The Rooted Maximum Node-Weight Connected Subgraph Problem

Eduardo Álvarez-Miranda[1], Ivana Ljubić[2], and Petra Mutzel[3]

[1] Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, Italy, e.alvarez@unibo.it
[2] Department of Statistics and Operations Research, University of Vienna, Vienna, Austria, ivana.ljubic@univie.ac.at
[3] Faculty of Computer Science, TU Dortmund, Otto-Hahn-Str. 14, 44227 Dortmund, Germany, petra.mutzel@tu-dortmund.de

Abstract. Given a connected node-weighted (di)graph, with a root node r, and a (possibly empty) set of nodes R, the *Rooted Maximum Node-Weight Connected Subgraph Problem* (RMWCS) is the problem of finding a connected subgraph rooted at r that connects all nodes in R with maximum total weight. In this paper we consider the RMWCS as well as its budget-constrained version, in which also non-negative costs of the nodes are given, and the solution is not allowed to exceed a given budget. The considered problems belong to the class of network design problems and have applications in various different areas such as wildlife preservation planning, forestry, system biology and computer vision.

We present three new integer linear programming formulations for the problem and its variant which are based on node variables only. These new models rely on a different representation of connectivity than the one previously presented in the RMWCS literature that rely on a transformation into the Steiner Arborescence problem. We theoretically compare the strength of the proposed and the existing formulations, and show that one of our models preserves the tight LP bounds of the previously proposed cut set model of Dilkina and Gomes. Moreover, we study the rooted connected subgraph polytope in the natural space of node variables. We conduct a computational study and (empirically) compare the theoretically strongest one of our formulations with the one previously proposed using ad-hoc branch-and-cut implementations.

1. Introduction

In this work we study a variant of the *connected subgraph problem* in which we are given a graph with a pre-specified root node (and possibly an additional set of terminals). Nodes of the graph are associated with (not necessarily positive) weights. The goal is to find a connected subgraph containing the root and the terminals that maximizes the sum of node-weights. In addition, a budget constraint may be imposed as well: in this case, each node is additionally associated with a non-negative cost, and the cost of connecting the nodes is not allowed to exceed the given budget. Both problem variants are NP-hard, unless all node weights are non-negative and no budget is imposed, in which case the problem is trivial. The problem is called the *Rooted Maximum Node-Weight Connected Subgraph Problem* (RMWCS), or the RMWCS with *Budget Constraint* (B-RMWCS), respectively.

The problem has been introduced by Lee and Dooly [12] in the context of the design of fiber-optic communication networks over time, where the authors refer to the problem as the *constrained maximum weight connected graph problem*. The authors impose K-cardinality constraints, i.e., they search for a connected subgraph containing K nodes (including a predetermined root) that maximizes the collected node-weights. Obviously, K-cardinality constraints are a special form of the budget constraints in which every node is associated a cost equal to one, and the budget is equal to K.

A budgeted version arises in the wildlife conservation planning, where the task is to select land parcels for conservation to ensure species viability, also called *corridor design* (see, e.g. [4, 5]). Here, the nodes correspond to land parcels, their weights are associated with the habitat suitability, and node costs are associated with land value. The task is to design wildlife corridors that maximize the suitability with a given limited budget. Also in forest planning, the connected subgraph arises as subproblem, e.g., for designing a contiguous site for a natural reserve or for preserving large contiguous patches of mature forest [2]. Moss and Rabani [15] have proposed an $O(\log n)$ approximation algorithm for the B-RMWCS with non-negative node-weights, where n is the number of nodes in the graph. For more details on the problems related on the RMWCS, see e.g., the literature review given in [5].

In this paper we will address the RMWCS in digraphs as well. This is motivated by some applications in systems biology where regulatory networks are represented using (not necessarily bidirected) digraphs and with node weights that can also be negative. The goal is to find a rooted subgraph in which there is a directed path from the root to any other node that maximizes the sum of node weights. In systems biology, the roots are frequently referred to as "seed genes" as they are assumed to be involved in a particular disease. In Backes et al. [1], for example, the authors search for the connected subgraph in a digraph without a prespecified root node (i.e., determination of the seed gene, also called the key player, is part of the optimization process). To solve the problem of Backes et al. [1] one can, for example, iterate over a set of potential key players, solve the corresponding RMWCS and choose the best solution.

Our Contribution. Previously studied mixed integer programming (MIP) formulations for the (B-)RMWCS use arc and possibly flow variables to model the problem (see Dilkina and Gomes [5]). In this paper we propose three new MIP models for the (B-)RMWCS derived in the natural space of node variables. We first provide a theoretical comparison of the quality of lower bounds of these models. We also show that one of our models which is based on the concept of *node separators*, preserves the tight LP bounds of the previously proposed *cut set* model of Dilkina and Gomes [5]. In the second part of the paper we study the rooted connected subgraph polytope (in the natural space of node variables) and show under which conditions the node separator inequalities are facet-defining. In an extensive computational study, we compare the node-separator and the cut-set model on a set of benchmark instances for the wildlife corridor design problem used in [5] and on a set of network design instances.

Outline of the Paper. Three new MIP models for the (B-)RMWCS are proposed in Section 2. A comparison of the MIP models and results regarding the facets of the rooted connected subgraph polytope are given in Section 3 and computational results are presented in Section 4.

2. MIP Formulations for the RMWCS

In this section we present three new MIP models for the RMWCS and its budget-constrained variant. Before that, we first review the model recently proposed by Dilkina and Gomes [5] which is based on the reformulation of the problem into the (budget-constrained) Steiner arborescence problem. The latter model is derived on the space of arc variables, while the remaining ones are defined in the natural space of node variables.

Since every RMWCS on undirected graphs can be considered as the same problem on digraphs (by replacing every edge with two oppositely directed arcs), in the remainder of this paper we will present the more general results for digraphs. The corresponding results for undirected graphs can be easily derived from them.

Definitions and Notation. Formally, we define the RMWCS as follows: Given a digraph $G = (V \cup \{r\}, A)$, with a root r, a set of terminals $R \subset V$, and node weights $p : V \to \mathbb{R}$, the RMWCS is the problem of finding a connected subgraph $T = (V_T, A_T)$, that spans the nodes from $\{r\} \cup R$ and such that every node $j \in V_T$ can be reached from r by a directed path in T, and that maximizes the sum of node weights $p(T) = \sum_{v \in V_T} p_v$. Additionally, in the B-RMWCS, node costs $c : V \to \mathbb{R}^+$ and a budget limit $B > 0$ are given. The goal is to find a connected subgraph T that maximizes $p(T)$ and such that its cost does not exceed the given budget, i.e., $c(T) = \sum_{v \in V_T} c_v \leq B$.

