-15	0.82	0.45	0.19	0.30	0.21	0.45	0.94
10-0.7-5	0.04	0.02	0.01	0.01	0.01	0.05	0.05
10-0.7-10	0.47	0.25	0.08	0.20	0.14	0.31	0.40
10-0.7-15	0.63	0.20	0.13	0.16	0.15	0.30	0.49
10-Average	0.32(45)	0.13(45)	0.07(45)	0.12(45)	0.09(45)	0.19(45)	0.32(45)
20-0.3-10	0.86	0.21	0.10	0.09	0.07	0.41	0.47
20-0.3-20	284.73	8.80	4.13	12.15	28.61	3.07	10.27
20-0.3-30	655.37	61.48	9.07	54.10	45.68	8.99	34.63
20-0.5-10	1.49	0.19	0.17	0.19	0.14	0.40	0.48
20-0.5-20	257.22	10.82	4.93	14.70	11.75	5.20	7.13
20-0.5-30	379.72 (2)	1095.67(4)	349.24	455.11	453.08	440.30	633.17(4)
20-0.7-10	0.93	0.20	0.12	0.14	0.11	0.33	0.40
20-0.7-20	308.80	56.19	23.96	80.47	65.89	13.70	23.79
20-0.7-30	622.18(1)	912.28	241.56	547.02	542.20	139.11	694.23
20-Average	279.03(38)	238.43(44)	70.36(45)	129.33(45)	127.50(45)	67.95(45)	156.06(44)
30-0.3-15	38.26	1.07	0.49	0.66	0.65	1.31	1.32
30-0.3-30	1595.92(2)	273.98	65.70	179.70	152.62	54.21	67.33
30-0.5-15	22.78	1.18	0.55	0.65	0.64	2.32	1.55
30-0.5-30	–	–	862.61	761.85	729.97	–	–
30-0.7-15	132.37	5.65	4.26	4.77	4.39	8.22	3.64
30-0.7-30	–	–	1095.17	1074.14	986.96	–	–
30-Average	447.33(17)	70.47(20)	338.13(30)	336.96(30)	312.54(30)	16.51(20)	18.46(20)

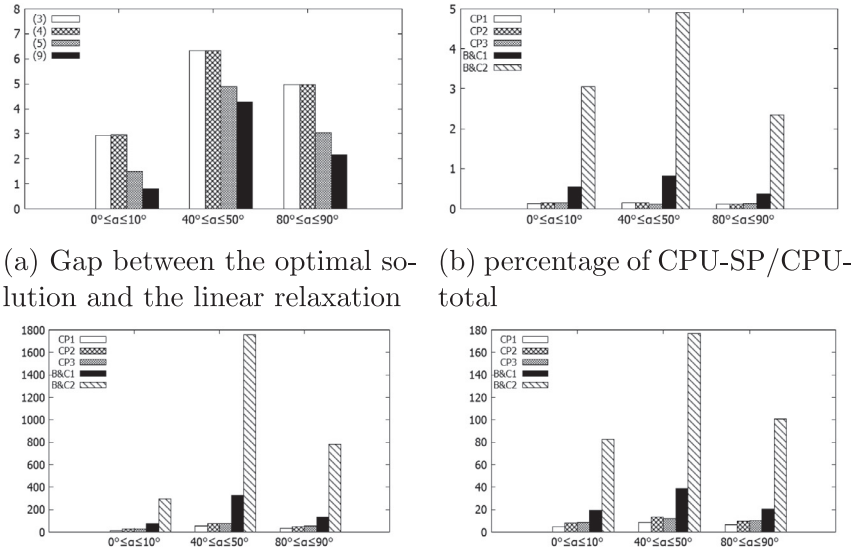
Table 3
CPU time in seconds for instances with $40^\circ \leq \alpha \leq 50^\circ$.

