

The Generalized Reserve Set Covering Problem with Connectivity and Buffer Requirements

Eduardo Álvarez-Miranda ^{*1}, Marcos Goycoolea ^{†3}, Ivana Ljubić ^{‡4}, and Markus Sinnl ^{§2}

¹*Department of Industrial Engineering, Universidad de Talca, Curicó, Chile*

³*School of Business, Universidad Adolfo Ibañez, Santiago, Chile*

⁴*ESSEC Business School of Paris, France*

²*Department of Statistics and Operations Research, Faculty of Business, Economics and Statistics, University of Vienna, Vienna, Austria*

Abstract

The design of nature reserves is becoming, more and more, a crucial task for ensuring the conservation of endangered wildlife. In order to guarantee the preservation of species and a general ecological functioning, the designed reserves must typically verify a series of spatial requirements. Among the required characteristics, practitioners and researchers have pointed out two crucial aspects: (i) connectivity, so as to avoid spatial fragmentation, and (ii) the design of buffer zones surrounding (or protecting) so-called core areas.

In this paper, we introduce the *Generalized Reserve Set Covering Problem with Connectivity and Buffer Requirements*. This problem extends the classical *Reserve Set Covering Problem* and allows to address these two requirements simultaneously. A solution framework based on Integer Linear Programming and branch-and-cut is developed. The framework is enhanced by valid inequalities, a construction and a primal heuristic and local branching.

An extensive computational study on grid-graph instances and real-life instances based on data from three states of the U.S. and one region of Australia is carried out to assess the suitability of the proposed model to deal with the challenges faced by decision-makers in natural reserve design. The results show, on the one hand, the flexibility of the proposed models to provide solutions according to the decision-makers' requirements, and on the other hand, the effectiveness of the devised algorithm for providing good solutions in reasonable computing times.

1 Introduction and Motivation

Demographic expansion, natural resource exploitation, and the consequences of climate change, are among the processes that had resulted in a dramatic loss of biodiversity in the last decades. In July 2012, at the Rio+20 Earth Summit, the International Union for Conservation of Nature revealed that about 20,000 species are threatened with extinction. Among them, 4,000 are described as *critically endangered* and 6,000 as *endangered*, while more than 10,000 species are listed as *vulnerable* (IUCN 2016).

The maintenance of biodiversity is crucial for the humankind, so its preservation is decisive for future generations (Cardinale et al. 2012). As a matter of fact, immense efforts have been devoted in the last decades by international organizations, governments, and foundations, for the establishment of protected

*ealvarez@utalca.cl

†marcos.goycoolea@uai.cl

‡ivana.ljubic@essec.edu

§markus.sinnl@univie.ac.at

areas aiming at ensuring a sustainable landscape for wildlife. The reader is referred to the books (Gergel and Turner 2002, Lindenmayer and Franklin 2002, Millsaugh and Thompson 2008) for in-depth analyses and discussion of motivations, models, cases, and challenges in the field of wildlife conservation and nature reserve planning.

The design of such nature reserves has led to the development of a plethora of mathematical models. The purpose of such models is in providing optimally designed reserves that respect ecological, economical and eventually other requirements (see Billionnet (2013) for a recent comprehensive review on optimization models for biodiversity conservation). In its most fundamental form, a nature reserve design problem can be stated as follows: one is given a set V of land sites (also known as land units or parcels) eligible for selection, a set of species or features S , and, for each species $s \in S$, a set of suitable land sites $V_s \subseteq V$. The goal is to find the *minimum* number of reserve sites such that each species is present in the selected set of sites at least once. This problem is known as the *Reserve Set Covering Problem* (RSC) (Church et al. 1996, Pressey et al. 1997).

The RSC can be formulated as an Integer Linear Programming (ILP) problem; let $\mathbf{x} \in \{0, 1\}^{|V|}$ be a vector of binary variables such that $x_i = 1$ if site $i \in V$ is selected, and $x_i = 0$ otherwise. Using this notation, the model

$$\begin{aligned}
 \text{(RSC)} \quad \min \pi(\mathbf{x}) &= \sum_{i \in V} x_i && \text{(CARD)} \\
 \text{s.t.} \quad \sum_{i \in V_s} x_i &\geq 1, \quad \forall s \in S && \text{(COV)} \\
 \mathbf{x} &\in \{0, 1\}^{|V|}, && \text{(BIN-}\mathbf{x}\text{)}
 \end{aligned}$$

allows to find the optimal (minimum size) reserve given by $V^* = \{i \in V \mid x_i^* = 1\}$. Observe that in the RSC, we are only given a set of land sites V . To work with spatial requirements, such as connectivity and buffer zones, one is also given a graph $G = (V, E)$, in which the set of edges E encodes the spatial relationship of the land sites (e.g., $\{i, j\} \in E$, if and only if i and j from V share a common border). In the remainder of the paper, the terms *land site* and *node* will be used interchangeably.

The use of mathematical optimization models, such as (CARD)-(BIN- \mathbf{x}), for the optimal design of nature reserves is a widely explored research area; some references covering two decades of work are found in (Beyer et al. 2016, Cayton et al. 2017b, Clemens et al. 1999, Öhman and Lämås 2005, Önal and Briers 2003, Williams 2008, Williams and ReVelle 1996, 1998, Williams et al. 2004). A natural alternative to this problem, is to find a set of exactly p parcels that *maximize* the number of protected species; this problem is known as the Maximal Covering Species Problem (Church et al. 1996).

Although the RSC enables decision makers to gain important insights about the suitable sites that need to be preserved, the obtained solutions typically fail in satisfying relevant spatial attributes. According to the reviews presented in (Billionnet 2013) and (Williams et al. 2005), the spatial requirements that are typically imposed when designing reserves can be classified as follows: (i) reserve size or *compactness*, (ii) *number* of reserves, (iii) reserves *proximity*, (iv) reserve *connectivity*, (v) reserve *shape*, and (iv) presence of *core* and *buffer* areas. As pointed out in the above mentioned reviews (and the references therein), these requirements correspond to different ecological needs which depend on the considered landscape, the species to be protected, and the human activities surrounding the potential reserve. Fundamental works on the design of reserves respecting spatial requirements can be found, for example, in (Önal and Briers 2002, Schwartz 1999).

Among relevant spacial requirements, connectivity is one of the most prevalent ones; it avoids habitat fragmentation improving the conditions for sustainable ecosystems (Beier and Noss 1998, Debinski and Holt 2000, Gergel and Turner 2002, Millsaugh and Thompson 2008). Various modeling approaches have been proposed for incorporating connectivity into the design of nature reserves (see, e.g., (Beyer et al. 2016, Billionnet 2012, Dilkina and Gomes 2010, Jafari and Hearne 2013, Jafari et al. 2017, Öhman and Lämås 2005, Önal et al. 2016, St. John et al. 2018, Wang and Önal 2011)). Note that ensuring connectivity does not necessarily mean that only a single reserve has to be designed. Instead, multiple connected components may be allowed too.

The second important criterion is the presence of core areas and buffer zones that allows the development of so-called *Biosphere* reserves (Batisse 1982, 1990). The purpose of the buffer zone is to surround the core area and protect it from negative external impacts, therefore promoting the long-term viability of critical species (see, e.g., Chapters 1 and 5 of Lindenmayer and Franklin (2002) and Chapters 19 and 20 of Millsbaugh and Thompson (2008)).

Finally, another important issue in reserve design is to ensure *minimum quotas of ecological suitability* for some species, especially critically endangered ones (see, e.g., Chapters 9 and 14 of Millsbaugh and Thompson (2008)).