A set of vertices $S \subset V$ ($S \neq \emptyset$) and its complement $\bar{S} = V \setminus R$, induce two directed cuts: $(S, \bar{S}) = \delta^+(S) = \left\{(i,j) \in A \mid i \in S, j \in \bar{S}\right\}$ and $(\bar{S}, S) = \delta^-(S) = \left\{(i,j) \in A \mid i \in \bar{S}, j \in S\right\}$. For a set $C \subset V$, let $D^-(C)$ denote the set of nodes outside of C that have ingoing arcs into C, i.e., $D^-(C) = \{i \in V \setminus C \mid \exists (i,v) \in A, v \in C\}$.

A digraph G is called strongly connected (or simply, *strong*) if for any two distinct nodes k and ℓ from V, there exists a (k,ℓ) path in G. A node i is a *cut point* in a strong digraph G if there exists a pair of distinct nodes k and ℓ from V such that there is no (k,ℓ) path in $G - i$. A node i is a *cut point with respect to* r if there exists a node $k \neq i, r$ such that there is no (r,k) path in $G - i$. For two distinct nodes k and ℓ from V, a subset of nodes $N \subseteq V \setminus \{k,\ell\}$ is called

(k, ℓ) *(node) separator* if there exists a (k, ℓ) path in G and after eliminating N from V there is no (k, ℓ) path in G. A (k, ℓ) separator N is *minimal* if $N \setminus \{i\}$ is not a (k, ℓ) separator, for any $i \in N$. Let $\mathcal{N}(k, \ell)$ denote the family of all (k, ℓ) separators. Obviously, if $\exists (k, \ell) \in A$ or if ℓ is not reachable from k, we have $\mathcal{N}(k, \ell) = \emptyset$.

For variables $\mathbf{a}$ defined on a finite set F, we denote by $a(F')$ the sum $\sum_{i \in F'} a_i$ for any subset $F' \subseteq F$. Throughout the paper, let the graph $G = (V \cup \{r\}, A)$, $n = |V|$, and $m = |A|$.

2.1 Directed Steiner Tree Model of Dilkina and Gomes [5]

Dilkina and Gomes [5] propose to solve the B-RMWCS as a budget-constrained directed Steiner tree problem rooted at r. Their models are based on the observation that it is sufficient to search for a subtree (subarborescence) since no costs are associated to arcs in G, hence every solution containing cycles can be reduced without changing the weight. It is sufficient to use arc variables to model the problem since in a directed tree, the in-degree of every node is equal to one, so that the objective function can be expressed as $\max \sum_{i \in V} p_i z(\delta^-(i))$, where z are binary variables associated with the arcs of A that encode the subarborescence. Dilkina and Gomes [5] proposed three MIP models for the B-RMWCS. Two of them are flow based formulations (a single-commodity flow and a multi-commodity flow based one). The authors showed that the flow-based formulations are computationally outperformed by the cut-set model which is presented below.

We further use a set of auxiliary binary variables y for the vertex set V, where y_i will be equal to one if node i is part of the subtree, and zero, otherwise. In other words, we basically perform the substitution $y_i = z(\delta^-(i))$. The set of feasible B-RMWCS solutions can be described using inequalities (1)-(4). Constraints (1) and (2) ensure that the solution is a Steiner arborescence rooted at r, equations (3) make sure that all terminals are connected and (4) is the budget constraint:

$$z(\delta^-(i)) = y_i \qquad \forall i \in V \setminus \{r\} \tag{1}$$
$$z(\delta^-(S)) \geq y_k \qquad \forall k \in S,\ \forall S \subseteq V \setminus \{r\},\ S \neq \emptyset \tag{2}$$
$$y_i = 1 \qquad \forall i \in R \tag{3}$$
$$\mathbf{c}^T y \leq B \tag{4}$$

Constraints (2), also known as *cut* or *connectivity inequalities* ensure that there is a directed path from the root r to each node k such that $y_k = 1$. *In-degree* constraints (1) guarantee that the in-degree of each vertex of the arborescence is equal to one. Thus, the rooted Steiner arborescence model for the B-RMWCS (denoted by (SA_r)) is given as

$$(SA)_r \quad \max\left\{p^T y \mid (\mathbf{y}, \mathbf{z}) \text{ satisfies (1)-(4)},\ (\mathbf{y}, \mathbf{z}) \in \{0,1\}^{n+m}\right\}.$$

We notice that in Ljubić et al. [14] these sets of constraints and the transformation into the directed Steiner tree were used for solving the Prize-Collecting

Steiner Tree problem (PCStT). A connection between the PCStT and the unrooted MWCS has been observed by Dittrich et al. [7]: the authors showed that the unrooted MWCS can be transformed into the PCStT and used the branch-and-cut approach from [14] to solve the MWCS on a large protein-protein interaction network. Consequently, the same relation holds for the rooted MWCS as well.

The previous model uses node and arc variables ($\mathbf{y}$ and $\mathbf{z}$) given that it relies on a transformation into the Steiner arborescence problem. However it seems more natural to find a formulation based only in the space of $\mathbf{y}$ variables since no arc costs are involved in the objective function. In the next section we will discuss several models that enable elimination of arc variables in the MIP models.

2.2 Node-Based Formulations for the RMWCS

We now propose three MIP models that are derived in the natural space of y variables defined as above. We search for an arborescence rooted at r, but this time, we avoid explicit use of arc variables.