n - d - K	B&B1	B&B2	CP1	CP2	CP3	B&C1	B&C2
10-0.3-5	0.04	0.02	0.01	0.01	0.01	0.04	0.07
10-0.3-10	0.16	0.05	0.05	0.05	0.04	0.17	0.19
10-0.3-15	0.28	0.09	0.10	0.14	0.10	0.30	0.44
10-0.5-5	0.02	0.08	0.04	0.02	0.01	0.05	0.08
10-0.5-10	0.24	0.44	0.35	0.39	0.19	0.31	0.47
10-0.5-15	0.77	6.60	1.34	1.91	1.01	0.90	1.11
10-0.7-5	0.02	0.04	0.01	0.02	0.01	0.06	0.06
10-0.7-10	0.47	7.86	1.17	3.51	1.43	0.86	0.92
10-0.7-15	4.30	256.09	9.03	29.70	11.85	3.87	12.62
10-Average	0.70(45)	30.14(45)	1.35(45)	3.97(45)	1.63(45)	0.73(45)	1.77(45)
20-0.3-10	1.65	0.25	0.16	0.20	0.15	0.53	0.52
20-0.3-20	1226.54	27.66	43.71	117.58	94.21	16.77	35.32
20-0.3-30	–	1427.53(4)	827.44	696.42	648.77	927.19	1147.03(3)
20-0.5-10	0.26	4.09	0.82	0.28	0.20	1.00	0.51
20-0.5-20	271.99(4)	308.38(2)	659.96(4)	329.23(4)	132.99	1182.37	841.59
20-0.5-30	–	3410.90	614.23	500.25	776.10	–	–
20-0.7-10	0.42	2.74	0.91	0.28	0.19	0.73	0.58
20-0.7-20	185.35(4)	648.43(2)	334.97	224.35	210.19	881.23(4)	386.98(4)
20-0.7-30	–	–	1196.37	1030.52	910.30	–	–
20-Average	281.04(28)	728.75(33)	408.73(44)	322.12(44)	308.12(45)	429.97(34)	344.65(32)
30-0.3-15	19.79	856.09	4.08	6.55	6.26	15.32	8.17
30-0.3-30	–	–	–	–	744.85	–	–
30-0.5-15	4.07	192.25	1.54	1.45	1.47	5.95	2.55
30-0.5-30	–	–	–	–	785.04	–	–
30-0.7-15	27.63	1171.09	5.82	12.69	11.34	24.51	12.80
30-0.7-30	–	–	–	–	674.06	–	–
30-Average	17.16(15)	739.81(15)	3.81(15)	6.90(15)	370.50(30)	15.26(15)	7.84(15)

algorithms. One of the reasons that explain those results is the time consumed on the separation problem in each algorithm. To study the effect of the separation problem on the results, we compute the percentage of the run time consumed by the separation problem on the CPU-total in Fig. 2b. This percentage is similar and negligible for the iterative cutting plane algorithms unlike the branch-and-cut algorithms where the solution of separation problem takes for many instances more than 20% (see Fig. 2d) of the total time consumed by the **B&C2** algorithm. To complement these results we compare the number of iterations and the number of cuts generated by each algorithm. As we can see in Fig. 2c, the **CP1** algorithm generates a smaller number of cuts to obtain the optimal solution than the other iterative cutting plane and branch-and-cut algorithms. Specifically, **B&C1** (resp. **B&C2**) generates more than six (resp. twenty) times the number of cuts of **CP1**. In addition, we can remark that the similar performances of **CP1** and **CP2** can be partly explained by the equivalence between the two valid inequalities used in these algorithms. The same remark can be shown for the number of iterations in Fig. 2d.

Table 4
CPU time in seconds for instances with $80^\circ \leq \alpha \leq 90^\circ$.

n - *d* - *K*	B&B1	B&B2	CP1	CP2	CP3	B&C1	B&C2
10-0.3-5	0.03	0.01	0.01	0.01	0.01	0.02	0.06
10-0.3-10	0.17	0.05	0.06	0.05	0.03	0.14	0.22
10-0.3-15	0.40	0.12	0.11	0.10	0.07	0.26	0.36
10-0.5-5	0.02	0.01	0.01	0.01	0.01	0.02	0.05
10-0.5-10	0.46	0.12	0.14	0.27	0.18	0.27	0.29
10-0.5-15	3.18	0.37	0.85	1.46	1.08	0.52	1.29
10-0.7-5	0.07	0.02	0.02	0.03	0.02	0.05	0.73
10-0.7-10	0.87	0.16	0.22	0.35	0.22	0.30	3.61
10-0.7-15	108.61	1.96	2.11	9.41	5.38	0.80	89.82
10-Average	12.65(45)	0.31(45)	0.39(45)	1.30(45)	0.78(45)	0.27(45)	10.71(45)
20-0.3-10	1.35	0.15	0.17	0.15	0.12	0.33	0.51
20-0.3-20	837.97(4)	27.22	31.18	85.09	77.47	70.48	20.97
20-0.3-30	–	923.06	279.67	292.00	269.23	72.24(4)	91.21
20-0.5-10	3.21	0.55	0.51	0.84	0.71	0.72	0.76
20-0.5-20	1161.59	41.18	22.61	50.50	39.08	18.35	33.82
20-0.5-30	33.02(4)	264.56(2)	943.67	655.29	644.81	1043.20(4)	583.20(4)
20-0.7-10	1.15	0.30	0.12	0.11	0.09	0.37	7.32
20-0.7-20	329.74(3)	548.91	209.45	130.58	131.29	120.45	2.59(4)
20-0.7-30	–	–	1444.21	885.10	893.73	1206.02(4)	2508.14(4)
20-Average	338.29(31)	225.74(37)	325.73(45)	233.30(45)	228.50(45)	281.89(42)	360.94(42)
30-0.3-15	87.28	4.07	1.79	2.46	2.19	4.43	4.99
30-0.3-30	–	–	–	–	771.18	–	–
30-0.5-15	159.66	1.59	1.16	1.07	1.05	2.41	1.51
30-0.5-30	–	–	–	–	737.51	–	–
30-0.7-15	9.62	1.00	0.50	0.51	0.46	1.39	1.10
30-0.7-30	–	–	–	–	688.11	–	–
30-Average	85.52(15)	2.22(15)	1.15(15)	1.35(15)	366.75(30)	2.74(15)	2.53(15)



(a) Gap between the optimal solution and the linear relaxation

(b) percentage of CPU-SP/CPU-total

(c) Number of cuts generated by **CP** and **B&C** algorithms

(d) Number of separation problems solved

Fig. 2. Additional statistics for the algorithms and formulations.

