Computers & Operations Research 40 (2013) 2485–2492

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Enhanced compact models for the connected subgraph problem and for the shortest path problem in digraphs with negative cycles

Mohamed Haouari [a,*], Nelson Maculan [b], Mehdi Mrad [c]

[a] Department of Mechanical and Industrial Engineering, College of Engineering, Qatar University, Doha, Qatar
[b] Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
[c] King Saud University, Riyadh, Saudi Arabia

ARTICLE INFO

Available online 18 January 2013

Keywords:
Connected subgraphs
Reformulation-Linearization Technique (RLT)
Shortest paths with negative cycles

ABSTRACT

We investigate the minimum-weight connected subgraph problem. The importance of this problem stems from the fact that it constitutes the backbone of many network design problems having applications in several areas including telecommunication, energy, and distribution planning. We show that this problem is $\mathcal{NP}$-hard, and we propose a new polynomial-size nonlinear mixed-integer programming model. We apply the Reformulation-Linearization Technique (RLT) to linearize the proposed model while keeping a polynomial number of variables and constraints. Furthermore, we show how similar modelling techniques enable an enhanced polynomial size formulation to be derived for the shortest elementary path. This latter problem is known to be intractable and has many applications (in particular, within the context of column generation). We report the results of extensive computational experiments on graphs with up to 1000 nodes. These results attest to the efficacy of the proposed compact formulations. In particular, we show that the proposed formulations consistently outperform compact formulations from the literature.

1. Introduction

Many combinatorial optimization problems in graphs require finding a minimum-weight connected subgraph with (possibly) some additional constraints. These problems find important real-world applications in telecommunication and energy network design, logistics, and transportation, to quote just a few. In this paper, we investigate the following (basic) network design problem. We are given a connected, undirected graph $G=(V,E)$, where $V=\{1,\ldots,n\}$ is the node set, and E is the edge set, along with a node weight γ_j associated with each node $j \in V$, and an edge weight c_e associated with each edge $e \in E$. An important distinctive feature of the model is that both the node weights and the edge weights are *unrestricted* in sign. The problem requires finding a *connected* subgraph $\tilde{G}=(\tilde{V},\tilde{E})$ (with $\tilde{E} \neq \emptyset$) that minimizes the sum of the weights of the covered nodes and edges. In the following, we refer to this problem as the *minimum-weight connected subgraph problem* (or, MWCSG for short). Our interest in the study of the MWCSG stems from the fact that it constitutes a basic "building block" for several important network design problems. That is, many well-known problems might be viewed as an MWCSG with additional constraints. In particular, if $\tilde{G}$ is constrained to be cycle-free then the MWCSG turns out to be closely related to several Steiner tree-type problems. Similarly, if $\tilde{G}$ should include a collection of cycles the MWCSG could be viewed as a sub-model of several distribution (i.e. vehicle routing) problems. The objective of this paper is to propose a new compact MIP model (i.e. with a polynomial number of variables and constraints) for the MWCSG. The importance of this research stems from the fact that the development of an effective compact formulation offers the significant advantage of making it feasible to solve the MWCSG (and some of its generalizations) using a general purpose solver. This is an appealing solution alternative that offers the significant advantage of avoiding investing time and effort in implementing sophisticated branch-and-cut or branch-and-price solution procedures. Furthermore, we provide evidence of the pertinence of the proposed model by showing how very similar ideas canbe used for deriving a new compact formulation for an important related connected subgraph problem, namely the shortest elementary path in digraphs with negative cycles. For both problems, we first derive nonlinear MIP formulations and then we apply the so-called *Reformulation-Linearization Technique* (RLT) of Sherali and Adams [14,15] to derive effective compact linear MIP formulations. It is worth mentioning that RLT has been previously successfully applied for deriving tight formulations for the Traveling Salesman Problem and its variants [13,16,17], and for the Prize Collecting Steiner Tree Problem [6].

The remainder of this paper is organized as follows. In Section 2, we propose a compact nonlinear MIP formulation for the MWCSG and then we show how to apply RLT to derive an equivalent linear

* Corresponding author.
E-mail address: mohamed.haouari@qu.edu.qa (M. Haouari).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.01.002

MIP model. In Section 3, we address the shortest elementary path in digraphs with negative cycles and we propose a compact RLT-based formulation. In Section 4, we present the results of a comprehensive computational study that provides strong empirical evidence that the proposed compact formulations renders it feasible to robustly solve both large-scale instances. Finally, some concluding remarks and recommendations for future research are provided in Section 5.

In the following, we shall conform with the following notation. Given a directed graph $G=(V,A)$, for a node $j \in V$, we define δ_j^+ and δ_j^- as the sets of arcs that are outgoing from and entering node j, respectively. Similarly, if the graph is undirected then δ_j refers the set of edges that are incident to node j.

2. The minimum-weight connected subgraph problem

2.1. Problem complexity

To the best of our knowledge, the complexity status of the MWCSG has never been established. In this section, we show that, despite its deceptive simplicity, the MWCSG cannot be solved efficiently unless $\mathcal{P}=\mathcal{NP}$.

Lemma 1. *The MWCSG is $\mathcal{NP}$-hard.*

Proof. The proof is based upon reduction from the Steiner tree problem (STP) in graphs which is known to be $\mathcal{NP}$-hard [7]. This problem is defined as follows. Assume that we are given a connected, undirected graph $G=(V,E)$, with a nonnegative weight c_e associated with each edge $e \in E$. The node set V is partitioned into two subsets S (set of *Steiner nodes*) and T (set of *terminal nodes*). The STP is to find a shortest tree that spans all the nodes in T, and possibly some additional nodes from $S=V\backslash T$. Given an STP instance, the reduction to an MWCSG instance that is defined on the same graph and with the same edge costs is achieved by further defining for each terminal node $j \in T$ a weight $\gamma_j=-M$ (where M is a very large nonnegative integer), and for each Steiner node a zero weight. Let $\tilde{G}=(\tilde{V},\tilde{E})$ denote the optimal solution of the derived MWCSG instance. We can make the following observations:

(i) $\tilde{G}$ is connected.
(ii) $\tilde{G}$ is acyclic (because the edge costs are nonnegative).
(iii) $T \subseteq \tilde{V}$. This is an immediate consequence of the large negative weights of the terminal nodes.