Model Based on Subtour Elimination Constraints. This model is an adaptation of the model by Backes et al. [1] that was recently proposed for the unrooted MWCS on directed graphs. The following inequalities will be called the *in-degree constraints*:

$$y(D^-(i)) \geq y_i, \quad \forall i \in V \setminus (\{r\} \cup D^+(r)) \tag{5}$$

They ensure that, whenever a node i is taken into a solution, at least one of its incoming neighbors has to be in the solution as well (notice that we do not need to impose this constraint for the outgoing neighbors of the root node). Constraints (5) however do not guarantee that the obtained solution is connected to the root. Let $\mathcal{C}$ denote the family of all directed cycles in G that do not contain the root node and are not "neighbors" of the root, i.e.:

$$\mathcal{C} = \{C \mid C \text{ is a cycle in } G, \text{ s.t. } r \notin C, \text{ and } r \notin D^-(C)\}.$$

In order to ensure connectivity of the solution, Backes et al. [1] add the following constraints, that we will refer to as the *subtour elimination constraints*:

$$y(C) - y(D^-(C)) \leq |C| - 1, \quad \forall C \in \mathcal{C}. \tag{6}$$

These constraints state that for each cycle $C \in \mathcal{C}$ whose node set is contained in the solution, at least one of the neighboring nodes outside of that cycle needs to belong to the solution as well. The model, that we will denote by $CYCLE_r$ reads as follows:

$$(CYCLE_r) \quad \max \left\{ p^T y \mid \mathbf{y} \text{ satisfies (3)-(6)}, \ \mathbf{y} \in \{0,1\}^n \right\}.$$

A Flow-Based Model. Alternatively to the previous model, to ensure connectivity, we can use multi-commodity flows where the available arc capacities are defined as the minimum node capacities at each end of an arc. Finding a feasible solution now means allocating node capacities that will enable to send one unit of flow from the root to each of the nodes taken into the subnetwork. In this context, constraints (5) and (6) can be replaced by the following set of constraints that ensure that there is enough capacity on the nodes so that a unit of flow can be sent from the root to any other node $i \in V \setminus \{r\}$ with $y_i = 1$. These constraints state that (i) whenever an arc is part of a feasible solution of the RMWCS, both of its end nodes are included into the solution and (ii) the induced subgraph is connected:

$$\sum_{(i,j)\in\delta^-(S)} \min\{y_i, y_j\} \geq y_k, \quad \forall k \notin \{r\} \cup D^+(r),\ \forall S \subseteq V \setminus \{r\},\ k \in S. \tag{7}$$

Constraints (7) represent just a compact way of writing $2^{|\delta^-(S)|}$ inequalities (see also [3] where these constraints have been proposed for a problem arising in the design of telecommunication networks). They can be separated in polynomial time by solving a maximum-flow problem in an auxiliary support graph. Observe finally that indegree constraints (5) are also implied by these constraints: For each node $i \notin r \cup D^+(r)$, we have $y(D^-(i)) \geq \sum_{(j,i)\in\delta^-(i)} \min\{y_j, y_i\} \geq y_i$. We can now define the B-RMWCS as

$$(CUT_m) \qquad \max\left\{p^T y \mid \mathbf{y} \text{ satisfies (3),(4),(7) and } \mathbf{y} \in \{0,1\}^n\right\}.$$

Formulation Based on Node Separators. The other way of modeling the connectivity of a solution using only node variables is to consider *node separators*. This idea has been recently used in Fügenschuh and Fügenschuh [8], Carvajal et al. [2] and Chen et al. [3] to model connectivity in the context of sheet metal design, forest planning, and telecommunication network design, respectively. The following inequalities will be called *node-separator constraints*:

$$y(N) \geq y_k, \quad \forall k \notin \{r\} \cup D^+(r),\ N \in \mathcal{N}(r,k). \tag{8}$$

These constraints ensure that for each node k taken into the solution, either k is a direct neighbor of r, or there has to be a path from r to k such that for each node i on this path, $y_i = 1$. Notice that whenever $\mathcal{N}(k,\ell) \neq \emptyset$, $D^-(k) \in \mathcal{N}(k,\ell)$ and in this case the in-degree inequalities (5) are contained in (8). Thus, we can formulate the B-RMWCS as

$$(CUT_r) \qquad \max\left\{p^T y \mid \mathbf{y} \text{ satisfies (3),(4),(8), } \mathbf{y} \in \{0,1\}^n\right\}.$$

2.3 Some More Useful Constraints

In case that the budget constraint (4) is imposed, the following family of cover inequalities can be used to cut off infeasible solutions.

Cover Inequalities. We say that a subset of nodes $V_C \subset V$ is a cover if the sum of node costs in V_C is greater than the allowed budget B. In that case, at least one node from V_C has to be left out in any feasible solution. A cover V_C is minimal if $C \setminus \{i\}$ for any $i \in V_C$ is not a cover anymore. Let $\mathcal{V}_C$ be a family of all minimal covers with respect to B. Then, the following *cover inequalities* are valid for the B-RMWCS:

$$\sum_{i \in V_C} y_i \leq |V_C| - 1, \quad \forall V_C \in \mathcal{V}_C \tag{9}$$

For further details on cover inequalities, see e.g. [10].

3. Polyhedral Results

In this section we compare the proposed MIP formulations with respect to their quality of LP bounds and we show that, under certain conditions, the newly introduced node-separator inequalities are facets of the rooted connected subgraph polytope.

3.1 Theoretical Comparison of MIP Models

Let $\mathcal{P}_{\mathrm{LP}}(.)$ denote the polytope of the LP-relaxations of the MIP models presented above and $v_{LP}(.)$ their optimal LP-values. We can show that:

Proposition 1 *We have $\mathcal{P}_{\mathrm{LP}}(CUT_r) \subsetneq \mathcal{P}_{\mathrm{LP}}(CUT_m) \subsetneq \mathcal{P}_{\mathrm{LP}}(CYCLE_r)$, and there exist instances for which the strict inequality holds.*