Table 5
Number of random instances solved to optimality.

	B&B1	B&B2	CP1	CP2	CP3	B&C1	B&C2
$0^\circ \leq \alpha \leq 10^\circ$	100	109	120	120	120	110	109
$40^\circ \leq \alpha \leq 50^\circ$	88	89	104	104	120	94	92
$80^\circ \leq \alpha \leq 90^\circ$	88	97	105	105	120	102	102
Sum	276	295	329	329	360	306	303

To evaluate the performance of the different formulations, a performance profile (Dolan and Moré, 2002) of solution time on the random instances is given in Fig. 3. The chart represents the proportion of instances for which each algorithm is not more than x-times worst than the best algorithm. For example, if we take $x = 1$, we can see that **CP3** is the best algorithm for 40% of instances, while **CP1** has the best CPU-time for more than 20% of instances.

Table 5 sums up the number of instances solved to optimality in 3600*s*. The iterative cutting plane approaches **CP1** and **CP2** are similar to each other. They can solve up to 19% (resp. 8%) more instances than the branch-and-bound algorithms (resp. the branch-and-cut algorithms). Also, **CP3** solves more instances than the other methods.

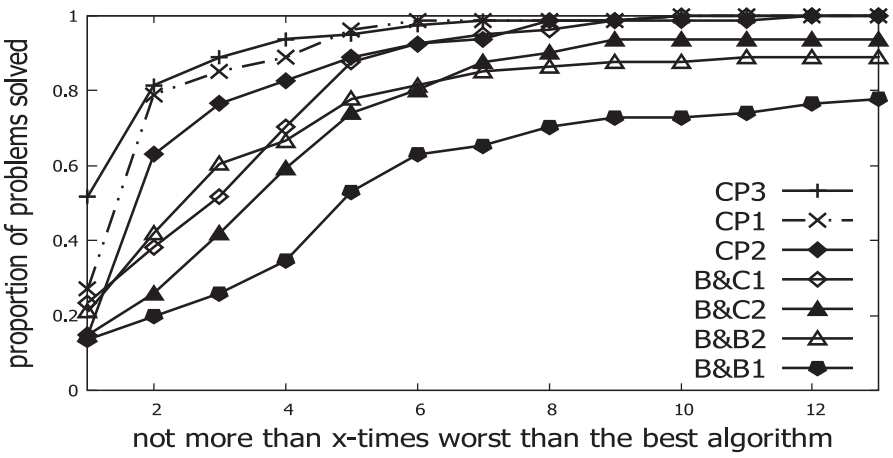


Fig. 3. Performance profile comparing the different algorithms for random instances.

5.2. Real instances

We apply the proposed algorithms to solve FCNDP-SPC on real instances from the literature representing the road networks of Ravenna (Italy) and Albany, NY (USA). In these instances, the fixed costs are equal to zero.

5.2.1. Ravenna data

The area of Ravenna, Italy, measures $28.8\,\text{km} \times 26\,\text{km}$ Bonvicini and Spadoni (2008). To describe its road network, 111 nodes and 143 edges are considered. There are 8 nodes with a transportation requirement for 4 hazmats: LPG, Methanol, Gasoline, and Chlorine. The 8 nodes form 35 origin-destination (O-D) pairs, and the number of commodities to be transported between each O-D varies between 16 and 29684. The variable cost associated to each edge is measured using population density. Seven instances are taken from the original Ravenna data with different number of commodities $K \in \{5, 10, 15, 20, 25, 30, 35\}$, where, in each case we take the first K pairs of origin-destination. The results obtained after applying the six models on these instances are displayed in Table 6.

The presented results show that the iterative cutting plane algorithms are faster, in average, than the other algorithms and are able to solve more instances to optimality. Also, our approach **CP1** is more efficient for this set of instances.

Unlike the other instances, in Ravenna data, many commodities have the same origin and the same destination. For this reason, the inclusion of the valid inequality (10) as a constraint in the initial model can improve the results. Table 7 presents the CPU time obtained after adding the valid inequality.

According to the results in Table 7, the addition of valid inequality (10) reduces the running time for all algorithms. Comparing Tables 6 and 7, we see that the branch-and-cut algorithms solved more instances to optimality when we include inequalities (10). Also, the results show that the CPU times for all the algorithms with the constraint (10) is about 2.25 times faster than the CPU of algorithms without it.