Hence, we see from (i) and (ii) that $\tilde{G}$ is a tree, and we deduce from (iii) that it covers all terminal nodes. Thus, the optimal solution of MWCSG is a feasible STP solution. Furthermore, the cost of this solution is $c^*-M|T|$ where c^* is the cost of the tree that covers the terminal nodes and (possibly) some Steiner nodes. Clearly, c^* is the value of the shortest tree that covers all the terminals, hence it is an optimal STP solution. Thus, if the MWCSG problem is solvable in polynomial time so is the STP. □

2.2. The compact formulation of Maculan, Plateau, and Lisser

To the best of our knowledge, the first (and only) reference that explicitly investigates the MWCSG is the paper by Maculan et al. [10]. In this paper, the authors present a compact mixed-integer programming formulation that is based on reformulating the MWCSG as a multicommodity network design problem. For the sake of self-containedness, we describe this formulation. First, we present the underlying digraph as well as the decision variables. Let $B=(V^+,A)$ denote the bidirected graph where the node set is derived from V by adding a zero-weight node 0 (that is, $V^+ \equiv V \cup \{0\}$), and the arc set A is obtained by replacing each edge $e=\{i,j\} \in E$ with two directed arcs (i,j) and (j,i) (with corresponding weights $c_{ij}=c_{ji}=c_e$), and by including n zero-weight arcs of the form $(0,j)$, $\forall j \in V$. The decision variables are defined as follows:

x_{ij}: binary variable that takes the value 1 if arc $(i,j) \in A$ belongs to the solution, and 0 otherwise.

y_j: binary variable that takes the value 1 if node $j \in V$ belongs to the solution, and 0 otherwise.

z_{ij}^k: value of the flow of commodity k ($k \in V$) in arc $(i,j) \in A$ (node 0 is the source and node k is the sink). The formulation is:

$$\textbf{MPL}: \quad \text{Minimize} \sum_{(i,j) \in A} c_{ij}x_{ij} + \sum_{j \in V} \gamma_j y_j \tag{1}$$

subject to

$$\sum_{j \in V} x_{0j} = 1, \tag{2}$$

$$\sum_{j \in \delta_0^+} z_{0j}^k = y_k \quad \forall k \in V, \tag{3}$$

$$\sum_{j \in \delta_i^+} z_{ij}^k - \sum_{j \in \delta_i^-} z_{ji}^k = 0 \quad \forall i \in V\backslash\{k\},\ k \in V, \tag{4}$$

$$\sum_{j \in \delta_k^-} z_{jk}^k - \sum_{j \in \delta_k^+} z_{kj}^k = y_k, \quad \forall k \in V, \tag{5}$$

$$x_{ij} \le y_j \quad \text{and} \quad x_{ji} \le y_j \quad \forall (i,j) \in A, \tag{6}$$

$$z_{ij}^k \le x_{ij} \quad \text{and} \quad z_{ji}^k \le x_{ij} \quad \forall (i,j) \in A,\ k \in V, \tag{7}$$

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

$$y_j \in \{0,1\} \quad \forall j \in V, \tag{9}$$

$$z_{ij}^k \ge 0 \quad \forall k \in V,\ (i,j) \in A. \tag{10}$$

The objective function (1) requires minimizing the total cost. Constraint (2) that there is exactly one arc outgoing from the root node. Constraints (3)–(5) enforce that if $y_k=1$, then z^k is the incidence vector of a path that originates at the root node and terminates at node k, and is null otherwise. Constraints (6) require that if node $\in V$ is not selected (that is, $y_j=0$) then no arc incident to that node can be included in the solution. Moreover, Constraints (7) enforces that if an arc $(i,j) \in A$ is not included in the arborescence, then no flow can be routed on this arc.

We make the following observations:

(i) Vectors z^k define a collection of paths that originate at the root node. Note that, by virtue of (2), all these paths share the same arc outgoing from the root. Thus, this collection of paths induces a connected subgraph.
(ii) Constraint (7) enforces that $z^k \le x$ $\forall k \in V$. Hence, x defines a connected subgraph as well. It is noteworthy that this latter constraint does not prevent x from including cycles.
(iii) From (2) we have just one node r such that $x_{0r}=1$, and by (6) we have $y_r=1$.

Hence, the subgraph associated with all $y_j=1$, $x_{ij}=1$ is a connected subgraph.

2.3. A nonlinear MIP formulation

Now, we turn our attention to describing a new valid formulation of MWCSG. This formulation is based on the observation that a connected subgraph in G necessarily induces an arborescence in the corresponding bi-directed graph B. Hence, the proposed formulation is based not only on the above-defined decision vectors x and y, but also on the binary decision variable w_e that takes the value 1 if edge $e \in E$ belongs to the solution, and 0 otherwise ($e \in E$). Hence, in the present model x is the incidence vector of an *arborescence* in B, and w is the incidence vector of a connected subgraph of G. In addition, we define the continuous decision variable u_j that represents the number of arcs in the dipath in B that is induced by the incidence vector x that connects node 0 to node $j \in V$.

The formulation is given as follows:

$$\textbf{MWCSG}: \quad \text{Minimize} \sum_{e \in E} c_e w_e + \sum_{j \in V} \gamma_j y_j \tag{11}$$

subject to

$$\sum_{j \in V} x_{0j} = 1, \tag{12}$$

$$\sum_{i:(i,j) \in A} x_{ij} = y_j \quad \forall j \in V, \tag{13}$$

$$x_{ij} + x_{ji} \le w_e \quad \forall e = \{i,j\} \in E, \tag{14}$$

$$w_e \le y_j \quad \forall j \in V, \ e \in \delta_j, \tag{15}$$

$$u_j x_{0j} = x_{0j} \quad \forall j \in V, \tag{16}$$

$$u_j x_{ij} = (u_i + 1) x_{ij} \quad \forall (i,j) \in A, \ i,j \in V \tag{17}$$

$$1 \le u_j \le n \quad \forall j \in V, \tag{18}$$

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

$$y_j \in \{0,1\} \quad \forall j \in V, \tag{20}$$

$$w_e \in \{0,1\} \quad \forall e \in E. \tag{21}$$

The objective (11) is to minimize the sum of the costs of the edges and the nodes that are included in the connected subgraph. The set of constraints enforce that

(C1): x is the incidence vector of an arborescence $\mathcal{A}(x)$ in B.
(C2): $\mathcal{A}(x)$ covers the node subset whose incidence vector is y.
(C3): If $(x_{ij} = 1) \vee (x_{ji} = 1)$ then $w_e = 1, \ \forall e = \{i,j\} \in E$ (this logical restriction requires that the solution w induces an arborescence x in B).
(C4): If $y_j = 0$ then $w_e = 0 \ \forall j \in V, \ e \in \delta_j$.