Proof. $\mathcal{P}_{\mathrm{LP}}(CUT_m) \subsetneq \mathcal{P}_{\mathrm{LP}}(CYCLE_r)$: Consider a feasible solution $\hat{\mathbf{y}}$ of the LP relaxation of model CUT_m. We will show that each such solution is feasible for the model $CYCLE_r$. Let C be an arbitrary cycle from $\mathcal{C}$. Then, obviously, for any node $k \in C$, we have $\hat{y}_i(D^-(C)) \geq \sum_{(i,j)\in\delta^-(C)} \min\{\hat{y}_i, \hat{y}_j\} \geq \hat{y}_k$. Adding up this inequality with inequalities $1 \geq \hat{y}_i$, for each $i \in C \setminus \{k\}$, we obtain: $\hat{y}(D^-(C)) + |C| - 1 \geq \hat{y}(C)$ which is exactly the subtour elimination inequality associated to C. To see that the strict inequality holds, consider the directed graph shown in Figure 1(a).
$\mathcal{P}_{\mathrm{LP}}(CUT_r) \subsetneq \mathcal{P}_{\mathrm{LP}}(CUT_m)$: Consider a feasible solution $\hat{\mathbf{y}}$ of the LP relaxation of the CUT_r model. Let $k \in V \setminus (\{r\} \cup D^+(r))$ be an arbitrary node such that $\hat{y}_k > 0$ and let $S \subset V \setminus \{r\}$ be a set such that $k \in S$. Then, we will show that $\sum_{(ij)\in\delta^-(S)} \{\hat{y}_i, \hat{y}_j\} \geq \hat{y}_k$, i.e., $\hat{\mathbf{y}}$ satisfies (7). Let $N_1 = \{i \mid (i,j) \in \delta^-(S)\}$. Observe that $r \notin N_1$ and by definition, N_1 is a node separator for k, i.e., $N_1 \in \mathcal{N}(r,k)$. Let $N_2 = \{j \mid (i,j) \in \delta^-(S)\}$: (i) If $k \notin N_2$, then N_2 is a node separator for k ($N_2 \in \mathcal{N}(r,k)$). Consider the bipartite graph defined by $\delta^-(S)$. Each possible vertex cover $N' \subset N_1 \cup N_2$ on this graph, induces a node separator for k, i.e., $N' \in \mathcal{N}(r,k)$. There are $2^{|\delta^-(S)|}$ vertex covers in total, and constraints (8) associated to them imply constraint (7); (ii) if $k \in N_2$, then all vertex covers involving k trivially satisfy $\hat{y}(N') \geq \hat{y}_k$ for $k \in N'$. Together with the remaining vertex covers, inequality (7) is implied. An example shown in Figure 1(b) shows an instance for which the strict inequality holds. $\square$

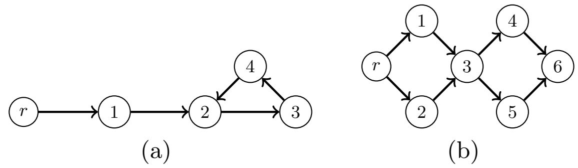

Fig. 1. Examples that prove the strength of the new formulations. (a) The LP-solution of $CYCLE_r$ sets $y_2 = y_3 = y_4 = 2/3$ and $y_1 = 0$, and this solution is infeasible for the model CUT_m. (b) The LP-solution of CUT_m satisfies $y_1 = \cdots = y_5 = 1/2$ and $y_6 = 1$. This solution is infeasible for CUT_r.

Proposition 2 *The* (SA_r) *model and the* (CUT_r) *model are equally strong, i.e.,* $v_{LP}(SA_r) = v_{LP}(CUT_r)$.

Proof. We first show that $v_{LP}(SA_r) \geq v_{LP}(CUT_r)$: Let $(\hat{\mathbf{z}}, \hat{\mathbf{y}})$ be a feasible solution for the relaxation of the SA_r model. Let $k \in V \backslash \{r\}$ be a node such that $\hat{y}_k > 0$ and let $N \in \mathcal{N}(r,k)$. Because of in-degree constraints of the SA_r model, we have that $\sum_{i \in N} \hat{y}_i = \sum_{i \in N} \hat{\mathbf{z}}(\delta^-(i))$. If N is removed from G, k cannot be reached from r. Let $S_r \subseteq V$, $r \in S_r$, be all the nodes i that can be reached from r after removing N, and let $S_k = V \setminus (N \cup S_r)$, $k \in S_k$. Because of inequalities (2), it holds that $\hat{\mathbf{z}}(\delta^+(S_r)) \geq \hat{y}_k$. Moreover, observe that for each $(i,j) \in \delta^+(S_r)$ we have that $i \in S_r$ and $j \in N$, which means that $\sum_{i \in N} \hat{\mathbf{z}}(\delta^-(i)) \geq \hat{\mathbf{z}}(\delta^+(S_r))$. Therefore, $\sum_{i \in N} \hat{y}_i \geq \hat{y}_k$, which proves that any LP solution of the SA_r model can be projected into a feasible solution of the CUT_r with the same objective value.

To show that $v_{LP}(CUT_r) \geq v_{LP}(SA_r)$ consider a solution $\check{\mathbf{y}} \in \mathcal{P}_{\mathrm{LP}}(CUT_r)$. We will construct a solution $(\hat{\mathbf{y}}, \hat{\mathbf{z}}) \in \mathcal{P}_{\mathrm{LP}}(SA_r)$ such that $\check{\mathbf{y}} = \hat{\mathbf{y}}$. On the graph G' (see Section 4.1, separation of separator inequalities) with arc capacities of (i_1, i_2) set to $\check{y}_i$ for each $i \in V \setminus \{r\}$ and to 1 otherwise, we are able to send $\check{y}_k$ units of flow from the root r to every (k_1, k_2) such that $\check{y}_k > 0$. Let f_{ij}^k denote the amount of flow of commodity k, sent along an arc $(i,j) \in A'$. Let $\mathbf{f}$ be the minimal feasible multi-commodity flow on G' (i.e., the effective capacities on G' used to route the flow cannot be reduced without violating the feasibility of this flow). We now define the values of $(\hat{\mathbf{y}}, \hat{\mathbf{z}})$ as follows:

$$\hat{z}_{ij} = \begin{cases} \max_{k \in V \backslash \{r\}} f_{i_2 j_1}^k, & i,j \in V \setminus \{r\} \\ \max_{k \in V \backslash \{r\}} f_{i,j_1}^k, & i = r, j \in V \setminus \{r\} \end{cases}, \forall (i,j) \in A, \qquad \text{and}$$

let $\hat{y}_i = \hat{\mathbf{z}}(\delta^-(i))$, for all $i \in V \setminus \{r\}$. Obviously, the constructed solution $(\hat{\mathbf{y}}, \hat{\mathbf{z}})$ is feasible for the (SA_r) model, and, due to the assumption that $\mathbf{f}$ is minimal feasible, it follows that $\check{\mathbf{y}} = \hat{\mathbf{y}}$, which concludes the proof. $\square$

Finally, regarding the strength of the three MIP models studied by Dilkina and Gomes [5], we notice that their single-commodity flow model is weaker than the multi-commodity model, which is equally strong as the cut-set model (SA_r) (see, e.g., [13]).