Table 7
CPU time in seconds on the Ravenna data with additional valid inequality.

K	B&B1	B&B2	CP1	CP2	CP3	B&C1	B&C2
5	41.91	0.45	1.75	1.66	1.53	1.59	1.57
10	160.03	1.10	3.49	3.61	3.09	2.32	2.38
15	585.17	2.43	5.94	6.40	5.34	3.66	3.92
20	–	8.98	70.90	42.98	35.28	205.94	35.47
25	–	151.27	63.50	55.25	43.95	105.33	355.98
30	–	371.25	57.83	105.59	66.78	156.80	290.37
35	–	–	84.20	174.00	119.83	229.43	414.51
Average	–	–	36.63	55.64	39.40	100.72	157.74

Table 8
Comparison of CPU time on the Albany data.

K	B&B1	B&B2	CP1	CP2	CP3	B&C1	B&C2
5	83.97	0.65	23.43	37.06	20.13	5.65	2.71
10	–	919.53	87.01	–	996.03	332.79	–
15	–	–	193.51	–	–	–	–
20	–	–	255.40	–	–	–	–
25	–	–	652.84	–	–	–	–

5.2.2. Albany data

The Albany data set is composed of information on the highway system of Albany, NY, USA, used for the routing of Hazmat shipments problem (Toumazis et al., 2013). The highway system is represented by a network of 90 nodes and 149 edges.

Table 6
Comparison of CPU time in seconds on the Ravenna data.

K	B&B1	B&B2	CP1	CP2	CP3	B&C1	B&C2
5	41.91	0.45	1.59	1.52	1.52	2.32	2.45
10	145.90	0.98	3.15	3.09	2.98	3.29	4.68
15	1,169.96	3.80	33.01	23.13	27.88	50.75	96.00
20	–	8.98	51.61	47.93	46.21	78.13	2,896.65
25	–	259.91	82.74	86.72	63.15	–	–
30	–	1,494.56	151.28	283.88	265.86	–	–
35	–	–	222.11	451.68	351.18	–	–
Average	–	–	77.93	128.28	108.40	–	–

We generated a set of origin-destination pairs for each shipment. The number of commodities takes a value in the set $K = \{5, 10, 15, 20, 25\}$, and the demand for each O-D is generated uniformly in [1,100]. Table 8 displays the results obtained by each method.

According to the results in Table 8, only **CP1** is able to find the optimal solution for up to 25 different commodities.

6. Conclusion

In this paper, we proposed two new BILP formulations for the fixed charge network design problem with shortest path constraints. The two models are combined with the iterative cutting plane (**CP1, CP3**) and a branch-and-cut (**B&C1, B&C2**) algorithms. We further provide a valid inequality for the formulation of FCNDP-SPC whenever there are commodities sharing the same origin and the same destination exist.

All algorithms are tested on two types of data. We generated a set of 405 random instances classified in three groups according to their difficulty. The results show that the FCNDP-SPC is much easier when $0^\circ \leq \alpha \leq 10^\circ$ and it becomes very difficult for $40^\circ \leq \alpha \leq 50^\circ$. In these instances, the iterative cutting plane algorithms are almost equivalent and more efficient than the branch-and-bound and the branch-and-cut algorithms. This somewhat surprising result can be partly explained by the high number of inequalities generated by the branch-and-cut algorithms and the time consumed by the separation problem.

We also used instances from the literature. Two real instances Ravenna (Italy) (Bonvicini and Spadoni, 2008) and Albany, NY, (USA) (Toumazis et al., 2013) used to test the different algorithms. The obtained results show the time efficiency of our cutting plane algorithm (**CP1**) in comparison with the other algorithms.

From the experiment results, we can make two major observations. First, branch-and-cut and cutting plane approaches are better than the branch-and-bound algorithms for all types of instances. We saw that the iterative cutting plane algorithms outperform the branch-and-cut algorithm in terms of CPU time. This is because the latter consumes more time in the cuts generation and the separation problems in each integer node of the branching tree. Also, our **CP1** cutting plane method has the best results for all real instances.

Second, the valid inequality generated for the Ravenna data improves the running time for all algorithms.

As future work, we intend to study exact approaches for a variation of the FCNDP-SPC: the capacitated fixed charge network design problem with user-optimal flow. We are interested in studying this variation because it frequently appears in telecommunication networks.

Appendix A

Table A.9
Gap between the optimal solution and the linear relaxation.