Contribution and Outline of the Paper Although the design of nature reserves considering buffer zone requirements has been addressed before (see, e.g., Clemens et al. (1999), Williams and ReVelle (1996, 1998)), the question on how to impose connectivity requirements to the buffer zones remained unanswered in the existing literature. In particular, in (Billionnet 2013), the authors point out that known models fail in providing suitable conditions for particular endangered species precisely due to the lack of connectivity of the resulting reserves. The main contribution of our work consists thus of providing, for the first time, a modeling and algorithmic framework for the optimal design of wildlife reserves by simultaneously integrating three criteria: connectivity requirements, construction of buffer zones and minimum quotas of ecological suitability. This is done by introducing the *Generalized Reserve Set Covering Problem with Connectivity and Buffer Requirements (GRSC-CB)*. The GRSC-CB allows to design a reserve comprised by one or more connected components; each of them consisting of a core surrounded by a buffer zone. In our extensive numerical study on grid-graph and real-life instances, we demonstrate that the GRSC-CB, and its variants, deliver solutions that properly embody different spatial and ecological features and significantly improve upon the spacial structure of reserves created by the simpler RSC models from the literature.

The article is organized as follows. In Section 2 we incrementally develop an ILP-formulation for the GRSC-CB by first giving a generalization of the RSC without considering connectivity and buffer requirements, and then introducing constraints to model these two requirements. In Section 3, we describe a branch-and-cut framework to solve the proposed formulation. Computational results on synthetic instances, as well as case studies on real-life instances based on data of the U.S. National Gap Analysis Program and of the Northern Australia Water Futures Assessment Program are presented in Section 4. Section 5 concludes the paper.

2 The Generalized Reserve Set Covering Problem with Connectivity and Buffer Requirements

In this section, we first give a generalization of the RSC, which takes into account that in real-life situations, different types of species need different levels of protection. The resulting problem is denoted as *Generalized Reserve Set Covering Problem (GRSC)*. We then introduce constraints for modeling buffer and connectivity requirements, and finally obtain the GRSC-CB. These constraints can also be added individually to obtain problems that we denote as *GRSC with buffer requirements (GRSC-B)* and *GRSC with connectivity requirements (GRSC-C)*. In Figure 1 optimal solutions for these four problems on the same underlying instance are shown. The influence of the spatial constraints imposed in the different problem variants can be easily seen. The GRSC solution is very fragmented. The GRSC-B solution has a more compact shape, but consists of two components, and one of these components is very small. The GRSC-C solution is connected, but its spatial distribution lacks compactness. Finally, the GRSC-CB solution has a similar compact shape as the GRSC-B solution, but consists of a single component only.

To account for the difference concerning the need of protection among species, we consider a partition of the set of species S into two subsets, namely S_1 and S_2 , $S = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$. Species in S_1 are the ones needing stronger protection (e.g., the land sites selected for hosting them need to be in the *core* of the designed reserve, if a model with core and buffer zones is considered).

Additionally, ecologists usually do not only know in which land sites a species lives, but they are able

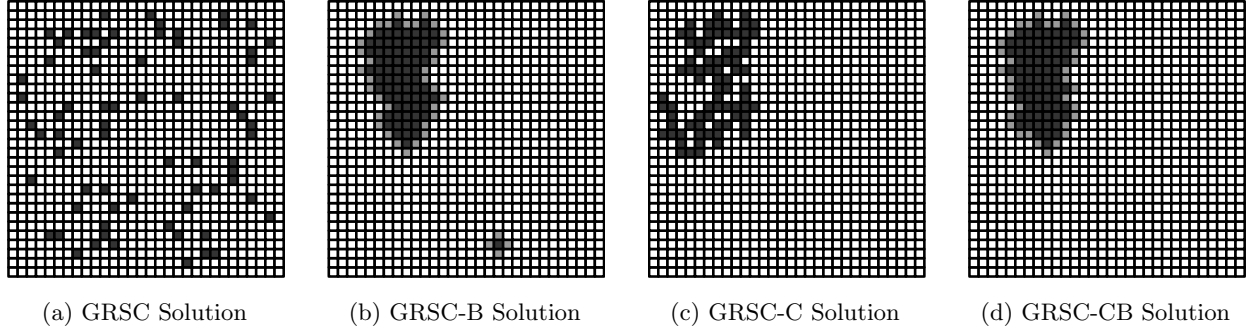


Figure 1: Optimal solutions on the same instance for the different problem variants introduced in the paper. For problems with buffer, i.e., GRSC-B and GRSC-CB, \blacksquare denotes core land sites, and \square denotes buffer land sites; for the problem without buffer, i.e., GRSC and GRSC-C, \blacksquare denotes selected land sites.

to assess the habitat features of the different land sites. We model this by using a habitat *suitability* score function $\mathbf{w} : V \times S \rightarrow \mathbb{R}_{\geq 0}$, such that w_i^s measures how advisable, with respect to species $s \in S$, it is to select the land site $i \in V$ as part of the reserve (see, e.g., Chapters 9 and 10 of Millspaugh and Thompson (2008)). A species $s \in S$ is considered to be protected by the designed reserve, if the w_i^s values of the land parcels contained in the reserve sum up to at least $\lambda_s \geq 0$, which represents a *minimum quota of ecological suitability* for species s . If a model with buffer requirements is used, for species $s \in S_1$, the w_i^s values of land parcels selected for the core of the designed reserve must sum up to at least λ_s (as only core area is suitable to protect the species in S_1). We denote with V_s the set of land parcels with $w_i^s > 0$. We observe that a simpler version of this suitability approach has already been used previously, e.g., in Polasky et al. (2001) the constraint (COV) is replaced by $\sum_{i \in V_s} x_i \geq h_s, \forall s \in S$, where $h_s \in \mathbb{Z}_{\geq 1}$ is the number of sites required by species $s \in S$. Finally, there is a cost function $\mathbf{c} : V \rightarrow \mathbb{R}_{> 0}$, such that c_i corresponds to the cost of selecting the land site $i \in V$ as part of the reserve (see, e.g., Chapter 5 of Millspaugh and Thompson (2008) for further details regarding economic considerations for planning wildlife conservation).

In the typical reserve design setting considered in the literature, all species under consideration must be hosted within the reserve. However, if the area under consideration hosts many different species, the obtained reserves might be impractical due to size considerations or undesirable shapes. To allow for more flexibility in the design, we introduce two parameters, $0 \leq P_1 \leq |S_1|$ and $0 \leq P_2 \leq |S_2|$, allowing the decision maker to specify for how many of the species in S_1 and S_2 the obtained reserve must fulfill the minimum quota of ecological suitability. We observe that the classical RSC described in the introduction can be obtained by setting $S_2 = S$, $w = \mathbf{1}$ and $\lambda_s = 1$ for every species s , and $P_2 = |S_2|$.

Definition 1 *Given the input data described above, the GRSC-CB is the problem of finding a minimum cost natural reserve that fulfills the following criteria:*

- (i) *at least P_1 species from S_1 are hosted in the core area,*
- (ii) *at least P_2 species from S_2 are hosted in the reserve,*
- (iii) *the minimum ecological suitability quotas for each of the hosted species are satisfied,*
- (iv) *the solution is comprised by at most k connected areas, and*
- (v) *each connected area consists of a core surrounded by a buffer of width d .*

In this article we also address three relaxations of this problem, namely: GRSC, in which the conditions (iv) and (v) are relaxed; GRSC-B, in which condition (iv) is relaxed, and GRSC-C, in which condition (v) is relaxed (i.e., $d = 0$).

In the following, we introduce the ILP models for all four variants in a modular fashion.