3.2 Facets of the RCS Polytope

In this section we consider the RMWCS with $R = \emptyset$, and let $\mathcal{P}$ denote the *rooted connected subgraph (RCS) polytope* defined in the natural space of $\mathbf{y}$ variables:

$$\mathcal{P} = \mathrm{conv}\{\mathbf{y} \in \{0,1\}^n \mid \mathbf{y} \text{ satisfies } (8)\}.$$

In this section we establish under which conditions some of the presented inequalities are facets of the RCS polytope.

Lemma 1. *The RCS polytope is full-dimensional (i.e.,* $\dim(\mathcal{P}) = n$*) if and only if there exists a directed path between* r *and any* $i \in V$*.*

Proof. We first generate a spanning arborescence T in G rooted at r. We will then apply a *tree pruning technique* in order to generate $n+1$ affine independent feasible RMWCS solutions. We start with the arborescence T in which case $\mathbf{y}$ consists of all ones. We iteratively remove one by one leaf from T, until we end up with a single root node (in which case $\mathbf{y}$ is a zero vector). Thereby, we generate a set of $n+1$ affinely independent solutions. Conversely, if $\mathcal{P}$ is full dimensional, then in order to create a feasible solution containing an arbitrary node $i \in V$, there has to be a directed path between r and i in G. □

Lemma 2. *Inequality* $y_i \geq 0$ *for* $i \in V$ *is facet defining if and only if in the graph* $G - i$*, any node* $j \in V \setminus \{i\}$ *can be reached from* r*.*

Lemma 3. *Inequality* $y_i \leq 1$ *for* $i \in V$ *is facet defining if and only if every node in* V *can be reached from* r *and there either exists* $(r, i) \in A$*, or there exist two node disjoint paths between* r *and* i *in* G*. [Proof: see Appendix]*

Given some $k \in V$ and $N \in \mathcal{N}(r,k)$, let us now consider the corresponding node separator inequalities: $y(N) \geq y_k$. Let $S_r \subset V$ denote the subset of nodes that can be reached from r in $G - N$, and let S_k be the remaining nodes, i.e., $S_k = V \setminus (N \cup S_r)$. Then, we have:

Proposition 3 *Given some* $k \in V$ *and* $N \in \mathcal{N}(r,k)$*, the associated node separator inequality* $y(N) \geq y_k$ *is facet defining if* N *is minimal, every node in* V *can be reached from* r *and every node in* S_k *can be reached from* k*.*

Proof. For a given $k \in V$ and $N \in \mathcal{N}(r,k)$, that satisfy the above properties we prove the statement using the indirect method. Let $F(k,N) = \{y \in \{0,1\}^n \mid \sum_{i \in N} y_i = y_k\}$. Consider a facet defining inequality of the form $\mathbf{a}^t\mathbf{y} \geq a_0$. We will show that if all points in $F(k,N)$ satisfy

$$\mathbf{a}^t\mathbf{y} = a_0 \tag{10}$$

then $\mathbf{a}^t\mathbf{y} \geq a_0$ is a positive multiple of (8). Observe first that the zero vector belongs to $F(k,N)$. By plugging it into (10), we get $a_0 = 0$. Consider now an arbitrary node $\ell \in S_r$. Consider a path P from r to ℓ in S_r, and its subpath Q obtained by deleting ℓ. Characteristic vectors of both of them belong to $F(k,N)$, and by subtracting them, we obtain $a_\ell = 0$, for all $\ell \in S_r$. Consider now an arbitrary $\ell \in S_k$. Let P be a path from r to ℓ that passes through exactly one node $i \in N$ and through k. We can find such a path for the following reasons: (i) A path from r to k over a single node $i \in N$ exists because N is minimal. (ii) A path from k to ℓ fully contained in S_k also exists by our assumption. Let Q be a subpath of P obtained by deleting ℓ. Characteristic vectors of P and Q belong to $F(k, N)$, and by subtracting them, we obtain $a_\ell = 0$, for all $\ell \in S_k$. Finally, consider an arbitrary $i \in N$ and a path P' from r to k passing through i and no other nodes from N. Characteristic vector of P' belongs to $F(k, n)$ and after plugging it into (10), we obtain $a_i + a_k = 0$, for all $i \in N$. Therefore, we have $a_i = -a_k = \alpha$, and (10) can be written as $\alpha(y(N) - y_k) = 0$, which concludes the proof. $\square$

4. Computational Results

In this section, we study the computational performance of Branch-and-Cut (B&C) algorithms for the models (SA_r) and (CUT_r) for both the RMWCS and the B-RMWCS.

4.1 Branch-and-Cut Algorithms

Constraint Separation. At each node of the branch-and-bound tree, constraints (2) of the (SA_r) formulation are separated by solving a max-flow problem (see Ljubić et al. [14] for further details). For the (CUT_r) model, inequalities (8) can be separated in polynomial time on an auxiliary support graph G' that splits all nodes except the root into arcs so that each $i \in V$ is replaced by an arc (i_1, i_2). All ingoing arcs into i are now connected to i_1, and all outgoing arcs from i are now connected from i_2. For a given node fractional solution $\tilde{y}$ and $k \in V \setminus (\{r\} \cup D^+(r))$ such that $\tilde{y}_k > 0$, to check whether there are violated inequalities of type (8) we calculate the maximum flow between r and (k_1, k_2) in G' whose arc capacities are defined as $\tilde{y}_i$ for splitted arcs and to zero, otherwise. For both cases, we also use *nested*, *back-flow* and *minimum cardinality* cuts in order to insert as many violated cuts as possible (see Koch and Martin [11], Ljubić et al. [14]). At each separation callback, we limit the number of inserted cuts to 25.