Angle	$0^\circ \leq \alpha \leq 10^\circ$				$40^\circ \leq \alpha \leq 50^\circ$				$80^\circ \leq \alpha \leq 90^\circ$			
Formulation	(3)	(4)	(5)	(9)	(3)	(4)	(5)	(9)	(3)	(4)	(5)	(9)
10-3-5	0.00	0.00	0.00	0.00	1.66	1.60	0.43	0.00	0.00	0.00	0.00	0.00
10-3-10	0.86	0.87	0.54	0.00	1.88	1.83	0.00	0.00	3.42	3.41	2.13	0.00
10-3-15	1.91	1.96	0.17	0.00	5.78	5.68	1.02	0.00	4.22	4.20	1.16	0.00
10-5-5	2.68	2.68	2.52	0.00	0.75	0.75	0.76	0.76	0.00	0.00	0.00	0.00
10-0.5-10	2.39	2.40	1.28	0.90	6.26	6.27	6.30	6.30	5.52	5.41	1.65	0.96
10-0.5-15	4.68	4.70	1.14	0.42	7.99	8.00	8.01	8.01	10.43	10.37	6.07	3.66
10.7.5	0.42	0.42	0.00	0.00	0.46	0.46	0.00	0.00	1.57	1.57	0.68	0.00
10.7.10	8.15	8.16	2.54	2.42	9.46	9.46	4.77	3.50	4.50	4.38	1.81	1.34
10.7.15	2.68	2.68	0.56	0.00	16.88	16.89	11.78	9.29	12.33	12.44	5.98	5.10
20-0.3-10	0.83	0.83	0.24	0.00	1.16	1.15	0.37	0.00	2.38	2.38	0.73	0.00
20-0.3-20	7.54	7.54	5.17	3.19	10.06	10.04	7.60	6.46	9.17	9.15	7.71	6.57
20-0.3-30	6.96	6.97	4.17	2.64	13.67	13.65	11.46	10.60	8.97	8.97	7.19	6.31
20-0.5-10	0.50	0.50	0.28	0.00	2.31	2.31	2.33	2.33	3.66	3.65	1.69	1.22
20-0.5-20	6.44	6.44	3.44	2.02	11.94	11.99	12.04	12.04	5.56	5.57	4.23	3.02
20-0.5-30	0.08	0.08	0.06	0.06	11.86	11.83	11.89	11.89	10.28	10.29	8.36	6.79
20.7.10	0.73	0.73	0.00	0.00	2.73	2.74	1.35	0.00	1.02	1.02	0.00	0.00
20.7.20	6.49	6.51	4.29	3.72	8.98	8.99	7.21	6.13	6.31	6.32	4.41	3.40
20.7.30	0.10	0.10	0.07	0.06	9.97	9.98	8.00	7.22	8.82	8.82	7.45	6.22
30-0.3-15	2.24	2.24	1.60	0.00	3.18	3.18	2.61	1.53	2.41	2.41	1.36	0.74
30-0.5-15	1.83	1.84	0.53	0.00	2.11	2.11	2.12	2.12	2.65	2.65	0.96	0.00
30.7.15	4.32	4.32	2.63	1.27	4.11	4.12	2.81	1.11	1.20	1.20	0.32	0.00
Average	2.94	2.95	1.49	0.79	6.34	6.33	4.90	4.25	4.97	4.96	3.04	2.16

Table A.10
% of CPU-SP/CPU-total.