(C1) and (C2) are enforced by Constraints (12), (13), (16)–(20). Indeed, Constraint (12) requires that the root node is connected to exactly one node. Constraint (13) enforces that each covered non-root node (that is, $y_j = 1$ for $j \in V$) has exactly one incoming arc incident to it. Otherwise, if $y_j = 0$, then node j has no incident arc. The nonlinear Constraints (16) and (17) are the *subtour-elimination constraints*. These constraints, similar to those proposed by Sherali and Driscoll [16] in the context of the Traveling Salesman Problem (TSP), enforce that for any $j \in V$, $u_j = 1$ if $x_{0j} = 1$, and $u_j = u_i + 1$ if $x_{ij} = 1$. It is easy to check that these latter constraints, together with (18) and (19), prevent any subtour. The validity of these constraints stems from the validity of the so-called *Miller–Tucker–Zemlin constraints* that were first proposed by Miller et al. [11] to derive a polynomial-size formulation for the TSP.

Furthermore, the logical restrictions (C3) and (C4) are enforced by Constraints (14) and (15), respectively.

Let $(\overline{x}, \overline{y}, \overline{w}, \overline{u})$ be a solution that satisfies (12)–(21). Moreover, let $S(\overline{y}) = \{j \in V : \overline{y}_j = 1\}$ be the set of covered nodes, and denote by $G(\overline{x})$ the subgraph that is induced by $\overline{x}$, and $G(\overline{w})$ the undirected graph that is induced by $\overline{w}$. We shall show that Formulation (11)–(21) is valid by successively showing that:

(i) $G(\overline{x})$ is an arborescence of B that is rooted at node 0 and spans the nodes within $S(\overline{y}) \cup \{0\}$.
(ii) $G(\overline{w})$ is a connected subgraph of G having a minimal total cost.

Proposition 1. *$G(\overline{x})$ is an arborescence of B that is rooted at node 0 and spans the nodes within $S(\overline{y}) \cup \{0\}$.*

Proof. First, note that (16) and (17) guarantee that $\overline{x}$ is acyclic. Now, assume that $G(\overline{x})$ is not connected. Since it is cycle-free, then it necessarily includes several rooted arborescences. Node 0 has no incoming arc and, by virtue of (12), has exactly one outgoing arc. Furthermore, Constraint (13) enforces that each node $j \in S(\overline{y})$ has one incoming arc and therefore cannot be a root. Thus, $G(\overline{x})$ includes a single acyclic connected component rooted at node 0. □

Denote by $r \in V$ the node such that $\overline{x}_{0r} = 1$, and by $\tilde{G}(\overline{x})$ the subgraph that is derived from $G(\overline{x})$ by deleting arc $(0,r)$ and node 0.

Remark 1. $\tilde{G}(\overline{x})$ is an arborescence of B that is rooted at node r and spans the nodes within $S(\overline{y})$.

Proof. This is an immediate consequence of Proposition 1 and the fact that: (i) $(0,r)$ is the only arc in $G(\overline{x})$ that is outgoing from the root node 0, and (ii) $(0,r)$ is the only arc in $G(\overline{x})$ that is incident to r. □

Proposition 2. *$G(\overline{w})$ is connected and covers the nodes in $S(\overline{y})$.*

Proof. We see from (14), that if for $e = \{i,j\}$, we have $\overline{x}_{ij} = 1$ or $\overline{x}_{ji} = 1$, then $w_e = 1$. Therefore, the connectedness of $G(\overline{w})$ is an immediate consequence of Remark 1. Moreover, Constraint (15) enforces that if $y_j = 0$ $(j \in V)$ then node j is not included in $G(\overline{w})$. □

Conversely, given a connected subgraph of G and the associated incidence vector w, one can easily derive corresponding vectors x, y, and u that satisfy Constraints (12)–(21). Thus, Model (11)–(21) is a valid formulation for Problem MWCSG.

2.4. Model reformulation and linearization

In this section, we show how to linearize Model (11)–(21) by applying the Reformulation-Linearization Technique (RLT) of Sherali and Adams [14,15]. To that aim, we introduce two new classes of variables $t \equiv (t_{ij})_{(i,j) \in A}$ and $v \equiv (v_j)_{j \in V}$ to linearize constraints (16) and (17) by using the substitutions

$$t_{ij} = u_i x_{ij} \quad \forall (i,j) \in A, \quad \text{and } v_j = u_j y_j \ \ \forall j \in V. \tag{22}$$

Using Eq. (13), we construct the following equalities:

$$u_j \left[\sum_{i:(i,j) \in A} x_{ij} = y_j \right] \quad \forall j \in V,$$

which yields

$$\sum_{i:(i,j) \in A} u_j x_{ij} = u_j y_j \quad \forall j \in V.$$

Hence, from (17), we obtain

$$\sum_{i:(i,j)\in A}(u_i+1)x_{ij}=u_jy_j \quad \forall j\in V.$$

Hence, by substitution we get the valid equality

$$\sum_{i:(i,j)\in A}t_{ij}+\sum_{i:(i,j)\in A}x_{ij}=v_j \quad \forall j\in V. \tag{23}$$

Clearly, the inequality $x_{0j}\le y_j$ holds for each node $j\in V$. Also, if $y_j-x_{0j}=1$, then we must have $u_j\ge 2$. Hence we get

$$(y_j-x_{0j})(u_j-2)\ge 0 \quad \forall j\in V.$$

Therefore, we derive the inequality

$$2y_j-x_{0j}\le v_j \quad \forall j\in V. \tag{24}$$

Likewise, since the inequality $u_j\le n$ is valid for any node $j\in V$, then

$$(y_j-x_{0j})(n-u_j)\ge 0 \quad \forall j\in V,$$

and therefore we get

$$v_j\le ny_j-(n-1)x_{0j} \quad \forall j\in V. \tag{25}$$

Noting that $x_{ij}=1$ implies that $u_i\le n-1$, the following inequalities are valid:

$$(n-1-u_i)x_{ij}\ge 0 \quad \forall i\in V,\ (i,j)\in A,$$

$$t_{ij}\le (n-1)x_{ij} \quad \forall i\in V,\ (i,j)\in A. \tag{26}$$

Also, $u_i\ge 1$ $\forall i\in V$, hence we have

$$t_{ij}\ge x_{ij} \quad \forall i\in V,\ (i,j)\in A. \tag{27}$$

Finally, the product of the constraints

$$x_{ij}+x_{ji}\le y_j \quad \forall\{(i,j),(j,i)\}\subset A,\ j\in V,$$

by the upper and lower bounding conditions (that is, $u_j\le n$ and $u_j\ge 1$), respectively yield

$$(y_j-x_{ij}-x_{ji})(u_j-1)\ge 0 \quad \forall\{(i,j),(j,i)\}\in A,\ j\in V,$$

and

$$(y_j-x_{ij}-x_{ji})(n-u_j)\ge 0 \quad \forall\{(i,j),(j,i)\}\in A,\ j\in V.$$

Hence, we derive the valid inequalities

$$v_j-y_j+x_{ji}\ge t_{ij}+t_{ji} \quad \forall\{(i,j),(j,i)\}\in A,\ j\in V, \tag{28}$$

and

$$v_j-ny_j+(n-1)x_{ij}+nx_{ji}\le t_{ij}+t_{ji} \quad \forall\{(i,j),(j,i)\}\in A,\ j\in V, \tag{29}$$