For the B-RMWCS, the cover inequalities (9) are separated by solving a knapsack problem (which is weakly NP-hard) for each fractional solution $\tilde{y}$:

$$(P_{CI}) \qquad \min\{\sum_{i \in V}(1 - \tilde{y}_i)a_i \mid \sum_{i \in V} c_i a_i > B, a_i \in \{0,1\}^n\};$$

if the optimal value of (P_{CI}) is less than one, the nodes $i \in V$ such that $a_i = 1$ are the nodes of a cover V_C for which the corresponding inequality (9) is violated. Finally, once the violated cover inequality is detected, we insert the following *extended cover inequality* in the MIP:

$$\sum_{i \in V_C \cup V^*(C)} y_i \leq |V_C| - 1, \quad \forall V_C \in \mathcal{V}_C \tag{11}$$

where $V^*(C) = \{i \in V \setminus V_C \mid c_i \geq \max_{j \in V_C} c_j\}$. We solve the knapsack problem P_{CI} within the B&C using CPLEX. Only at the root node of the branch-and-bound tree the problem P_{CI} is solved to optimality; in the remaining nodes it is solved until reaching a 0.01% gap.

Primal Heuristic. At a given node of the branch-and-bound tree, we use the information of the current LP solution $\tilde{\mathbf{y}}$ in order to construct feasible primal solutions for the (B-)RMWCS. The procedure, which is equivalent for both (SA_r) and (CUT_r), consists of a (restricted) breadth-first search (BFS) that starts from the root node r and constructs a connected component. A node is incorporated into this component if its weight $\tilde{p}_v := p_v \tilde{y}_v$ is non-negative and its cost c_v added to the cost of the current component does not violate the budget B.

MIP Initialization. As described in §4.2, part of our benchmark set consists of 4-grid graphs. In this case, all 4-cycles are easily enumerated by embedding the grid into the plane and iterating over all faces except for the outer face. Let $\mathcal{C}_4$ be the set of all 4-cycles C such that $r \notin C \cup D^-(C)$ and let $A[C]$ be the set of arcs associated to it. Therefore, in case of 4-grids, the (SA_r) model is initialized with the following *4-cycle inequalities*:

$$z(A[C]) \leq y(C \setminus i), \quad \forall i \in C, \ \forall C \in \mathcal{C}_4. \tag{12}$$

The corresponding 4-cycle inequalities for the (CUT_r) model are:

$$y(D^-(C)) \geq y_i, \quad \forall i \in C, \ \forall C \in \mathcal{C}_4. \tag{13}$$

Additionally, indegree constraints (1) (or (5)) and $z_{ij} + z_{ji} \leq y_i \ \forall e : \{i_{\neq r}, j\} \in E$ are added to the MIP.

Implementation. The B&C algorithms were implemented using CPLEX™12.3 and Concert Technology. All CPLEX parameters were set to their default values, except that: (i) CPLEX cuts, CPLEX heuristics, and CPLEX preprocessing were turned off, and (ii) higher branching priorities were given to $\mathbf{y}$ variables in the case of the (SA_r) model. All the experiments were performed on a Intel Core2 Quad 2.33 GHz machine with 3.25 GB RAM, where each run was performed on a single processor. We denote as "Basic" the B&C implementation for which neither the separation of CI nor the addition of 4-cycle inequalities, (12) or (13), is considered.

4.2 Benchmark Instances

Wildlife Corridor Design Instances. We have considered three real instances provided in [5] that are instances of the corridor design problem for grizzly bears in the Rocky mountains, labeled as `CD-40×40-sq` (242 nodes, 469 edges), `CD-10×10-sq` (3299 nodes, 6509 edges) and `CD-25-hex` (12889 nodes, 38065 edges). In all of them, three reserves are given and the root is chosen as one of them.

We have also considered 4-grid instances generated using the instance generator of Dilkina and Gomes [5]. The description of the parameters used for setting up the instances and the generator itself are available online at [6]. These instances are labeled as `CD-O-C-T` (see [6] for further details). In our experiments we have generated instances with $n + 1 = \mathtt{O}^2$, where $\mathtt{O} \in \{10, 15, 20\}$. We also generated both, correlated and uncorrelated instances ($\mathtt{C} = \{\mathtt{U},\mathtt{W}\}$). Weights and costs are independently and uniformly taken from $\{1, \ldots, 10\}$. We also considered $\mathtt{T} = \{\mathtt{2fR},\mathtt{R}\}$ and, in addition to the root, we consider two more terminals. For each combination of these parameters we have generated 20 instances.

These instances were used for both the RMWCS and the B-RMWCS. For the B-RMWCS, for a given instance `I` with set of terminals R, let $\hat{C}_{min}$ be the cost of the minimum Steiner Tree on R with arc costs $\hat{c}_{ij} = c_j$. Values of the available budget B are defined using slacks over $\hat{C}_{min}$ (see also [5]). For example, a 10% of budget slack corresponds to $B = 1.10 \times \hat{C}_{min}$. For the RMWCS, we redefine weights as $w'_v = p_v - c_v$, which can be done because p_v and c_v have comparable units. That way, w'_v somehow represents the net-profit of including node v into the solution. For the RMWCS we set $R = \emptyset$ and we take as root node the reserve node with the smallest index.

Network Design Instances. These Euclidean instances with a topology similar to street networks are generated as proposed in Johnson et al. [9]: First, n nodes are randomly located in a unit Euclidean square. A link between two nodes i and j is established if the Euclidean distance d_{ij} between them is no more than $\alpha/\sqrt{n}$, for a fixed $\alpha > 0$. For a given n and a given α, weights and costs are independently and uniformly taken from $\{1, \ldots, 10\}$.

We generated instances using $n = \{500, 750, 1000\}$ and $\alpha = \{0.6, 1.0\}$; in case that for a given distribution of n nodes in the plane the value of α is not enough for defining a connected graph, it is increased by 0.01 until connecting all components. For each combination of n and α, 20 instances are generated. We take as root the node with index 0 and when considering a set of terminals, these correponds to those nodes with labels 1 and 2.