2.1 Modeling the GRSC Problem

In our formulation, vector $\mathbf{x} \in \{0, 1\}^{|V|}$ is associated with the decision of selecting sites as part of the reserve (either as part of the core or part of the buffer zone). In addition, let vector $\mathbf{z} \in \{0, 1\}^{|V|}$ be associated with the decision of selecting sites as part of the core area. Let $\mathbf{u} \in \{0, 1\}^{|S|}$ be a vector of variables so that $u_s = 1$ if species $s \in S$ is hosted by the reserve, and $u_s = 0$, otherwise. A triplet $(\mathbf{u}, \mathbf{x}, \mathbf{z})$ is associated with an appropriate territorial coverage of the species if the following inequalities hold:

$$\sum_{i \in V_s} w_i^s z_i \geq \lambda_s u_s, \quad \forall s \in S_1 \quad (S_1\text{-SQ})$$

$$\sum_{i \in V_s} w_i^s x_i \geq \lambda_s u_s, \quad \forall s \in S_2 \quad (S_2\text{-SQ})$$

$$\sum_{s \in S_1} u_s \geq P_1 \quad (S_1\text{-PROTECT})$$

$$\sum_{s \in S_2} u_s \geq P_2. \quad (S_2\text{-PROTECT})$$

Constraints $(S_1\text{-SQ})$ ensure that if a species $s \in S_1$ is hosted by the reserve ($u_s = 1$), then the ecological suitability of the core of the reserve with respect to s must be at least λ_s . Likewise, constraints $(S_2\text{-SQ})$ ensure that if a species $s \in S_2$ is protected ($u_s = 1$), then the ecological suitability of the reserve must be at least λ_s . These two set of constraints will be referred to as *suitability quota constraints*. Note that if a planner wants to ensure that a given species s *must* be part of the reserve, then she/he can achieve this by simply adding the constraint $u_s = 1$. Constraints $(S_1\text{-PROTECT})$ and $(S_2\text{-PROTECT})$ imposes that at least P_1 core species and P_2 buffer species must be protected. These two constraints will be referred to as *species protection constraints*.

The cost of the reserve is calculated as the sum of the cost of all of the selected sites. To model this, we need the following set of linking constraints

$$z_i \leq x_i, \quad \forall i \in V, \quad (\text{LINK})$$

i.e., if a site is considered to be in the core, the site must be in the designed reserve. The cost of a reserve encoded by $(\mathbf{u}, \mathbf{x}, \mathbf{z})$ is then given by

$$\gamma(\mathbf{u}, \mathbf{x}, \mathbf{z}) = \sum_{i \in V} c_i x_i. \quad (\text{COST})$$

The inequalities presented so far allow to formulate a first generalization of the RSC, denoted as *Generalized Reserve Set Covering Problem (GRSC)*, given by

$$(\text{GRSC}) \quad \min \left\{ \gamma(\mathbf{u}, \mathbf{x}, \mathbf{z}) \mid (S_1\text{-SQ}), (S_2\text{-SQ}), (S_1\text{-PROTECT}), (S_2\text{-PROTECT}), (\text{LINK}), (\mathbf{u}, \mathbf{x}, \mathbf{z}) \in \{0, 1\}^{|S|+2|V|} \right\}.$$

The GRSC takes into account the minimum suitability quotas, but it ensures neither the connectivity of the reserve, nor the existence of a buffer around the core. This means that in particular situations, like the one depicted in Figure 1a, very fragmented reserves could be obtained.

In the following, these two missing properties are characterized and, thereafter, the complete ILP formulation for GRSC-CB is presented. (We observe that the GRSC could be modeled only using (\mathbf{x}, \mathbf{u}) ; \mathbf{z} will be used in the following for modeling the core/buffer interplay.)

2.2 Modeling the Buffer Zone

To model the buffer surrounding the selected core areas, the following definition of *d-neighborhood set of a node i* is needed.

Definition 2 (*d-neighborhood*) For a given integer $d \geq 0$ and a given land site $i \in V$, the *d-neighborhood* of i , $\delta_d(i)$, is defined as

$$\delta_d(i) = \{j \in V_{i \neq j} \mid \text{the minimum number of hops between } i \text{ and } j \text{ is at most } d\}.$$

We also define the set $\delta_d^+(i) = \delta_d(i) \cup \{i\}$.

Set $\delta_d(i)$ corresponds to the set of all nodes $j \in V$ that are separated by at most d edges from i , e.g., all adjacent nodes of i are given by $\delta_1(i)$ (for sake of readability, we will write $\delta_1(i) = \delta(i)$ in the following). As the graph in our setting is undirected, for two nodes i, j , we have $i \in \delta_d(j)$ if and only if $j \in \delta_d(i)$. The definition allows to model the buffer zones with the following set of constraints

$$z_i \leq x_j, \forall j \in \delta_d(i), \forall i \in V \quad (d\text{-BUFF.1})$$

i.e., if i is taken as part of the core ($z_i = 1$), then all of the land sites $j \in \delta_d(i)$ must verify $x_j = 1$ (i.e., they must at least be part of the buffer zone). Hence, it ensures that the each core land site is surrounded by *at least* d other sites within the reserve. However, using only constraints (d-BUFF.1), it is possible, that some land parcel i is selected (i.e., $x_i = 1$ and $z_i = 0$) without any parcel j forcing it to be in the solution as a buffer side for j . This can happen in presence of constraints (S_2 -SQ) and (S_2 -PROTECT) (without these constraints, the objective function together with $c_i > 0, i \in V$ takes care of the issue). This situation is not desired, as it can lead to a fragmented reserve. To deal with this issue, we propose the following set of inequalities;

$$x_i \leq \sum_{j \in \delta_d(i)} z_j, \forall i \in V. \quad (d\text{-BUFF.2})$$

Inequalities (d-BUFF.2) ensure that whenever some $x_i = 1$, at least one core node z_j is selected in the d -neighborhood. Therefore, combining (d-BUFF.1) and (d-BUFF.2) leads to cores that are *nested* within buffer zones of width d . Adding constraints (d-BUFF.1) and (d-BUFF.2) to GRSC leads to the *GRSC with Buffer Requirements (GRSC-B)*.

2.3 Modeling Connectivity of the Reserve

Connectivity is modeled using a variant of a concept called *node-separators* (see, e.g., Álvarez-Miranda et al. (2013a,b), Carvajal et al. (2013), Fischetti et al. (2017)). For modeling purposes, let $G_r = (V_r, E_r)$ with $V_r = V \cup \{r\}$ and $E_r = E \cup \{(r, i) \mid i \in V\}$, i.e., G_r corresponds to the original graph extended by an artificial root r and $|V|$ additional directed arcs connecting r with every node in V (denoted r -arcs, A_r). For a given site ℓ from V , a tuple $W = (W_V, W_A)$, $W_V \subseteq V$ and $W_A \subseteq A_r$, is called an *r-arc-node-separator* if and only if after removing W from G_r , site ℓ cannot be reached from r . For a given node ℓ from V , let \mathcal{W}_ℓ be the set of all r -arc-node-separators with respect to ℓ .

Let $\mathbf{y} \in \{0, 1\}^{|V|}$ be a vector of auxiliary variables, such that $y_i = 1$ if $i \in V$ is connected with r through arc (r, i) , and $y_i = 0$ otherwise. Using these variables, we model connectivity (of both the core area and the buffer area) by ensuring that in the obtained solution, there is a path from r to all core nodes and all buffer nodes selected in the solution (i.e., nodes with $z_i = 1$, resp., $x_i = 1$). This means, the obtained feasible solutions are comprised of connected components rooted at nodes with $y_i = 1$. Thus, the number of connected components within the reserve can be controlled with the constraint

$$\sum_{j \in V} y_j \leq k. \quad (\text{NCOMP})$$

Constraint (NCOMP) can also be written in equality form, if desired by a decision maker. Moreover, if a land parcel is connected to the root (i.e., $y_i = 1$) the land parcel must be taken in the core; this ensures that every connected component in the solution has at least one core land parcel. This linking is done with the following set of constraints:

$$y_i \leq z_i, \forall i \in V. \quad (\text{YZ})$$

The connectivity inequalities follow easily from the definition of the r -arc-node-separators: If a node ℓ is in the solution, then for each separator in $W = (W_V, W_A) \in \mathcal{W}_\ell$, at least one element (i.e., node from W_V or arc from W_A) must be taken. We obtain the following family of inequalities for connectivity of the selected core sites

$$\sum_{i \in W_V} z_i + \sum_{j \in W_A} y_j \geq z_\ell, \forall W \in \mathcal{W}_\ell, \forall \ell \in V; \quad (\text{CORECON})$$

inequalities (CORECON) are denoted as *connectivity cuts*. They are exponential in number, and the resulting ILP is tackled by means of branch-and-cut (see Section 3 for a separation procedure for these inequalities). We observe that these cuts can be down-lifted by not considering all $j \in W_A$, but only $j \leq \ell$, i.e., if node ℓ is in the solution, the component containing ℓ must be rooted either at ℓ , or a node with index smaller than ℓ . By using this down-lifted variant, all k components of a feasible solution must be rooted at the node with minimal index within each component, e.g., symmetric solutions giving the same components, but rooted at other nodes, are not allowed.