Angle	$0^\circ \leq \alpha \leq 10^\circ$					$40^\circ \leq \alpha \leq 50^\circ$					$80^\circ \leq \alpha \leq 90^\circ$				
n - d - K	CP1	CP2	CP3	B&C1	B&C2	CP1	CP2	CP3	B&C1	B&C2	CP1	CP2	CP3	B&C1	B&C2
10-0.3-5	0.28	0.27	0.27	0.00	0.01	0.24	0.25	0.22	0.01	0.01	0.28	0.26	0.26	0.00	0.01
10-0.3-10	0.30	0.31	0.31	0.01	0.02	0.25	0.28	0.29	0.03	0.03	0.27	0.28	0.30	0.02	0.03
10-0.3-15	0.32	0.32	0.30	0.03	0.04	0.25	0.25	0.27	0.05	0.07	0.23	0.28	0.29	0.04	0.05
10-0.5-5	0.22	0.22	0.23	0.01	0.01	0.45	0.31	0.26	0.01	0.01	0.29	0.28	0.27	0.00	0.01
10-0.5-10	0.23	0.24	0.25	0.04	0.05	0.29	0.38	0.21	0.04	0.06	0.18	0.22	0.23	0.04	0.04
10-0.5-15	0.19	0.22	0.24	0.06	0.16	0.26	0.19	0.13	0.12	0.13	0.09	0.08	0.08	0.09	0.19
10-0.7-5	0.26	0.27	0.27	0.01	0.01	0.41	0.26	0.28	0.01	0.01	0.23	0.22	0.19	0.01	0.01
10-0.7-10	0.20	0.20	0.19	0.04	0.06	0.20	0.26	0.14	0.09	0.10	0.16	0.17	0.19	0.04	0.08
10-0.7-15	0.25	0.24	0.23	0.04	0.07	0.04	0.02	0.02	0.28	0.53	0.06	0.05	0.06	0.12	0.45
20-0.3-10	0.19	0.21	0.22	0.04	0.04	0.18	0.19	0.21	0.07	0.05	0.17	0.18	0.20	0.04	0.05
20-0.3-20	0.07	0.09	0.07	0.40	1.27	0.02	0.03	0.03	0.82	3.00	0.01	0.01	0.01	0.85	1.80
20-0.3-30	0.03	0.03	0.03	0.73	4.06	0.00	0.00	0.00	2.86	32.67	0.00	0.00	0.00	1.36	3.82
20-0.5-10	0.16	0.16	0.18	0.04	0.05	0.20	0.26	0.17	0.09	0.06	0.10	0.11	0.12	0.07	0.08
20-0.5-20	0.06	0.07	0.08	0.36	0.41	0.01	0.01	0.01	1.78	8.97	0.03	0.03	0.04	0.46	2.42
20-0.5-30	0.01	0.01	0.01	1.21	7.91	0.00	0.00	0.00	2.01	12.47	0.00	0.00	0.00	1.71	21.05
20-0.7-10	0.14	0.15	0.16	0.03	0.04	0.11	0.25	0.14	0.05	0.07	0.11	0.12	0.13	0.04	0.04
20-0.7-20	0.04	0.05	0.05	0.51	1.59	0.02	0.04	0.02	2.94	15.14	0.02	0.03	0.03	0.49	2.28
20-0.7-30	0.00	0.00	0.00	1.58	5.53	0.00	0.00	0.00	4.16	28.09	0.00	0.00	0.00	1.70	15.99
30-0.3-15	0.12	0.12	0.12	0.14	0.15	0.05	0.05	0.05	0.81	0.64	0.09	0.10	0.11	0.34	0.58
30-0.3-30	0.02	0.02	0.02	1.91	3.93	–	–	–	–	–	–	–	–	–	–
30-0.5-15	0.09	0.11	0.11	0.26	0.23	0.07	0.08	0.08	0.28	0.13	0.08	0.09	0.09	0.23	0.21
30-0.5-30	0.00	0.00	0.00	2.88	24.01	–	–	–	–	–	–	–	–	–	–
30-0.7-15	0.04	0.04	0.05	0.44	0.33	0.03	0.03	0.04	0.69	0.75	0.09	0.11	0.10	0.08	0.08
30-0.7-30	0.00	0.00	0.00	2.41	23.24	–	–	–	–	–	–	–	–	–	–
Average	0.13	0.14	0.14	0.55	3.05	0.15	0.15	0.12	0.82	4.90	0.12	0.12	0.13	0.37	2.35

Table A.11
Number of cuts generated by the cutting plane and the Branch-and-Cut algorithms.

Angle	$0^\circ \leq \alpha \leq 10^\circ$					$40^\circ \leq \alpha \leq 50^\circ$					$80^\circ \leq \alpha \leq 90^\circ$				
n - d - K	CP1	CP2	CP3	B&C1	B&C2	CP1	CP2	CP3	B&C1	B&C2	CP1	CP2	CP3	B&C1	B&C2
10-0.3-5	0.00	0.00	0.00	1.80	3.80	0.80	0.80	0.80	4.60	7.00	0.00	0.00	0.00	1.60	5.80
10-0.3-10	0.60	0.60	0.60	7.00	13.60	5.40	6.40	6.40	18.20	21.00	2.60	3.40	3.40	12.20	15.00
10-0.3-15	2.00	3.40	3.40	15.80	29.80	6.80	8.60	8.60	28.20	30.20	8.20	10.20	10.00	33.00	34.40
10-0.5-5	0.80	1.80	1.80	3.20	6.80	0.80	0.80	0.80	5.20	8.40	0.00	0.00	0.00	2.00	3.40
10-0.5-10	4.80	7.80	7.40	20.40	24.80	10.00	11.20	11.20	34.40	48.00	5.80	8.60	8.20	31.40	27.20
10-0.5-15	8.60	11.20	11.20	39.60	67.80	28.00	29.00	29.00	87.80	102.60	17.60	30.60	30.80	55.80	103.40
10-0.7-5	0.20	0.20	0.20	3.00	3.40	0.40	0.60	0.60	5.00	10.20	1.60	2.20	2.20	6.40	9.60
10-0.7 -10	2.40	6.20	6.40	23.40	32.80	17.40	24.80	25.20	69.20	70.40	8.00	10.40	10.40	33.80	61.40
10-0.7-15	4.60	7.40	8.40	22.60	38.00	49.20	95.40	94.80	209.80	342.40	29.00	54.20	53.60	90.80	266.00
20-0.3-10	1.80	2.20	2.20	16.40	17.20	3.00	4.80	4.80	22.00	22.80	3.60	3.40	3.60	13.80	24.40
20-0.3-20	17.00	28.20	54.60	118.60	259.40	65.00	127.40	124.00	293.20	812.60	59.40	119.60	123.60	273.40	421.20
20-0.3-30	28.60	63.20	62.60	145.40	586.80	178.20	271.20	285.40	1131.40	6726.80	107.80	183.00	186.40	460.20	911.20
20-0.5-10	1.60	2.40	2.40	11.80	22.00	5.20	6.00	6.00	32.80	25.00	4.60	7.60	7.80	17.80	28.40
20-0.5-20	15.00	31.60	30.40	104.20	135.20	172.20	233.00	184.40	317.67	665.33	39.40	74.40	76.00	167.00	727.60
20-0.5-30	50.80	126.20	129.00	296.40	2586.80	109.40	160.60	266.40	1248.40	7362.20	128.60	221.60	239.60	671.00	6853.60
20-0.7-10	1.40	2.40	2.40	9.80	16.80	4.20	4.80	4.80	19.40	25.80	0.60	0.60	0.60	13.80	16.60
20-0.7-20	30.60	65.60	74.80	112.60	299.00	167.80	245.20	184.00	879.75	6182.75	65.20	108.00	108.80	179.00	794.20
20-0.7-30	52.20	134.60	137.80	388.40	1765.00	217.00	255.40	271.40	2170.20	14154.60	130.60	207.60	218.00	675.40	5977.60
30-0.3-15	4.00	6.00	6.00	26.40	33.60	19.20	33.20	32.80	127.60	120.40	10.60	13.80	13.20	52.20	79.20
30-0.3-30	32.60	75.80	78.20	212.80	451.80	–	–	–	–	–	–	–	–	–	–
30-0.5-15	3.40	6.20	6.20	32.80	36.40	11.60	12.40	12.80	59.00	43.80	7.80	9.00	9.00	35.80	39.80
30-0.7-15	14.60	19.80	19.80	63.00	55.00	26.20	45.20	46.00	136.40	138.80	2.80	3.20	2.80	14.80	24.80
Average	12.62	27.40	29.35	76.15	294.81	52.28	75.09	76.20	328.58	1758.15	30.18	51.02	52.76	135.30	782.13