Finally, we get the following valid MILP formulation

$$\textbf{MWCSG-RLT}: \quad \text{Minimize} \sum_{e\in E}c_ew_e+\sum_{j\in V}\gamma_jy_j \tag{30}$$

subject to

$$\sum_{j\in V}x_{0j}=1, \tag{31}$$

$$\sum_{i:(i,j)\in A}x_{ij}=y_j \quad \forall j\in V, \tag{32}$$

$$\sum_{i:(i,j)\in A}t_{ij}+\sum_{i:(i,j)\in A}x_{ij}=v_j \quad \forall j\in V, \tag{33}$$

$$2y_j-x_{0j}\le v_j\le ny_j-(n-1)x_{0j} \quad \forall j\in V, \tag{34}$$

$$x_{ij}\le t_{ij}\le (n-1)x_{ij} \quad \forall (i,j)\in A, \tag{35}$$

$$x_{ij}+x_{ji}\le w_e \quad \forall e=\{i,j\}\in E, \tag{36}$$

$$v_j-y_j+x_{ji}\ge t_{ij}+t_{ji} \quad \forall (i,j)\in A,\ j\in V, \tag{37}$$

$$v_j-ny_j+(n-1)x_{ij}+nx_{ji}\le t_{ij}+t_{ji} \quad \forall (i,j)\in A,\ j\in V, \tag{38}$$

$$w_e\le y_j \quad \forall e\in\delta_j,\ j\in V, \tag{39}$$

$$x_{ij}\in\{0,1\} \quad \forall (i,j)\in A, \tag{40}$$

$$y_j\in\{0,1\} \quad \forall j\in V, \tag{41}$$

$$w_e\in\{0,1\} \quad \forall e\in E, \tag{42}$$

$$t_{ij}\ge 0 \quad \forall (i,j)\in A,\ i\in V, \tag{43}$$

$$v_j\ge 0 \quad \forall j\in V. \tag{44}$$

Proposition 3. *For any feasible solution* (x,y,w,v,t) *to* (31)–(44), *we can associate a solution* (x,y,w,u) *that is feasible to* (12)–(21) *and having the same objective and vice versa.*

Proof. Let (x,y,w,v,t) be a solution that is feasible to (31)–(44). We set $u_j=v_jy_j$ for $j\in V$ and we need to show that (x,y,w,u) is feasible to (12)–(21). To that aim, we only need to show that u satisfies Constraints (16)–(18).

Assume that $y_j=1$. Then, $u_j=v_j$. From (32), we see that there exists exactly one node i^* such that $x_{i^*j}=1$. Two cases may occur:

*Case*1: $i^*=0$. Then, from (34) we have $v_j=1$. Thus, (16) holds.

*Case*2: $i^*>0$. Then, from (34), we get

$$2\le v_j\le n.$$

Hence, (18) holds. Also, from (33) we get

$$u_j=\sum_{i:(i,j)\in A}t_{ij}+1. \tag{45}$$

Since (35) holds, we have $t_{ij}=0$ for all $i\ne i^*$. Thus, we have

$$u_j=t_{i^*j}+1. \tag{46}$$

Since $x_{i^*j}=1$, then we deduce from (36) and (39) that $y_{i^*}=1$. Since $x_{ji^*}=0$ (by virtue of (36)), then it follows that $t_{ji^*}=0$ (by virtue of (35)). Thus, from (37) we get $t_{i^*j}\le v_{i^*}$. Similarly, from (38) we get $t_{i^*j}\ge v_{i^*}$. Hence, we have $t_{i^*j}=v_{i^*}$, and therefore (46) yields

$$u_j=u_{i^*}+1. \tag{47}$$

Thus, (17) is satisfied.

Conversely, given a feasible solution (x,y,w,u), we use the substitutions (22) to derive a corresponding feasible solution (x,y,w,v,t). □

Clearly, Model **MWCSG-RLT** includes $O(n^2)$ binary variables, $O(n^2)$ continuous variables, and $O(n^2)$ constraints.

3. The shortest path problem in digraphs with negative cycles

3.1. Problem definition

The shortest path problem is probably the most basic combinatorial optimization problem in graphs. Since the early days of operations research, two well-known methods were proposed by Ford (1957) [20] and Dijkstra (1958) [19], respectively. In these seminal papers, the authors propose algorithms that are efficient and relatively easy. However, both algorithms address special cases of the shortest path problem. Indeed, while Bellman-Ford's algorithm requires that the graph includes no cycles of negative weight, Dijsktra's algorithm is even more stringent as it requires that the arc/edge weights are nonnegative. In addition, several nontrivial generalizations of this basic problem have been investigated in the literature. In particular, the *resource constrained shortest path problem* has been intensely investigated by many authors (see [1] and the references therein). By contrast, a glaring fact is that the variant of the shortest path problem, where the graph includes negative cycles, received scant attention. This might be due to the fact that this latter problem is known to be $\mathcal{NP}$ -hard and therefore cannot be solved efficiently unless $\mathcal{P}=\mathcal{NP}$ [5]. Yet, the shortest path problem in graphs with negative cycles has many important applications in particular in the context of column generation. Indeed, in these specific applications the pricing problem amounts to finding an elementary shortest path in an appropriate graph where the arc weights represent reduced costs that are unrestricted in sign [4]. In this context, Rousseau et al. [12] proposed to solve the shortest elementary path with negative cycles arising in the solution of the vehicle routing problem with time windows using constraint programming. Also, Chabrier [2] investigated labeling algorithms for the same problem. Furthermore, Ibrahim et al. [9] proposed a compact size formulation (that shall be described in the next section). Actually, it is worth mentioning that most of the papers so far published that investigate the shortest path problem in graphs with negative cycles deal with the issue of the detection of negative cycles (see [3,8,18] and the references therein).

In this section, we turn our attention to building a new compact MIP formulation (in the same vein as the one that we derived for the MWCSG) for the *Shortest Elementary Path with Negative Cycles* (SEPNC). Formally, the SEPNC is defined as follows. We are given a connected digraph $G=(V,A)$ and two nodes $s \in V$ and $t \in V$. With each arc $(i,j) \in A$ is associated a (possibly negative) integer weight c_{ij}. The problem requires finding an elementary path (i.e. cycle-free) between s and t and such that the total weight is minimum.

In the following, we assume with no loss of generality that $\delta_s^- = \emptyset$, $\delta_t^+ = \emptyset$, and $(s,t) \notin A$.