4.3 Analyzing the Computational Performance

Results for the B-RMWCS. Table 1 shows a comparison of (SA_r) and (CUT_r) models (including 4-cycle and CI) on the set of corridor design instances. The first three rows correspond to the real instances provided by [5], so for each of them we report statistics over a set of 18 problems (obtained for different budget slacks taken from $\{10, 15, \ldots, 95\}$). For the remaining rows, since we create 20 instances for each parameter setting, the reported values correspond to statistics over $18{\times}20 = 360$ instances. In columns T_{av}(s) and T_{med}(s) we report the average and median running times (in seconds), respectively, of those instances solved to optimality, in columns Gap we show the gaps (as percentages) of those instances that were not solved to optimality within 1800 seconds. Columns #(2) and #(8) show the number of connectivity cuts of the (SA_r) and (CUT_r) model, respectively. Column #NOpt shows the number of instances that are not solved

	SA_r					CUT_r				
Instance	T_{av}(s)	T_{med}(s)	Gap	#(2)	#NOpt	T_{av}(s)	T_{med}(s)	Gap	#(8)	#NOpt
`CD-40×40-sq`	5.28	4.45	0.00	388	0	4.28	3.27	0.00	90	0
`CD-10×10-sq`	619.58	332.40	0.07	1262	10	1389.07	1441.68	1.39	871	14
`CD-25-hex`	–	–	5.17	11524	18	–	–	4.81	2958	18
`CD-10-U-2fR`	1.67	1.12	–	527	0	2.71	1.82	–	360	0
`CD-10-W-2fR`	1.80	1.00	–	535	0	2.22	1.50	–	389	0
`CD-10-U-3R`	0.91	0.71	–	362	0	0.63	0.38	–	157	0
`CD-10-W-3R`	3.08	0.50	–	389	0	0.82	0.42	–	190	0
`CD-15-U-2fR`	12.47	7.71	–	1085	0	26.33	13.78	–	883	0
`CD-15-W-2fR`	12.40	8.08	–	1222	0	26.61	10.98	–	1071	0
`CD-15-U-3R`	4.56	2.98	–	814	0	7.84	2.81	–	513	0
`CD-15-W-3R`	4.86	2.88	–	809	0	7.34	3.24	–	539	0

Table 1. Computational performance on B-RMWCS (+C4+CI) instances from [5].

to optimality within 1800 seconds. We observe that for all 4-grid instances, except for the `CD-10×10-sq` graph for which a more detailed analysis is given below, both approaches are able to solve all instances in more or less reasonable times, although the (SA_r) model is slightly better than the (CUT_r) model. On the other hand, the number of inserted violated cuts of the (CUT_r) model is in all of the cases significantly smaller than the corresponding number for the (SA_r) model. The efficacy of the (SA_r) model can be explained by the sparsity of 4-grid graphs. On the contrary, for the only more dense instance of this group, namely `CD-25-hex`, which is a 6-grid with 12889 nodes and 38065 edges, the (CUT_r) model performs better than the (SA_r) model. More precisely, the avg. gap and its standard deviation for the (SA_r) model are 5.17% and 1.11%, resp., while for the (CUT_r) model these values are 4.81% and 0.81%, resp.'

To analyze the effects of special inequalities, namely 4-cycle and CI, we compare three approaches: Basic, Basic plus 4-cycle inequalities (denoted by "+C4") and Basic plus 4-cycle and CI (denoted by "+C4+CI"). In Figure 2 we present the box-plots of the gaps attained within 1800 seconds when solving real instance `CD-10×10-sq` for budget slacks taken from $\{10, 15, \ldots, 95\}$. The values marked with an asterisk and $\times$ correspond to the mean and maximum running time, respectively. Below the bottom of each box the number of instances solved to optimality is indicated, and next to "#Cuts:" we report the average number of detected cuts of type (2) and (8), respectively.

The box-plots indicate that for the Basic setting the (CUT_r) model significantly outperforms the (SA_r) model on this instance, in terms of the quality of the solutions (smaller gaps), the stability of the approach (smaller dispersion), and the number of instances solved to optimality. This is mainly due to the fact that in the (CUT_r) model there are less variables, so the optimization becomes easier and more stable. However, when including 4-cycle inequalities, although both approaches perform better, (SA_r) now outperforms (CUT_r). The average number of inserted cuts of type (2) decreases from 5989 to 1264 when 4-cycle inequalities are added, while for the (CUT_r) model this reduction is more attenuated (only 18%). This means that for this instance constraints (12) are empirically more effective than (13) in reducing too frequent calls of the maximum flow procedure. When adding the separation of CI ("+CI") we observe that

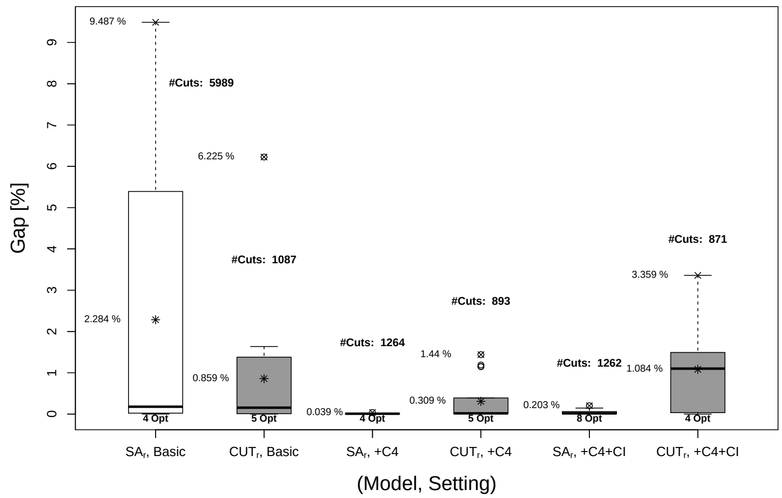

Fig. 2. Box-plots of the gaps [%] reached within 1800 sec for the `CD-10×10-sq` instance considering (SA_r) and (CUT_r) and three different settings of the B&C (Budget slack [%] taken from $\{10, 15, \ldots, 95\}$).

these constraints are more beneficial for the (SA_r) model than for the (CUT_r) model - the latter one even slows down with addition of these cuts. This can be explained by some numerical instability that can appear when dealing with the separation of CI. We conclude that the advantage of the (CUT_r) model of having less variables vanishes when more sophisticated ideas are considered.

For the Network Design instances (whose complete results are not reported due to space limitation), the graph density plays a role in the performance of the two models. For instance, for $n = \{500, 750\}$ and $\alpha = 0.6$, the (SA_r) model solves 536 instances out of 760 within the time limit, while the (CUT_r) model solves 443. However, when $\alpha = 1.0$, the (SA_r) approach solves 483 while the (CUT_r) approach solves 502. In both cases, the average running times of the (CUT_r) model needed to prove optimality are smaller than those of the (SA_r) model.

Results for the RMWCS. For the RMWCS we have considered the same corridor design instances and, in addition, the network design instances with a weight transformation as described in § 4.2. In Table 2, equivalent to Table 1, we report the results obtained for the corridor design instances. In this case, time limit is set to 3600 seconds. We observe that the (CUT_r) model outperforms the (SA_r) model on real instances, and on random lattices it is the other way around, although the differences are less visible.