When using inequalities (d -BUFF.1) and (d -BUFF.2) together with the inequalities proposed in this section, connectivity of the land parcels selected by $x_i = 1$ is automatically ensured, since all parcels with $x_i = 1$ and $z_i = 0$ form a buffer of thickness d around all i with $z_i = 1$. The reserve induced by $z_i = 1$ form (at most) k connected components and the buffer around them is connected by construction (as argued in the previous section).

If a decision maker does not want a buffer-zone, but only a reserve comprised of (at most) k connected components, connectivity cuts must be written in x instead of z , i.e.,

$$\sum_{i \in W_V} x_i + \sum_{j \in W_A} y_j \geq x_\ell, \forall W \in \mathcal{W}_\ell, \forall \ell \in V; \quad (\text{ALLCON})$$

and also the linking-constraints (LINK) should be replaced by

$$y_i \leq x_i, \forall i \in V. \quad (\text{YX})$$

We define the *GRSC with Connectivity Requirements (GRSC-C)* as the problem obtained by adding (ALLCON), (YX) and (NCOMP) to GRSC.

2.4 The Complete Model

We are now ready to give the ILP model for the *Generalized Reserve Set Covering Problem with Connectivity and Buffer Requirements (GRSC-CB)*:

$$\begin{aligned} (\text{GRSC-CB}) \quad \min \left\{ \gamma(\mathbf{u}, \mathbf{x}, \mathbf{z}) \right. & \left. \middle| (S_1\text{-SQ}), (S_2\text{-SQ}), (S_1\text{-PROTECT}), (S_2\text{-PROTECT}), (\text{LINK}), \right. \\ & \left. (d\text{-BUFF.1}), (d\text{-BUFF.2}), (\text{CORECON}), (\text{YZ}), (\text{NCOMP}), (\mathbf{u}, \mathbf{x}, \mathbf{z}, \mathbf{y}) \in \{0, 1\}^{|S|+3|V|} \right\}. \end{aligned}$$

In order to solve (GRSC-CB), we propose in the following an algorithmic framework based on branch-and-cut. Besides dealing with the separation of inequalities which are exponential in size, this framework also integrates multiple heuristics that allow to find high-quality solutions for large-scale instances, for which the optimal solution cannot be obtained.

3 A Branch-and-Cut Framework for the GRSC-CB

The core of our branch-and-cut framework is based on the separation of the connectivity cuts (CORECON). In addition, to improve the quality of lower bounds, we propose additional valid inequalities for the model (GRSC-CB), denoted as *species cuts*, *cover inequalities*, and *species-cover cuts*. These inequalities (which

are described in Section 3.1) are dynamically separated, together with (CORECON), and the underlying separation routines are given in Section 3.2. In order to find feasible solutions within the framework, we also designed heuristics that are detailed in Section 3.3.

3.1 Valid Inequalities

Species Cuts If a species $s \in S_1$ is in the solution, i.e., $u_s = 1$, then there must be a path from the root node to at least one land parcel in V_s (which are the nodes with $w_i^s > 0$). To enforce this, we again use connectivity cuts based on r -arc-node separators. The cuts are defined using a graph G_r^s , which is an extension of G_r with an additional node s . Let $G_r^s = (V_r^s, E_r^s)$ with $V_r^s = V_r \cup \{s\}$ and $E_r^s = E_s \cup \{(i, s) \mid i \in V_s\}$, i.e., G_r^s corresponds to G_r plus an artificial sink node s and $|V_s|$ additional directed arcs connecting every node in V_s with s (denoted as A_s). For a given species $s \in S_1$, let \mathcal{W}_s be the set of all r -arc-node-separator with respect to s . The species cuts for a given species s read as follows

$$\sum_{i \in W_V} z_i + \sum_{j \in W_A} y_j \geq u_s, \forall W \in \mathcal{W}_s, s \in S_1. \quad (\text{SC})$$

Cover Inequalities and Species-cover Cuts Let $W_s = \sum_{i \in V} w_i^s$. For a given species $s \in S$, a set $C_s \subset V_s$ is called a *cover* if $\sum_{i \in C_s} w_i^s \geq W_s - \lambda_s$, i.e., the set of land parcels in $V_s \setminus C_s$ is not sufficient for satisfying the suitability quota. Thus, if species s is to be hosted by the reserve, it must hold that

$$\sum_{i \in C_s} z_i \geq u_s, \text{ if } s \in S_1, \quad \text{or} \quad \sum_{i \in C_s} x_i \geq u_s, \text{ if } s \in S_2, \quad (\text{COVER})$$

i.e., at least one land parcel in C_s must be taken. We note that these inequalities are variants of the well-known *cover*-inequalities for knapsack constraints (see, e.g., (Kaparlis and Letchford 2010)).

The covers C_s can be used to define a stronger version of inequalities (SC) by replacing V_s with C_s (obtaining graphs $G_r^{C_s}$ with arcs A_s replaced by $\{(i, s) \mid i \in C_s\}$). The resulting cuts are denoted as *species-cover cuts* (SCC).

3.2 Constraint-Separation

Separation of Connectivity Cuts and Species(-cover) Cuts Let $\rho = (\tilde{\mathbf{u}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{\mathbf{y}})$ be the solution of the LP relaxation at the current node of the branch-and-bound tree. Connectivity cuts (CORECON), (ALLCON) (if a model using them is desired), species cuts (SC) and species-cover cuts (SCC) can be separated using maximum-flow computations on a bi-directed graph D_r based on G_r (resp., $G_r^s, G_r^{C_s}$), where capacities are defined based on ρ (see also Álvarez-Miranda et al. (2013a,b), Carvajal et al. (2013), Fischetti et al. (2017)). While our description applies to (CORECON), the other cuts can be separated in a similar way by a straightforward adaption of the graph used for separation. The idea behind the separation is to define a directed graph by splitting the nodes $i \in V$ to directed arcs (i_1, i_2) with associated capacities \tilde{z}_i . For $\ell \in V$, a minimum $r - \ell$ cut with capacity smaller than \tilde{z}_ℓ can then be used to construct a violated inequality (CORECON).

To be more precise, starting from $G_r = (V \cup \{r\}, E \cup A_r)$, the digraph $D_r = (N, A)$ used in the separation of (CORECON) is defined as follows: For $i \in V$, define nodes i_1 and i_2 , i.e., $N = \{r\} \cup \{i_1, i_2 \mid i \in V\}$ and set $A_z = \{(i_1, i_2) \mid i \in V\}$ and $A'_r = \{(r, i_1) \mid (r, i) \in A_r\}$. Bi-direct the edges $\{i, j\}$ in E to arcs $(i, j), (j, i)$. After bi-directing, all ingoing arcs into i are connected to i_1 , and all outgoing arcs from node i are connected to i_2 . The set A is defined as the set of arcs obtained this way, plus A_z and A'_r , i.e., $A = \{(i_2, j_1), (i_1, j_2) \mid \{i, j\} \in E\} \cup \{(i_1, i_2) \mid i \in V\} \cup \{(r, i_1) \mid i \in V\}$. We then define arc-capacities cap as follows based on ρ :

$$cap_{tv} = \begin{cases} \tilde{z}_i, & \text{if } t = i_1, v = i_2, i \in V, \\ \tilde{y}_i, & \text{if } t = r, v = i_1, i \in V, \\ \infty, & \text{otherwise.} \end{cases}$$

Thus, arcs (i_1, i_2) obtained by splitting nodes $i \in V$ are assigned the value of the corresponding node variable z_i , root-arcs (r, i_1) are assigned the value of the associated root-variable y_i , and all other arcs (i.e., the ones obtained by bi-directing E) are assigned infinite capacity. It is easy to see that with these capacities, any minimum $r - \ell$ cut (for $\ell \in V$) will not involve arcs (i_2, j_1) for $i, j \in V$, as such a cut has infinite capacity. Thus the arcs $\bar{A}_z \cup \bar{A}_r$ induced by any minimal $r - \ell$ cut are subsets of $A_z \cup A'_r$. Any $\bar{A}_z \cup \bar{A}_r$ can be mapped to a set (\bar{W}_V, \bar{W}_A) by defining $\bar{W}_V = \{i \mid (i_1, i_2) \in A_z\}$ and $\bar{W}_A = \{(r, i) \mid (r, i_1) \in A'_r\}$. Now, if

$$\sum_{i \in \bar{W}_V} \bar{z}_i + \sum_{j \in \bar{W}_A} \bar{y}_j < \bar{z}_\ell,$$

then (\bar{W}_V, \bar{W}_A) defines a violated connectivity cut (CORECON). Thus, the separation problem for (CORECON) and a given $\ell \in V$ can be solved in polynomial time by using a maximum-flow/minimum-cut algorithm.