3.2. The compact formulation of Ibrahim, Maculan, and Minoux

A first compact formulation for the SEPNC has been presented by Ibrahim et al. [9]. This formulation uses binary decision variables x_{ij} ($(i,j) \in A$), and y_j ($j \in V$), and z_{ij}^k ($(i,j) \in A$, $k \in V\backslash\{s\}$) that are defined similarly to those used in Formulation MPL. The compact formulation of Ibrahim, Maculan, and Minoux (IMM) is the following:

$$\textbf{IMM}: \quad \text{Minimize} \sum_{(i,j) \in A} c_{ij}x_{ij} \tag{48}$$

subject to

$$\sum_{j \in \delta_s^+} x_{sj} = 1, \tag{49}$$

$$\sum_{j \in \delta_t^-} x_{jt} = 1, \tag{50}$$

$$\sum_{j \in \delta_i^+} x_{ij} = y_i \quad \forall i \in V\backslash\{s,t\}, \tag{51}$$

$$\sum_{j \in \delta_i^-} x_{ji} = y_i \quad \forall i \in V\backslash\{s,t\}, \tag{52}$$

$$\sum_{j \in \delta_s^+} z_{sj}^k - \sum_{j \in \delta_s^-} z_{js}^k = y_k \quad \forall k \in V\backslash\{s\}, \tag{53}$$

$$\sum_{j \in \delta_i^+} z_{ij}^k - \sum_{j \in \delta_i^-} z_{ji}^k = 0 \quad \forall k \in V\backslash\{s\},\ i \in V\backslash\{s,k\}, \tag{54}$$

$$\sum_{j \in \delta_k^+} z_{kj}^k - \sum_{j \in \delta_k^-} z_{jk}^k = -y_k \quad \forall k \in V\backslash\{s\}, \tag{55}$$

$$z_{ij}^k \le x_{ij} \quad \forall k \in V\backslash\{s\},\ (i,j) \in A, \tag{56}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A, \tag{57}$$

$$y_j \in \{0,1\} \quad \forall j \in V, \tag{58}$$

$$z_{ij}^k \ge 0 \quad \forall (i,j) \in A,\ k \in V\backslash\{s\}. \tag{59}$$

This formulation includes $O(n^2)$ binary variables, $O(n^3)$ continuous variables, and $O(n^3)$ constraints.

3.3. A nonlinear MIP formulation and application of the RLT process

In this section, we present an RLT-based compact formulation for the SEPNC in the spirit of Formulation **MWCSG-RLT**. In particular, both formulations use the same x and u decision vectors. This formulation is the following:

$$\textbf{SEPNC-RLT}: \quad \text{Minimize} \sum_{(i,j) \in A} c_{ij}x_{ij} \tag{60}$$

subject to

$$\sum_{j \in \delta_s^+} x_{sj} = 1, \tag{61}$$

$$\sum_{j \in \delta_t^-} x_{jt} = 1, \tag{62}$$

$$\sum_{i \in \delta_j^-} x_{ij} - \sum_{i \in \delta_j^+} x_{ji} = 0 \quad \forall j \in V\backslash\{s,t\}, \tag{63}$$

$$\sum_{i \in \delta_j^-} x_{ij} \le 1 \quad \forall j \in V\backslash\{s,t\}, \tag{64}$$

$$u_j x_{ij} = (u_i+1)x_{ij} \quad \forall j \in V\backslash\{s,t\}, i \in \delta_j^-\backslash\{s\}, \tag{65}$$

$$u_j x_{sj} = x_{sj} \quad \forall j \in \delta_s^+, \tag{66}$$

$$1 \le u_j \le n-1 \quad \forall j \in V\backslash\{s\}, \tag{67}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A. \tag{68}$$

Constraints (61)–(63), and (68) enforce that x defines a path that connects s to t. Furthermore, Constraints (64)–(67) enforce that the path is elementary. Indeed, Constraint (64) requires that each node j ($j \in V\backslash\{s,t\}$) has at most one incident arc, and Constraints (65)–(67) are the standard subtour-elimination constraints.

We can linearize (65) by using the substitutions

$$\alpha_{ij} = u_j x_{ij} \quad \forall j \in V\backslash\{s\},\ i \in \delta_j^-\backslash\{s\}, \tag{69}$$

and

$$\beta_{ij} = u_i x_{ij} \quad \forall j \in V\backslash\{s\},\ i \in \delta_j^-\backslash\{s\}. \tag{70}$$

Thus, (65) get transformed to the following:

$$\alpha_{ij} = \beta_{ij} + x_{ij} \quad \forall j \in V\backslash\{s,t\},\ i \in \delta_j^-\backslash\{s\}. \tag{71}$$

Also, using (63) we construct the equality

$$u_j \left[\sum_{i \in \delta_j^-} x_{ij} - \sum_{i \in \delta_j^+} x_{ji} = 0 \right] \quad \forall j \in V\backslash\{s,t\},$$

which yields

$$\sum_{i\in\delta_j^-} u_j x_{ij} - \sum_{i\in\delta_j^+} u_j x_{ji} = 0 \quad \forall j\in V\backslash\{s,t\}.$$

Two cases are considered

*Case*1: If $j\in\delta_s^+$, then using (66) and upon substituting we get

$$x_{sj} + \sum_{i\in\delta_j^-\backslash\{s\}} \alpha_{ij} - \sum_{i\in\delta_j^+} \beta_{ji} = 0 \quad \forall j\in\delta_s^+. \tag{72}$$

*Case*2: If $j\in V\backslash\{\delta_s^+\cup\{s,t\}\}$, then we derive the equality

$$\sum_{i\in\delta_j^-\backslash\{s\}} \alpha_{ij} - \sum_{i\in\delta_j^+} \beta_{ji} = 0 \quad \forall j\in V\backslash\{\delta_s^+\cup\{s,t\}\}. \tag{73}$$

Finally, from (67) we derive

$$x_{ij} \le \alpha_{ij} \le (n-1)x_{ij} \quad \forall j\in V\backslash\{s\},\ i\in\delta_j^-\backslash\{s\}, \tag{74}$$

and (upon rearranging)

$$x_{ij} \le \beta_{ij} \le (n-1)x_{ij} \quad \forall j\in V\backslash\{s\},\ i\in\delta_j^-\backslash\{s\}. \tag{75}$$