Table A.12
Number of separation problems solved by the iterative cutting plane and the Branch-and-Cut algorithms.

Angle	$0^\circ \leq \alpha \leq 10^\circ$					$40^\circ \leq \alpha \leq 50^\circ$					$80^\circ \leq \alpha \leq 90^\circ$				
n - d - K	CP1	CP2	CP3	B&C1	B&C2	CP1	CP2	CP3	B&C1	B&C2	CP1	CP2	CP3	B&C1	B&C2
10-0.3-5	1.00	1.00	1.00	3.00	3.60	1.40	1.40	1.40	4.20	4.80	1.00	1.00	1.00	3.00	4.00
10-0.3-10	1.40	1.40	1.40	3.60	4.60	2.60	3.00	3.00	7.40	6.40	2.40	2.60	2.60	6.20	7.20
10-0.3-15	1.80	2.40	2.40	6.20	6.60	3.40	4.40	4.40	8.80	10.20	3.00	3.60	3.60	7.80	7.80
10-0.5-5	1.80	2.80	2.80	4.20	6.00	1.40	1.40	1.40	5.00	5.80	1.00	1.00	1.00	3.00	3.20
10-0.5-10	2.80	4.40	4.20	9.60	11.80	4.40	4.80	4.80	11.20	14.20	3.40	4.80	4.60	10.60	9.80
10-0.5-15	3.40	4.40	4.20	10.80	22.80	7.00	9.00	9.00	17.80	20.40	6.00	10.00	10.00	14.40	28.40
10-0.7-5	1.20	1.20	1.20	4.40	3.40	1.40	1.60	1.60	5.20	5.40	1.80	2.20	2.20	5.00	6.00
10-0.7-10	2.60	4.60	4.60	10.40	12.40	7.40	11.00	10.80	22.80	22.60	4.20	4.80	4.80	11.20	17.80
10-0.7-15	3.00	4.00	4.60	7.60	11.00	11.00	19.40	19.20	40.00	69.60	8.40	13.00	12.40	19.20	55.80
20-0.3-10	2.00	2.00	2.00	6.40	5.60	2.60	3.60	3.60	9.60	7.60	2.80	3.00	3.00	6.20	7.60
20-0.3-20	6.20	9.60	18.00	23.00	64.00	13.80	26.40	26.00	47.60	138.00	13.60	30.40	31.60	49.60	97.40
20-0.3-30	6.40	15.40	14.80	23.80	106.80	17.40	21.60	23.60	105.20	722.40	14.40	23.40	23.60	49.60	106.40
20-0.5-10	2.40	2.80	2.80	6.80	7.20	3.20	3.80	3.80	13.00	8.00	3.60	5.20	5.40	10.60	10.80
20-0.5-20	5.80	9.20	8.40	22.80	27.20	24.60	47.20	33.20	77.60	361.00	10.00	19.60	19.60	28.20	110.60
20-0.5-30	9.20	18.80	18.80	43.20	301.60	15.67	19.00	20.80	76.00	471.00	16.60	24.20	26.00	66.60	829.00
20-0.7-10	1.80	2.20	2.20	6.00	5.80	3.20	3.40	3.40	8.20	9.40	1.60	1.60	1.60	6.60	5.20
20-0.7-20	10.40	25.00	28.80	31.80	106.00	26.40	44.80	26.60	110.25	655.25	15.40	23.00	23.40	29.60	138.80
20-0.7-30	9.20	17.40	18.00	60.00	222.60	15.00	15.80	16.80	156.40	1096.20	13.20	17.80	18.60	67.00	622.80
30-0.3-15	2.80	3.80	3.80	8.80	8.00	8.80	13.80	13.60	39.80	33.00	5.20	7.60	7.40	16.80	31.60
30-0.3-30	8.80	17.40	18.20	35.00	81.80	–	–	–	–	–	–	–	–	–	–
30-0.5-15	2.40	3.80	3.80	13.40	10.20	6.40	7.00	7.00	15.40	9.60	5.00	5.40	5.40	12.60	9.40
30-0.5-30	11.40	18.20	18.40	54.80	458.60	–	–	–	–	–	–	–	–	–	–
30-0.7-15	6.60	8.60	8.60	24.00	17.60	9.00	15.80	16.00	34.80	38.00	2.60	3.00	2.80	8.00	7.60
30-0.7-30	11.20	19.00	19.60	47.20	480.60	–	–	–	–	–	–	–	–	–	–
Average	4.82	8.31	8.86	19.45	82.74	8.86	13.25	11.90	38.87	176.61	6.44	9.87	10.03	20.56	100.82