The results on the network design instances are reported in Table 3. For a given n and α equal to 0.6 and 1.0, respectively, column #nodes shows $n + 1$ and column #edges shows the average number of edges for a set of 20 instances created using this setting. All instances of this group were solved to optimality,

	SA_r				CUT_r			
Instance	Time(sec)	Gap(%)	#(2)	#NOpt	Time(sec)	Gap(%)	#(8)	#NOpt
`CD-40×40-sq`	0.70	–	254	0	0.16	–	10	0
`CD-10×10-sq`	316.11	–	3998	0	88.70	–	60	0
`CD-25-hex`	3600.00	1.99	20304	1	2611.13	–	14756	0
`CD-10-U-2fR`	0.15	–	231	0	0.14	–	34	0
`CD-10-W-2fR`	0.14	–	239	0	0.18	–	40	0
`CD-10-U-3R`	0.13	–	226	0	0.13	–	28	0
`CD-10-W-3R`	0.15	–	241	0	0.12	–	26	0
`CD-15-U-2fR`	1.28	–	720	0	11.59	–	99	0
`CD-15-W-2fR`	1.35	–	755	0	3.66	–	94	0
`CD-15-U-3R`	1.24	–	763	0	2.02	–	73	0
`CD-15-W-3R`	1.45	–	809	0	2.26	–	78	0
`CD-20-U-2fR`	7.67	–	1618	0	166.32	–	223	0
`CD-20-W-2fR`	7.41	–	1615	0	74.46	–	234	0
`CD-20-U-3R`	7.57	–	1667	0	16.90	–	133	0
`CD-20-W-3R`	8.39	–	1765	0	86.18	–	195	0

Table 2. Computational performance on instances from [5] when solving the RMWCS.

		SA_r		CUT_r	
#nodes	#edges	Time(sec)	#(2)	Time(sec)	#(8)
500	2535	11.42	1218	2.29	22.8
500	6484	3.50	211	0.84	<10
750	3845	57.07	2541	5.67	25.8
750	9944	7.69	287	1.71	<10
1000	5180	97.41	3188	15.59	36.3
1000	13397	10.16	302	2.77	<10

Table 3. Computational performance on the RMWCS network design instances.

therefore in Table 3 we only report the average running times and the average number of detected connectivity cuts. For these instances, the (CUT_r) approach clearly outperforms the (SA_r) approach; for these instances, the ratio between the number of edges and the number of nodes is, depending on the value of α, around 5 or 13, in contrast to the corridor design instances, where this ratio is close to two. This characteristic implies a practical difficulty for the (SA_r) model due to the increase of the number of variables. Besides, for this group of instances, 4-cycle constraints and CI cannot be used in the initialization.

Conclusion. The obtained computational results let us conclude that both models (CUT_r) and (SA_r) perform very well in practice, and that their performance is complementary. Using the (CUT_r) model (i.e., having less variables) pays off for denser graphs with many zero-weight nodes for both, B-RMWCS and RMWCS.

Acknowledgement We are thankful to Bistra Dilkina from the Department of Computer Science, Cornell University, who helped in the understanding and interpretation of the Wildlife Corridor Design instances considered in this paper. This research is partially conducted during the research stay of Ivana Ljubić at the TU Dortmund, supported by the APART Fellowship of the Austrian Academy of Sciences. This support is greatly acknowledged. Eduardo Álvarez-Miranda thanks the Institute of Advanced Studies of the Università di Bologna from where he is a PhD Fellow.

Bibliography

[1] C. Backes, A. Rurainski, G. Klau, O. Müller, D. Stöckel, A. Gerasch, J. Küntzer, D. Maisel, N. Ludwig, M. Hein, A. Keller, H. Burtscher, M. Kaufmann, E. Meese, and H. Lenhof. An integer linear programming approach for finding deregulated subgraphs in regulatory networks. *Nucleic Acids Research*, 1:1–13, 2011.

[2] R. Carvajal, M. Constantino, M. Goycoolea, J.P. Vielma, and A. Weintraub. Imposing connectivity constraints in forest planning models. *Submitted*, 2011.

[3] S. Chen, I. Ljubić, and S. Raghavan. The generalized regenerator location problem. *Submitted*, 2012.

[4] J.M. Conrad, C.P. Gomes, W.-J. van Hoeve, A. Sabharwal, and J.F. Suter. Wildlife corridors as a connected subgraph problem. *Journal of Environmental Economics and Management*, 63(1):1–18, 2012.

[5] B. Dilkina and C. Gomes. Solving connected subgraph problems in wildlife conservation. In A. Lodi, M. Milano, and P. Toth, editors, *CPAIOR*, volume 6140 of *LNCS*, pages 102–116. Springer, 2010.

[6] B. Dilkina and C. Gomes. Synthetic corridor problem generator, 2012. URL `http://www.cs.cornell.edu/~bistra/connectedsubgraph.htm`.

[7] M. Dittrich, G. Klau, A. Rosenwald, T. Dandekar, and T. Müller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, 24:i223–i231, 2008.

[8] A. Fügenschuh and M. Fügenschuh. Integer linear programming models for topology optimization in sheet metal design. *Mathematical Methods of Operations Research*, 68(2):313–331, 2008.

[9] D. S. Johnson, M. Minkoff, and S. Phillips. The prize-collecting Steiner tree problem: Theory and practice. *Proc. 11th ACM-SIAM Symp. Discrete Algorithms, 9-11 January 2000, San Francisco, USA*, SODA 2000:760–769, 2000.

[10] K. Kaparis and A.N. Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical Programming*, 124(1-2):69–91, 2010.

[11] T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32:207–232, 1998.

[12] H. Lee and D. Dooly. Decomposition algorithms for the maximum-weight connected graph problem. *Naval Research Logistics*, 45:817–837, 1998.

[13] I. Ljubić. *Exact and Memetic Algorithms for Two Network Design Problems*. PhD thesis, Vienna University of Technology, 2004.

[14] I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical Programming*, Series B 105:427?–449, 2006.

[15] A. Moss and Y. Rabani. Approximation algorithms for constrained node weighted Steiner tree problems. *SIAM J. Comput.*, 37(2):460–481, 2007.