Hence, we derive the following valid compact formulation:

$$\textbf{SEPNC-RLT}: \quad \text{Minimize} \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{76}$$

subject to (61)–(64), (68) and

$$\alpha_{ij} = \beta_{ij} + x_{ij} \quad \forall j\in V\backslash\{s\},\ i\in\delta_j^-\backslash\{s\}, \tag{77}$$

$$x_{sj} + \sum_{i\in\delta_j^-\backslash\{s\}} \alpha_{ij} - \sum_{i\in\delta_j^+} \beta_{ji} = 0 \quad \forall j\in\delta_s^+, \tag{78}$$

$$\sum_{i\in\delta_j^-} \alpha_{ij} - \sum_{i\in\delta_j^+} \beta_{ji} = 0 \quad \forall j\in V\backslash\{\delta_s^+\cup\{s,t\}\}, \tag{79}$$

$$x_{ij} \le \alpha_{ij} \le (n-1)x_{ij} \quad \forall j\in V\backslash\{s\},\ i\in\delta_j^-\backslash\{s\}, \tag{80}$$

$$x_{ij} \le \beta_{ij} \le (n-1)x_{ij}, \quad \forall j\in V\backslash\{s\},\ i\in\delta_j^-\backslash\{s\}. \tag{81}$$

Theorem 1. *Model SPP-RLT is a valid formulation of Problem SEPNC.*

Proof. Let $\zeta=(x,\alpha,\beta)$ be a solution that is feasible to (61)–(64), (68), and (77)–(81). First, we observe that (61)–(63) and (68) enforce that x defines a directed path from s to t, plus a (possibly empty) collection of disjoint cycles (which are disjoint also w.r.t. the path from s to t). Thus, we need to show that x includes no cycles. The proof will be by contradiction. Assume that x includes a cycle $(\sigma(1),\ldots,\sigma(p))$ that covers p nodes ($p>0$). That is, we have $x_{\sigma(q),\sigma(q+1)}=1\ \forall q=1,\ldots p$ (with $p+1\equiv 1$). Clearly, since we assumed that $\delta_s^-=\emptyset$, $\delta_t^+=\emptyset$, then the cycle neither covers s nor t (that is, $\sigma(q)\notin\{s,t\}\ \forall q=1,\ldots p$).

We set

$$u_j = \sum_{i\in\delta_j^+} \beta_{ji} \quad \forall j\in\{\sigma(1),\ldots,\sigma(p)\}. \tag{82}$$

From (64), we deduce that $x_{s,\sigma(q)}=0\ \forall q=1,\ldots p$. Hence, from (78)–(79) and (82), we get

$$u_j = \sum_{i\in\delta_j^-\backslash\{s\}} \alpha_{ij} \quad \forall j\in\{\sigma(1),\ldots,\sigma(p)\}. \tag{83}$$

Note that from (64) we deduce that if $x_{\sigma(q),\sigma(q+1)}=1$ then $x_{i,\sigma(q+1)}=0\ \forall i\in\delta_{\sigma(q+1)}^-\backslash\{\sigma(q)\}$. Hence, from (80) we get that $\alpha_{i,\sigma(q+1)}=0$ for all $i\in\delta_{\sigma(q+1)}^-\backslash\{\sigma(q)\}$. Thus, (83) yields

$$u_{\sigma(q+1)} = \alpha_{\sigma(q),\sigma(q+1)} \text{ for all } q=1,\ldots p. \tag{84}$$

Similarly, from (63) to (64) we note that if $x_{\sigma(q+1),\sigma(q+2)}=1$ then $x_{\sigma(q+1),i}=0\ \forall i\in\delta_{\sigma(q+1)}^+\backslash\{j_{\sigma(q+2)}\}$. Hence, from (81) we get that $\beta_{\sigma(q+1),i}=0$ for all $i\in\delta_{\sigma(q+1)}^+\backslash\{\sigma(q+2)\}$. Thus, (82) yields

$$u_{\sigma(q+1)} = \beta_{\sigma(q+1),\sigma(q+2)} \text{ for all } q=1,\ldots p. \tag{85}$$

From (77), we have: $\alpha_{\sigma(q),\sigma(q+1)}=\beta_{\sigma(q),\sigma(q+1)}+1$, for all $q=1,\ldots p$. By summing up these latter p equalities, we get $\sum_{q=1,p}\alpha_{\sigma(q),\sigma(q+1)}=\sum_{q=1,p}\beta_{\sigma(q),\sigma(q+1)}+p$. Hence, from (84) and (85) we get

$$\sum_{q=1,p} u_{\sigma(q+1)} = \sum_{q=1,p} u_{\sigma(q)} + p. \tag{86}$$

This leads to a contradiction with the fact that $p>0$ and proves that ζ defines an elementary path.

Conversely, given an elementary path $\mathcal{P}=(s,\sigma(1),\ldots,\sigma(p),t)$ between nodes s and t, it is easy to construct a corresponding feasible solution ζ by setting: $x_{s,\sigma(1)}=1, x_{\sigma(p),t}=1$, and $x_{\sigma(q),\sigma(q+1)}=1$ for $q=1,\ldots,p-1$, and $x_{ij}=0$ otherwise. Furthermore, from (78) and (81) we set $\beta_{\sigma(1),\sigma(2)}=1$. Hence, it follows from (77) that $\alpha_{\sigma(1),\sigma(2)}=2$. More generally, it is easy to check that we have $\beta_{\sigma(q),\sigma(q+1)}=q$ and $\alpha_{\sigma(q),\sigma(q+1)}=q+1$ for $q=1,\ldots,p$ (with $\sigma(p+1)\equiv t$), and $\alpha_{ij}=\beta_{ij}=0$ otherwise. □

4. Computational results

In this section, we provide the results of an extensive computational study that we conducted for assessing the empirical performance of the proposed formulations. To that aim, we randomly generated digraphs using the following procedure [9]. First, for each node j ($j=1,\ldots,n-1$), the outdegree d_j^+ is randomly drawn form $U[1,\min\{3,n-j-1\}]$. Second, d_j^+ arcs that are outgoing from j ($j=1,\ldots,n-1$) are created by randomly choosing d_j^+ endpoints in $\{j+1,\ldots,n\}$. If in the resulting partial graph there exists a node j ($j=2,\ldots,n$) having no incident arc, then we further include an arc (i,j) where i is randomly drawn from in $\{1,\ldots j-1\}$. To create cycles, for each node j ($j=2,\ldots,n$), we create with a probability 0.5 an arc (j, k) where k is randomly chosen in $\{2,\ldots,j-1\}$. Finally, for each arc $(i,j)\in A$, the corresponding cost is first randomly drawn from $U[1,50]$ and then, to favor the existence of negative cycles, is multiplied by -1 with probability $\frac{2}{3}$. The node weights are generated in a similar way. It is worth noting that for generating undirected graphs for the MWCSG instances, we used the same procedure but we simply replaced each pair of opposite arcs (if any) by a single randomly chosen arc, and then we replace each arc by an edge.

All runs were made on an Intel Pentium 2.5 GHz eight-processor computer having 16 GB of RAM. The commercial solver CPLEX 12.1 was used as the solver with the default parameter settings. For each run, the CPU time limit was set to 1 h.