In case ρ is integer, the separation procedure can be much simplified: We construct all connected components H induced by $\bar{z}_i = 1$ (by using, e.g., breadth-first search). If for such an H , $\bar{y}_i = 0$ holds all $i \in H$, the component is not connected to the root node. A connectivity cut (CORECON) can be constructed by defining $W_A = H$ and $W_V = \{j \mid \{i, j\} \in E : i \in H, j \notin H\}$ (i.e., either a node in H must be directly connected to the root node, or one of the nodes neighboring to H must be in the solution) and taking any $\ell \in H$ for the right-hand-side of the cut (we take the node one with smallest index in H).

In both cases (fractional and integer separation), we use the down-lifting by only allowing $j \leq \ell$ for z_ℓ on the left hand-side. For separation of (SCC), we consider the set \bar{C} obtained in the separation routine of the cover inequalities as detailed below.

Separating Cover Inequalities For a given species $s \in S_1$, a cover C_s can be found by solving the following separation problem

$$\min \left\{ \sum_{j \in V_s} \tilde{z}_j q_j \mid \sum_{j \in V_s} w_j^s q_j \geq W_s - \lambda_s \text{ and } \mathbf{q} \in \{0, 1\}^{|V_s|} \right\}. \quad (\text{COVER-Sep})$$

Clearly, the feasible solutions of (COVER-Sep) are the covers C_s , and the objective function minimizes the sum of the \tilde{z}_i for the nodes i in the obtained C_s . Thus, if the solution of (COVER-Sep) is smaller than \tilde{u}_s , a violated inequality (COVER) is obtained (for $s \in S_2$, the z need to be replaced by x). The separation problem (COVER-Sep) is a knapsack-problem in minimization form. We do not solve the problem exactly, but use the following heuristic: First, we sort the nodes in a non-decreasing way by \tilde{z}_j/w_j^s ; then, construct we a cover C_s by iteratively picking the nodes sorted in this way, starting with the smallest ratio, until $\sum_{j \in C_s} w_j^s \geq W_s - \lambda_s$.

Implementation of the Cut-Loop At any branch-and-bound node, we do not aim to separate all violated inequalities, but follow the outline described in the following list, and only move from one separation routine to the next, if no violated inequalities have been found:

1. Separate (COVER) and (SCC) or separate (SC) (depending on the chosen separation strategy, note that (SCC) dominate (SC)).
2. Separate (CORECON)

This scheme is followed in order to avoid excessive calls of the (time consuming) separation routines and also to avoid overloading the LP with too many inequalities. To further achieve this goal, the separation of (CORECON) is not done for all nodes $\ell \in V$ with $\tilde{z}_\ell > 0$, but only for nodes with $\tilde{z}_\ell \geq \tau$, where τ is a given separation threshold parameter (we tried values of 0.5 and 0.1 in our computational experiments, see Section 4). Moreover, once a violated inequality (CORECON) is found for some ℓ , we do not consider the nodes $\{i \mid (r, i) \in W_A\}$ for separation.

For integer solutions of the LP relaxation, only connectivity cuts (CORECON) are separated (observe that for correctness of our approach, only this separation is necessary). In our computational experiments,

we also tried other separation strategies, such as separating fractional points only at the root node, or separating integer points ρ only. Details are given in Section 4.

3.3 Construction Heuristic and Primal Heuristic

We have designed a construction heuristic, to generate a starting solution for initializing the branch-and-cut, and a primal heuristic, which is incorporated inside of the branch-and-cut framework. Moreover, to improve the solution found by the construction heuristic, we also implemented a local-branching ILP-heuristic Fischetti and Lodi (2003).

The use of such heuristics was crucial, since initial computational experiments showed that the internal heuristics of CPLEX, the general purpose ILP-solver we used, did only find feasible solutions of very poor quality (if any solution was found at all). This is likely due to the symmetric nature of our problem, the structure of the instances and the fact that, due to the cutting plane approach, only a partial information about the nature of the problem at hand is given to the ILP-solver.

Construction Heuristic: Phase One The construction heuristic first creates a feasible solution in a greedy fashion, and then tries to remove unnecessary land parcels in a post-processing phase. The greedy heuristic is based on the TM heuristic by Takahashi and Matsuyama (1980) for the Steiner tree problem. Recall that in the Steiner tree problem, we are given a graph $G = (V, E)$, edge costs $c : E \rightarrow \mathbb{R}_{>0}$ and terminal set $T \subset V$ and we want to find the minimum cost tree containing all T terminals. In the TM heuristic, one starts with a partial solution \mathcal{S} consisting of a single node from T . Let $T' = T \setminus \mathcal{S}$. Shortest paths to all $t \in T'$ are calculated. Let $t^* \in T'$ be the terminal in T' with minimum shortest-path distance to \mathcal{S} . The terminal t^* and all nodes and edges on the shortest path from \mathcal{S} to t^* are added to \mathcal{S} . This process is repeated, until all terminals are added in \mathcal{S} .

To use a similar heuristic for the GRSC-CB, some adaptations need to be made to deal with the following differences to the Steiner tree problem: i) the solution can have up to k components, ii) each component consists of a connected core surrounded by a buffer, iii) there is no set of terminals T to be connected, but the species protection constraints (S_1 -PROTECT) and (S_2 -PROTECT) must be fulfilled instead, i.e., $P_1(P_2)$ species from $S_1(S_2)$ must be protected in the solution (which in turn depends on fulfilling the suitability quota constraints (S_1 -SQ), resp., (S_2 -SQ)).

In our heuristic (see Algorithm 1), the partial solution \mathcal{S} is stored as $(\mathcal{S}_z, \mathcal{S}_x)$, where \mathcal{S}_z contains the core nodes of the partial solution, and \mathcal{S}_x contains all nodes of the partial solution. For each species s , we also keep track of the habitat score of the nodes in the partial solution; these values are stored in $W_s(\mathcal{S})$. The function $u_s(\mathcal{S})$ is **true**, if and only if the suitability quota constraint (S_1 -SQ), resp., (S_2 -SQ) for s is fulfilled by \mathcal{S} and it is **false**, otherwise. The function $protectedS_1(\mathcal{S})$, resp., $protectedS_2(\mathcal{S})$ is **true**, if and only if $P_1(P_2)$ species from $S_1(S_2)$ are protected in the partial solution \mathcal{S} and it is **false**, otherwise. A solution \mathcal{S} is feasible if and only if both $protectedS_1(\mathcal{S})$ and $protectedS_2(\mathcal{S})$ are **true**. The algorithm also uses a function $computeShortestPaths(V', V'')$ which computes the (node-weighted) shortest-paths between V' and any node $j \in V''$ and returns the distances. Moreover, the method $nodesOnShortestPath(V', V'')$ returns the nodes on the shortest-path between V' and V'' . More details are given below.

During the course of the heuristic, we select nodes i to add to \mathcal{S}_z based on shortest-path calculations similar to the TM heuristic, i.e., we build connected cores. We initialize \mathcal{S}_z by randomly selecting k nodes, which ensures that the final solution has at most k connected components. Whenever a node is added to \mathcal{S}_z , its d -neighborhood is added to \mathcal{S}_x (i.e., to accommodate for the the buffer-constraint).