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Amaldi, E., Bruglieri, M., Fortz, B., 2011. On the hazmat transport network design problem. In: Network Optimization. Springer, pp. 327–338.

Batta, R., Kwon, C., 2013. Handbook of OR/MS Models in Hazardous Materials Transportation. Springer.

Bazaraa, M.S., Jarvis, J.J., Sherali, H.D., 2011. Linear Programming and Network Flows. John Wiley & Sons.

Bianco, L., Caramia, M., Giordani, S., 2009. A bilevel flow model for hazmat transportation network design. Transp. Res. Part C 17 (2), 175–196.

Bonvicini, S., Spadoni, G., 2008. A hazmat multi-commodity routing model satisfying risk criteria: a case study. J. Loss Prev. Process Ind. 21 (4), 345–358.

Dempe, S., Kalashnikov, V., Perez-Valdes, G., Kalashnykova, N., 2015. Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks. Springer.

Dolan, E.D., Moré, J.J., 2002. Benchmarking optimization software with performance profiles. Math. Program. 91 (2), 201–213.

Erkut, E., Alp, O., 2007. Designing a road network for hazardous materials shipments. Comput. Oper. Res. 34 (5), 1389–1405.

Erkut, E., Gzara, F., 2008. Solving the hazmat transport network design problem. Comput. Oper. Res. 35 (7), 2234–2247.

González, P.H., Simonetti, L., Michelon, P., Martinhon, C., Santos, E., 2016. A variable fixing heuristic with local branching for the fixed charge uncapacitated network design problem with user-optimal flow. Comput. Oper. Res. 76, 134–146.

Gzara, F., 2013. A cutting plane approach for bilevel hazardous material transport network design. Oper. Res. Lett. 41 (1), 40–46.

Kara, B.Y., Verter, V., 2004. Designing a road network for hazardous materials transportation. Transp. Sci. 38 (2), 188–196.

LeBlanc, L.J., Boyce, D.E., 1986. A bilevel programming algorithm for exact solution of the network design problem with user-optimal flows. Transp. Res. Part B 20 (3), 259–265. doi:10.1016/0191-2615(86)90021-4.

Magnanti, T.L., Wong, R.T., 1984. Network design and transportation planning: models and algorithms. Transp. Sci. 18 (1), 1–55.

Marcotte, P., 1988. A note on a bilevel programming algorithm by LeBlanc and Boyce. Transp. Res. Part B 22 (3), 233–236. doi:10.1016/0191-2615(88)90018-5.

Mauttone, A., Labbé, M., Figueiredo, R., 2008. A tabu search approach to solve a network design problem with user-optimal flows. VI ALIO/EURO Workshop on Applied Combinatorial Optimization.

Toumazis, I., Kwon, C., Batta, R., 2013. Value-at-risk and conditional value-at-risk minimization for hazardous materials routing. In: Handbook of OR/MS Models in Hazardous Materials Transportation. Springer, pp. 127–154.

Verter, V., Kara, B.Y., 2008. A path-based approach for hazmat transport network design. Manage. Sci. 54 (1), 29–40.