4.1. Performance of formulation MWCSG-RLT

A summary of the results are displayed in Table 1. The column headings in these tables are as follows: n: number of nodes; m: number of edges; *OPT*: value of the optimal solution; *Time* 1: CPU time (in s) required by **MPL**, *Time* 2: CPU time (in s) required by **MWCSG-RLT**, *LR1_Gap*: Absolute value of the percentage deviation of the LP relaxation of **MPL**; *LR2_Gap*: Absolute value of the percentage deviation of the LP relaxation of **MWCSG-RLT**; *LR1_Time*: CPU time (in s) required by the LP relaxation of **MPL**; *LR2_Time*: CPU time (in s) required by the LP relaxation of **MWCSG-RLT**.

A first striking observation is that **MWCSG-RLT** successfully delivered optimal solutions for all the instances (with up to 1000

Table 1
Performance of the **MPL** and **MWCSP-RLT** formulations.

n	*m*	*OPT*	*Time 1*	*Time 2*	*LR1_Gap*	*LR2_Gap*	*LR1_Time*	*LR2_Time*
10	**21**	−352	0.007	0.000	0.000	0.000	0.003	0.000
10	**25**	−361	0.007	0.000	0.000	0.000	0.003	0.000
20	**63**	−951	0.025	0.016	0.000	0.000	0.007	0.015
20	**69**	−1089	0.027	0.016	0.000	0.000	0.014	0.015
30	**105**	−1519	0.122	0.093	0.000	0.000	0.076	0.016
30	**102**	−1417	0.063	0.047	0.000	0.000	0.068	0.000
40	**141**	−1902	0.184	0.078	0.000	0.000	0.079	0.000
40	**142**	−1809	0.158	0.031	0.000	0.000	0.062	0.000
50	**189**	−2916	0.364	0.125	0.000	0.000	0.158	0.016
50	**185**	−3228	0.204	0.093	0.000	0.000	0.062	0.016
60	**220**	−3173	0.318	0.047	0.000	0.000	0.104	0.031
60	**223**	−4175	0.836	0.047	0.000	0.000	0.310	0.031
70	**261**	−3234	0.836	0.062	0.000	0.000	0.397	0.031
70	**261**	−3780	0.761	0.140	0.000	0.000	0.309	0.016
80	**303**	−4282	1.060	0.171	0.000	0.000	0.515	0.032
80	**306**	−5501	1.326	0.078	0.000	0.000	0.640	0.032
90	**348**	−5484	1.529	0.250	0.000	0.000	0.609	0.047
90	**360**	−5517	1.747	0.514	0.000	0.000	0.733	0.047
100	**396**	−6366	1.762	0.094	0.000	0.000	0.765	0.047
100	**391**	−7049	2.933	0.109	0.000	0.000	1.264	0.031
200	**799**	−12 939	17.456	1.638	0.000	0.000	12.917	0.094
200	**815**	−12 971	22.745	1.623	0.000	0.000	19.749	0.093
300	**1213**	−18 938	84.850	2.449	0.000	0.000	70.750	0.156
300	**1215**	−21 172	88.000	4.790	0.000	0.000	73.740	0.156
400	**1640**	−26 084	OM*	2.886	–	0.000	–	0.234
400	**1667**	−26 881	OM	1.654	–	0.000	–	0.234
500	**2058**	−33 345	OM	4.899	–	0.000	–	0.312
500	**2072**	−32 799	OM	3.214	–	0.000	–	0.328
600	**2471**	−38 912	OM	79.934	–	0.000	–	0.515
600	**2474**	−39 551	OM	16.255	–	0.000	–	0.562
700	**2902**	−47 771	OM	75.317	–	0.000	–	0.795
700	**2867**	−45 800	OM	74.537	–	0.000	–	0.780
800	**3266**	−51 549	OM	29.266	–	0.000	–	0.795
800	**3315**	−53 699	OM	192.350	–	0.000	–	0.827
900	**3724**	−58 330	OM	83.554	–	0.000	–	0.982
900	**3732**	59 022	OM	10.81	–	0.000	–	0.983
1000	**4141**	−66 664	OM	110.056	–	0.000	–	1.248
1000	**4176**	−65 836	OM	275.009	–	0.000	–	1.560

(*) OM means "Out of Memory".

nodes) while requiring very short CPU times. In contrast, **MPL** failed to solve all instances having more than 400 nodes. For these large instances, we observed that **MPL** requires a huge memory that our computer (though having 16 GB RAM) cannot handle. Furthermore, for all the instances that were solved by both formulations, **MWCSG-RLT** requires significantly shorter CPU times than **MPL** does. A second nice observation, is that both LP relaxations yield zero gaps but the LP relaxation of the RLT-based formulation is faster. This is a clear indication of the efficacy of the proposed formulation.

4.2. Performance of formulation SEPNC-RLT

For all the instances, the path origin and destination are nodes 1 and n, respectively. The results are provided in Table 2. In this table, columns "*Time* 3", "*LR3_Gap*", and "*LR3_Time*" display the results obtained by **IMM**, while columns "*Time* 4", "*LR4_Gap*", and "*LR4_Time*" refer to **SEPNC-RLT.**

We see from Table 2 that, here again, the RLT-based formulation consistently outperforms the multicommodity-flow formulation as it yielded proven optimal solutions for all the instances. Also, we observe that the required CPU are extremely short. Indeed, the largest instance (having 1000 nodes) required less that 5 s, while all instances having fewer than 600 nodes only required less that 1 s. Therefore, there is empirical evidence that the proposed RLT-based formulation might prove useful in the context of column generation where similar shortest path problems (with possibly additional constraints) require to be iteratively solved. Finally, we observe that even though both LP relaxations exhibit very small gaps, the LP relaxation of **IMM** is often tighter.

5. Summary and conclusions

In this paper, we have proposed an enhanced formulation for the minimum-weight connected subgraph problem. This problem appears as the backbone of many important network design problems. We show that despite its deceptive simplicity, this problem is $\mathcal{NP}$-hard. Also, we proposed an enhanced mixed-integer linear model with a polynomial number of variables and constraints. This formulation was derived by applying the Reformulation-Linearization Technique. To provide evidence of the practical relevance of this research, we have shown how similar modeling features could be implemented for the shortest elementary path problem with negative cycles. This latter problem is notoriously $\mathcal{NP}$-hard and finds many applications in column generation. Furthermore, extensive computational results were presented using a large set of randomly generated instances to analyze the empirical efficacy of the proposed formulations. An analysis of the performance of the proposed formulations show that large instances are optimally solved within reasonable CPU times. By contrast, we found that multicommodity flow-based formulations from the literature, though yielding very tight LP relaxations, are consistently outperformed on large instances.