In contrast to the standard TM heuristic, in our approach the terminal set for the shortest-path calculations is not known in advance. Instead, the set of terminals, denoted by $T(\mathcal{S})$, is dynamically updated in each iteration, based on the current solution \mathcal{S} and the associated values of $u_s(\mathcal{S})$, $protectedS_1(\mathcal{S})$ and $protectedS_2(\mathcal{S})$. Given a partial solution \mathcal{S} , a node i belongs to $T(\mathcal{S})$ if and only if adding it to \mathcal{S}_z is *helpful* with respect to $protectedS_1(\mathcal{S})$ or $protectedS_2(\mathcal{S})$. A node i is deemed *helpful* with respect to $protectedS_1(\mathcal{S})$, if $protectedS_1(\mathcal{S})$ is **false**, and $i \in V_s$ for at least one $s \in S_1$ with $u_s(\mathcal{S}) = \mathbf{false}$, i.e., if adding it as a core node increases the habitat suitability score for a species from S_1 not hosted by \mathcal{S} , and \mathcal{S} does not already

Algorithm 1: Construction Heuristic

Data: An instance of the GRSC-CB

Result: A feasible solution $\mathcal{S} = (\mathcal{S}_z, \mathcal{S}_x)$ for the GRSC-CB

/ the set $T(\mathcal{S})$ of terminals is dynamically updated, see the text for details */*
randomly pick k nodes from V to initialize \mathcal{S}_z

$\mathcal{S}_x \leftarrow \bigcup_{i \in \mathcal{S}_z} \delta_d^+(i)$

while $\neg \text{protected}S_1(\mathcal{S}) \vee \neg \text{protected}S_2(\mathcal{S})$ **do**

$d \leftarrow \text{computeShortestPaths}(\mathcal{S}_z, T(\mathcal{S}))$

$i^* \leftarrow \arg \min_{j \in T(\mathcal{S})} d_j$

$\text{nodes}SP \leftarrow \text{nodesOnShortestPath}(\mathcal{S}_z, i^*) \cup \{i^*\}$

for $i \in \text{nodes}SP$ **do**

$\mathcal{S}_z \leftarrow \mathcal{S}_z \cup \{i\}$

$\mathcal{S}_x \leftarrow \mathcal{S}_x \cup \delta_d^+(i)$

/ update $T(\mathcal{S})$ */*

**/*

host P_1 species from S_1 . Helpfulness of i with respect to $\text{protected}S_2(\mathcal{S})$ is defined similarly; in addition, all nodes $j \in \delta_d^+(i)$ are also included in the helpfulness-check (and not just i).

To measure the level of helpfulness of a node, we introduce a node-cost function $\Delta_i(\mathcal{S})$ that is employed as cost function for the shortest-path calculations (which are done with respect to the node-cost). The value of the function $\Delta_i(\mathcal{S})$ is dynamically updated based on the partial solution given by \mathcal{S} . This is done to take into account that adding a node i to the core (i.e., to \mathcal{S}_z) may also cause some other nodes j to be added to \mathcal{S}_x , due to the buffer-constraints. Moreover, by $\Delta_i(\mathcal{S})$ we also try to take into account that the helpfulness of a node i changes, depending on the suitability scores w_i^s of the node and the species already protected by \mathcal{S} . Let

$$\mathcal{C}_i(\mathcal{S}) = \sum_{j \in \delta_d^+(i) \setminus \mathcal{S}_x} c_j,$$

$$\mathcal{W}_s^1(i, \mathcal{S}) = \left(w_i^s + W_s(\mathcal{S}) - \lambda_s \right) \left(1 - u_s(\mathcal{S}) \right) \quad \text{for } s \in S_1$$

and

$$\mathcal{W}_s^2(i, \mathcal{S}) = \left(\sum_{j \in \delta_d^+(i) \setminus \mathcal{S}_x} w_j^s + W_s(\mathcal{S}) - \lambda_s \right) \left(1 - u_s(\mathcal{S}) \right) \quad \text{for } s \in S_2,$$

where the value **true** as output of $u_s(\mathcal{S})$ is interpreted as one and **false** as zero. The value $\mathcal{C}_i(\mathcal{S})$ measures the cost of adding node i to the solution, while $\mathcal{W}_s^1(\mathcal{S})$ and $\mathcal{W}_s^2(\mathcal{S})$ tries to capture the helpfulness of adding i to the solution with respect to a species s . The node cost $\Delta_i(\mathcal{S})$ is finally defined as follows:

$$\Delta_i(\mathcal{S}) = \frac{\mathcal{C}_i(\mathcal{S}) + 0.001}{\left(\sum_{s \in S_1} \mathcal{W}_s^1(i, \mathcal{S}) \right) \left(1 - \text{protected}S_1(\mathcal{S}) \right) + \left(\sum_{s \in S_2} \mathcal{W}_s^2(i, \mathcal{S}) \right) \left(1 - \text{protected}S_2(\mathcal{S}) \right) + 0.0001},$$

where the value **true** as output of $\text{protected}S_1(\mathcal{S})$, $\text{protected}S_2(\mathcal{S})$ is interpreted as one and **false** as zero.

The construction heuristic is run for $nstarts = 20$ different random starting solutions \mathcal{S}_z .

Primal Heuristic: Phase One As a primal heuristic during the branch-and-cut, we use a slightly modified version of the construction heuristic: In the calculation of the node-cost \mathcal{C}_i , we use $c_i(1 - \tilde{x}_i)$ instead of c_i . Moreover, the randomly generated starting solutions are constructed by considering the of nodes with $\tilde{y}_i \geq 0.001$. In both cases, we run a post-processing procedure, in which we try to remove unnecessary nodes from \mathcal{S}_z , as described below.

Construction and Primal Heuristic: Phase Two (Post-Processing) The post-processing is a greedy local improvement procedure in which we iterate through the nodes $i \in \mathcal{S}_z$, and check, if after removing i , the solution remains feasible. Note that, together with i we are also removing all nodes from $\delta_d(i)$ which become redundant after removing i . Let us denote this set of nodes with \mathcal{S}_x^i where

$$\mathcal{S}_x^i = \delta_d^+(i) \setminus \bigcup_{j \in \mathcal{S}_z, j \neq i} \delta_d^+(j).$$

Let i^* be the node whose removal (together with \mathcal{S}_x^i) results in the largest improvement in the objective function. We remove i^* from \mathcal{S}_z and \mathcal{S}_x^i from \mathcal{S}_x , and repeat the process until no additional node can be removed.

Finally, we point out that a similar heuristic can also be used for GRSC-C by setting the buffer size to zero, so we have $\delta_d^+(i) = \{i\}$.

3.4 Local Branching-based Heuristic

The solution found by the construction heuristic is further improved using an ILP-based local-search procedure known as *local branching* by Fischetti and Lodi (2003). Given a feasible solution \mathcal{S} , local branching explores its r -neighborhood by employing an ILP-solver in a black-box fashion. In the following, we provide specific details of our implementation that deviate from the standard recipe given in Fischetti and Lodi (2003), following an improved scheme from Fischetti et al. (2017).

In each local search iteration, we start with the basic ILP-formulation of the problem (given in Section 2), and extend it through an additional local branching constraint which specifies the r -neighborhood with respect to \mathcal{S} . In our case, this ILP-formulation is solved through the branch-and-cut, enhanced by the primal heuristic. Even though the complexity of the resulting formulation inherits the complexity of the original problem, its feasible region is significantly smaller due to the choice of the parameter r . Furthermore, the resulting ILP does not have to be solved to optimality; instead, one interrupts the solver as soon as a feasible solution is found, i.e., the *first-improvement* local search strategy is applied. In addition, we impose a time-limit for each local search iteration. If this time-limit is reached, this means that no improving solution is found in the current neighborhood. In the latter case, the size of the neighborhood is increased by Δ_r . Whenever a new best solution is found, the size of the neighborhood is reset to r . The procedure is then repeated with the improved solution \mathcal{S}' (or with the larger neighborhood), until one of the stopping criteria is satisfied: (i) the maximum number of local search iterations is reached, (ii) the maximum neighborhood size is reached, or (iii) the overall time limit for the local branching phase is reached.