An interesting issue that is worthy of future research is to carry out a theoretical comparison of multicommodity-based and

Table 2
Performance of the **IMM** and **SEPNC-RLT** formulations.

n	*m*	*OPT*	*Time 3*	*Time 4*	*LR3_Gap*	*LR4_Gap*	*LR3_Time*	*LR4_Time*
10	**29**	−185	0.005	0.002	0.00	0.00	0.002	0.001
10	**29**	−232	0.006	0.002	0.00	0.00	0.002	0.001
20	**70**	−392	0.029	0.005	0.00	0.00	0.013	0.001
20	**74**	−446	0.029	0.005	0.00	0.00	0.011	0.001
30	**114**	−815	0.932	0.092	0.96	2.45	0.099	0.004
30	**110**	−744	0.113	0.007	0.00	0.00	0.043	0.003
40	**154**	−1057	2.452	0.127	0.24	3.94	0.441	0.008
40	**158**	−928	4.252	0.145	1.15	7.59	0.385	0.006
50	**196**	−1281	10.927	0.171	1.77	2.40	0.742	0.008
50	**193**	−1214	5.755	0.111	0.99	1.56	0.774	0.009
60	**242**	−1519	15.21	0.274	0.88	2.65	1.018	0.011
60	**234**	−1623	18.996	0.208	0.84	1.52	1.057	0.013
70	**276**	−1824	110.32	0.202	0.95	1.78	1.641	0.011
70	**285**	−1912	38.257	0.078	0.25	0.64	1.715	0.017
80	**323**	−2012	137.502	0.405	0.83	1.66	3.158	0.020
80	**318**	−2176	87.945	0.289	0.22	0.60	2.576	0.024
90	**353**	−2339	179.777	0.498	0.45	1.28	4.002	0.025
90	**357**	−2293	108.026	0.162	0.38	0.44	2.269	0.020
100	**414**	−2768	341.877	0.693	0.35	0.42	4.272	0.027
100	**407**	−2628	232.364	0.344	0.40	0.44	2.412	0.027
200	**827**	−5288	1563.551	2.973	0.17	0.47	154.505	0.105
200	**821**	−5133	2791.88	3.276	0.12	0.28	46.95	0.111
300	**1227**	−7856	> 3600	13.860	0.09	0.49	211.421	0.265
300	**1229**	−7988	> 3600	13.582	0.11	0.26	274.493	0.383
400	**1657**	−10 327	OM*	16.637	–	0.10	–	0.459
400	**1660**	−10 276	OM	34.980	–	1.07	–	0.493
500	**2078**	−12 999	OM	30.057	–	0.14	–	0.915
500	**2085**	−13 086	OM	51.680	–	0.17	–	0.924
600	**2486**	−15 169	OM	78.311	–	0.07	–	1.234
600	**2469**	−15 573	OM	116.784	–	0.09	–	1.316
700	**2907**	−18 188	OM	195.525	–	0.09	–	2.113
700	**2932**	−17 841	OM	151.365	–	0.15	–	1.99
800	**3280**	−20 510	OM	423.538	–	0.11	–	2.693
800	**3329**	−20 269	OM	205.748	–	0.04	–	2.268
900	**3693**	−23 049	OM	644.500	–	0.05	–	3.498
900	**3712**	−23 719	OM	231.671	–	0.04	–	3.790
1000	**4158**	−25 680	OM	344.832	–	0.08	–	3.430
1000	**4176**	−26 076	OM	477.425	–	0.11	–	4.623

(*) OM means "Out of Memory".

RLT-based formulations for similar problems in graphs. Also, a second issue that deserves further investigation is to derive tight compact formulations for more complex optimization problems in graphs. These models would offer the significant advantage of making it feasible to solve small or (possibly) medium-sized instances using general-purpose MIP solvers. We believe that these models would positively impinge on the spread of real-world applications of combinatorial optimization.

References

[1] Avella P, Boccia M, Sforza A. Resource constrained shortest path problems in path planning for fleet management. Journal of Mathematical Modelling and Algorithms 2004;3:1–17.

[2] Chabrier A. Vehicle routing problem with elementary shortest path based column generation. Computers & Operations Research 2006;33:2972–90.

[3] Deng X, Hopcroft JE, Xue J. Improved algorithms for detecting negative cost cycles in undirected graphs. Lecture Notes in Computer Science 2009;5598: 40–50.

[4] Desaulniers G, Desrosiers J, Solomon MM. Column generation. Berlin, Heidelberg, New York: Springer; 2005.

[5] Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP-completeness. San Francisco, CA: W.H. Freeman; 1979.

[6] Haouari M, Layeb-Bhar S, Sherali HD. Tight compact models and comparative analysis for the prize collecting Steiner tree problem. Discrete Applied Mathematics 2013;161:618–32.

[7] Hwang FK, Richards DS, Winter P. The Steiner tree problem. Amsterdam: North-Holland; 1992.

[8] Hougardy S. The Floyd–Warshall algorithm on graphs with negative cycles. Information Processing Letters 2010;110:279–81.

[9] Ibrahim MS, Maculan N, Minoux M. A strong flow-based formulation for the shortest path problem in digraphs with negative cycles. International Transactions in Operational Research 2009;16:361–9.

[10] Maculan N, Plateau G, Lisser A. Integer linear models with a polynomial number of variables and constraints for some classical combinatorial optimization problems. Pesquisa Operacional 2003;23:161–8.

[11] Miller CE, Tucker AW, Zemlin RA. Integer programming formulations and traveling salesman problems. Journal of the Association for Computing Machinery 1960;7:326–9.

[12] Rousseau LM, Gendreau M, Pesant G. Solving VRPTWs with constraint programming based column generation. Annals of Operations Research 2004;130:199–216.

[13] Sarin SC, Sherali HD, Bhootra A. New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints. Operations Research Letters 2003;33:62–70.

[14] Sherali HD, Adams WP. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM Journal of Discrete Mathematics 1990;3:411–30.

[15] Sherali HD, Adams WP. A hierarchy of relaxations and convex hull characteristics for mixed-integer zero-one programming problems. Discrete Applied Mathematics 1994;52:83–106.

[16] Sherali HD, Driscoll PJ. On tightening the relaxations of Miller–Tucker–Zemlin formulations for asymmetric traveling salesman problems. Operations Research 2002;50:656–69.

[17] Sherali HD, Sarin SC, Tsai P. A class of lifted path and flow-based formulations for the asymmetric traveling salesman problem with and without precedence constraints. Discrete Optimization 2006;3:20–32.

[18] Subramani K, Tauras C, Madduri K. Space-time tradeoffs in negative cycle detection-an empirical analysis of the stressing algorithm. Applied Mathematics and Computation 2010;215:3563–75.

[19] Dijkstra EW. A note on two problems in connexion with graphs. Numerische Mathematik 1957;1:269–71.

[20] Ford Jr. LR. Network Flow Theory, Technical report. Rand Corporation; 1956.