We point out that our local branching takes an additional advantage of the primal heuristic as follows: if the heuristic manages to produce a feasible solution \mathcal{S}' which improves upon the currently best known one, but is infeasible with respect to the local branching constraint, the current local search iteration is interrupted, and the procedure is repeated by exploring the neighborhood of \mathcal{S}' .

Let \mathcal{S}_z be the set of i with $z_i = 1$ in a given solution \mathcal{S} and let r be a given radius. The r -neighborhood with respect to \mathcal{S} is defined as a set of all feasible solutions whose Hamming distance with respect to \mathcal{S}_z is not bigger than r . Consequently, the following local branching constraint is utilized in our framework:

$$\sum_{i \in \mathcal{S}_z} z_i \geq |\mathcal{S}_z| - r. \tag{LOCBRA}$$

The constraint (LOCBRA) ensures that at least $|\mathcal{S}_z| - r$ of the core land parcels of the solution \mathcal{S} also belong to the new solution \mathcal{S}' .

Upon the termination of the local branching procedure, the branch-and-cut is started. The second important and non-standard feature of our local branching implementation is the utilization of a *cutpool*, which collects all violated inequalities detected during the local branching phase. These inequalities are used to initialize the final call of the branch-and-cut procedure. The arc-node separator inequalities (CORECON) found during the local branching phase are globally valid, and hence, their collection and recycling through the cutpool significantly influences the overall computing time. Furthermore, the cuts from the cutpool added

at the root node also trigger general-purpose cutting planes implemented within an ILP-solver, resulting in stronger bounds at the root node. Furthermore, the inequalities (CORECON) from the *cutpool* are also used to initialize each subsequent local search iteration.

In our implementation, we use $r = 5$. As a time-limit for the ILPs (i.e., for each single local search iteration) we set 20 seconds. If no improved solution is found, r is increased by $\Delta_r = 5$, until the maximum neighborhood size of 20 is reached. If an improved solution is found, r is reset to five. These settings have been determined using preliminary computations.

4 Computational Results

In order to assess the effectiveness and suitability of the proposed approach, we implemented our branch-and-cut framework and tested it on three data sets. The first data set contains synthetic instances based on grid-graphs, while the second and third data sets are real-life case studies encompassing instances retrieved from the geographic and ecological survey data.

The implementation of the branch-and-cut is done using CPLEX 12.7 as a generic-purpose ILP-solver. All CPLEX-parameters were left at their default values. The experiments were carried out on an Intel Xeon CPU with 2.5 GHz and 16GB of RAM using a single-thread mode.

4.1 Benchmark Instances

Grid-Graph Instances Following the procedure proposed in Dilkina and Gomes (2010), Wang and Önal (2016), our grid-graph instances were generated as follows: A grid of $n \times n$ nodes (set V) was created, and an edge $\{i, j\}$ between $i, j \in V$ exists if and only if i and j are adjacent in this grid. For each node i , the cost c_i was set to an integer value taken uniformly at random from the range $[1, 100]$. The habitat suitability w_i^s for node i and species s was set to an integer value taken uniformly at random from the range $[20, 100]$. After generating this value, w_i^s is re-set to zero with probability 20% (for $s \in S_1$), resp. 10% (for $s \in S_2$) to account for the fact that normally not all land parcels are suited for all species.

We generated four sets of ten instances according to the following scheme:

- *Set 1*: $n = 20$ (hence $V = 400$, $E = 760$), $|S_1| = 1$, $|S_2| = 3$.
- *Set 2*: $n = 20$ (hence $V = 400$, $E = 760$), $|S_1| = 3$, $|S_2| = 9$.
- *Set 3*: $n = 30$ (hence $V = 900$, $E = 1740$), $|S_1| = 1$, $|S_2| = 3$.
- *Set 4*: $n = 30$ (hence $V = 900$, $E = 1740$), $|S_1| = 3$, $|S_2| = 9$.

We considered the following three conservation scenarios:

- *Scenario A*: $P_1 = |S_1|$, $P_2 = |S_2|$.
- *Scenario B*: $P_1 = |S_1|$, $P_2 = \lceil 0.5 \cdot |S_2| \rceil$.
- *Scenario C*: $P_1 = |S_1|$, $P_2 = 0$.

In our experiments, the buffer width d is set to one and scores w_i^s are set to zero for all nodes i at the boundary of an instance, as nodes at the boundary cannot have buffer nodes at one or more sides (as no such nodes exist). We considered $k = 1$ and $k = 3$, i.e., the solution can consist of one connected component, or at most three connected components, and we set $\lambda_s = \lceil 0.05 \rceil \sum_{i \in V_s} w_i^s$ for all $s \in S$. Thus, in total, we have 240 grid-graph instances (four sets times ten instances times three scenarios times two different values of k).

GAP-Instances These real-life instances are based on data of the National Gap Analysis Program (GAP), an initiative of the U.S. Geological Survey (USGS) (U.S. Geological Survey 2016a,b). We considered three states of different size, namely Oregon, Pennsylvania and Vermont. In total, 18 real-life instances are generated – three states times three scenarios (A,B,C, mentioned above) times two different values of k . A detailed case-study which is conducted on this data set is given in Section 4.4.

4.2 Computational Setting

In order to analyze the influence of valid inequalities proposed in this paper, and the role of primal and local branching heuristics, the following four settings are compared:

- **Basic**: This is a basic setting in which only arc-node-separator cuts (CORECON) are separated.
- **Basic+**: In this setting, in addition to (CORECON), (COVER) and (SCC) are separated as well.
- **Basic+CP**: The setting is the same as **Basic+**, but the construction heuristic and the primal heuristic are turned on.
- **Basic+CPLB**: Finally, in this extension of the **Basic+CP** setting, the local branching procedure has been invoked between the construction heuristic and the branch-and-cut.

By default, in each of the settings, fractional points are separated only at the root node (with at most 20 cuts of type (COVER), (SCC) being separated). Since (SC) are dominated by (SCC), they are not separated in our framework. The separation threshold τ for (CORECON) cuts is set to 0.5. In the experiments for the grid-graphs, we used a time-limit of 1800 seconds. A time-limit of 180 seconds was given to the local branching phase.

In the following, we provide results of the computational comparison of the four settings on the grid-graph instances, before we provide a detailed analysis concerning the important structural and performance indicators on the set of real-life instances.

4.3 Results on Grid-Graph Instances

Influence of Valid Inequalities In this section we study the influence of the valid inequalities to the quality of bounds obtained at the root node of the branch-and-cut tree (*root bounds* in the following) and the overall computing time. We also analyze the problem difficulty with respect to the imposed value of k , which is the maximum allowed number of core components. To this end, we compare the computing times to optimality for the settings **Basic** and **Basic+**, and the relative improvement of the root bound obtained by additionally including inequalities (COVER) and (SCC). The results of such comparison are reported in charts of Figure 2, which are analyzed below.

The performance chart given in Figure 2a depicts the cumulative computing time for the settings **Basic** and **Basic+**, and for $k = 1$ and $k = 3$. In this chart, a point with coordinates (x, y) indicates that for $y\%$ of the instances of the considered data set, the total computing time was $\leq x$ seconds. The first observation that can be made from the obtained result is that solving GRSC-CB with a single connected component is computationally much more challenging than solving GRSC-CB in which the number of components is relaxed to a greater value. Whereas almost all of the 240 grid-graph instances could be solved to optimality for $k = 3$, around 20% of them remain unsolved within the same time limit if a single core component is imposed. Furthermore, for a fixed value of k , comparison of computing times between **Basic** and **Basic+** reveals that the time invested into a (rather time-consuming) separation of (COVER) and (SCC) inequalities does not significantly influence the overall performance. This is due to our moderate separation strategy which allows for up to 20 cuts of these types to be added at the root node.

To compare the influence of the valid inequalities to the quality of the root bound, we report the relative improvement of the root bound obtained from the setting **Basic+** with respect to the setting **Basic**. Figure 2b depicts these values in the cumulative fashion, for $k = 1$ and $k = 3$. Formally, the relative root bound

improvement is defined as $\frac{RB(\text{Basic+}) - RB(\text{Basic})}{RB(\text{Basic})} \cdot 100\%$, where $RB(\cdot)$ stands for the the lower bound obtained at the root node of the branch-and-cut tree. Notice that the reported root bounds already take into account the general purpose cuts found by CPLEX, which are turned on by default in all our settings. This also explains why the obtained relative improvements can sometimes take negative values.

The obtained results indicate that the root bound of the setting **Basic** can be improved by up to 6%, due to our valid inequalities. These relative improvements are particularly pronounced for the more challenging setting in which a single core component is imposed.

We therefore conclude that the computational overhead imposed by the separation of valid inequalities is very moderate compared to the benefits achieved through the improvement of the root bounds, and keep the separation of valid inequalities turned on by default in the remainder of this study.

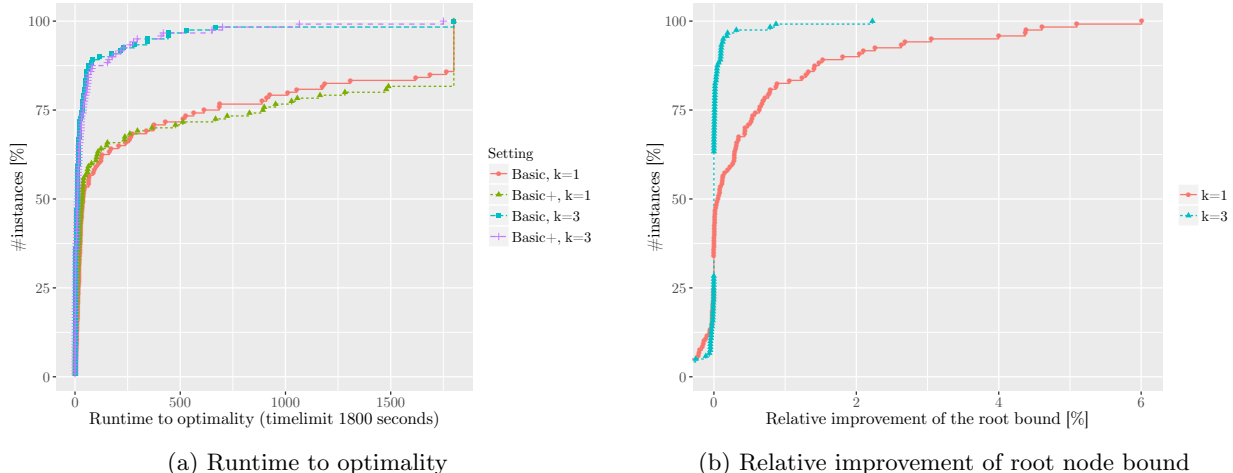


Figure 2: Comparison of the settings **Basic** and **Basic+** for the grid-graphs and $k = 1$, $k = 3$.

Influence of Heuristics We now turn our focus on the quality of upper bounds obtained by our solution framework. To this end, we compare the quality of *primal bounds* used to initialize the branch-and-cut procedure. On the one hand, these bounds are obtained by running the construction heuristic only (setting **Basic+CP**) or, by running the local branching procedure in addition (setting **Basic+CPLB**). Figure 3 analyzes the quality of the construction heuristic and the local branching procedure. In these two cumulative charts, we plot the primal gap $pg[\%]$ against the number of instances: Figure 3a shows the results for $k = 1$ and Figure 3b shows the results for $k = 3$. The primal gap is calculated as $100 \cdot (z^H - z^*)/z^*$, where z^H is the solution value obtained by the construction heuristic/local branching procedure, and z^* is the optimal (or the best known) solution value for the instance. The charts show that the construction heuristic works already quite well: for around 90% ($k = 1$), resp., over 60% ($k = 3$) the primal gap is under 20%. Using the local branching procedure after the construction heuristic significantly improves this result, for both $k = 1$ and $k = 3$. In case of a single core component, for more than 75% of the instances, the optimal solution is found upon the termination of the local branching, and the worst primal gap is less than 7%. Similarly, for $k = 3$, local branching finds the optimal solution for about 75% of the instances, whereas the worst obtained gap remains below 5%.

We note that there is a moderate computational overhead imposed by the local branching: the average computing time needed for the construction heuristic is below one second, whereas the local branching phase requires around 60 seconds, on average over all 240 grid-graph instances. This indicates that local branching does not necessarily pay off for the small instances that can be solved quickly by enumeration in the branch-and-cut tree. On the contrary, high-quality solutions are particularly important for larger instances where the branch-and-cut framework encounters difficulties in closing the final gap. We therefore decide to use

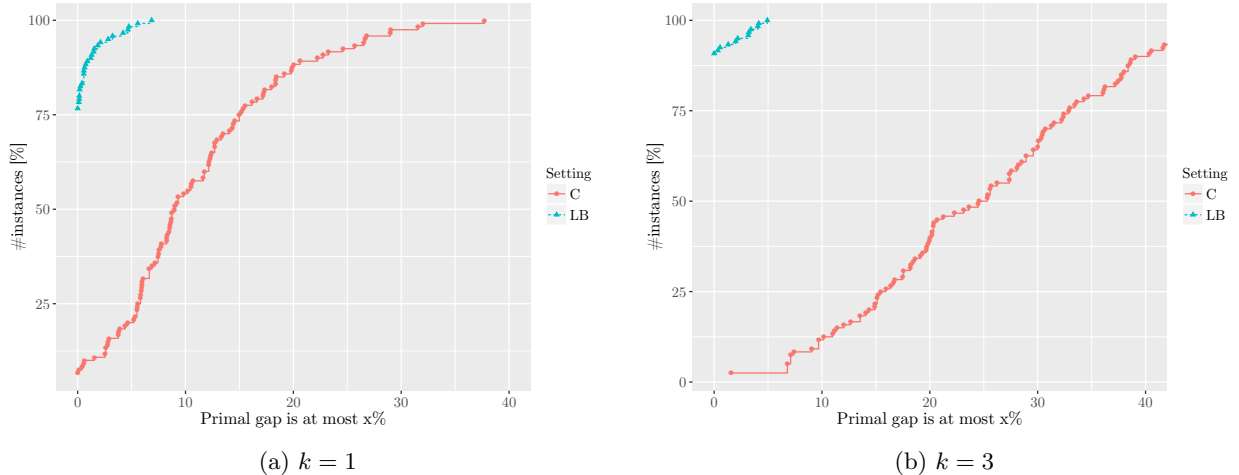


Figure 3: Primal gap of the solutions obtained by the construction heuristic (C) and by local branching (LB) and $k = 1$, $k = 3$

the setting **Basic+CPLB** for the case study presented in Section 4.4, since most of the instances considered therein are of the latter type.

4.4 Case Study: U.S. Wildlife Conservation

With the strong increase in the number of endangered species around the globe, several organizations have established sophisticated procedures to gather ecological information of vast areas of the habitat of different types of flora and fauna. Typically, these habitat surveys are comprised of ecological assessment of the studied area, classification of the species of interest, economical characterizations, and much more. Relevant examples of research groups, institutes, and public institutions can be found, for instance, in (Cayton et al. 2017a, Corporation 2016, Department of Biosciences (U. of Helsinki) 2016).

A prominent example of an ecology information system is the aforementioned GAP of the USGS. This program follows the methodology proposed in the seminal work of (Scott et al. 1993), and it is the result of decades of exhaustive efforts devoted to provide clear, geographically-explicit information on the distribution of native vertebrate species, their habitat preferences, and their management status, in order to determine actual needs in biodiversity protection.

Among the different datasets provided by the GAP program, it is possible to obtain, for each U.S. state, a representation of its territory mapped into so-called hydrological units (HU); these units and their adjacencies are used to build $G = (V, E)$. Although the databases supplied by the GAP program have been used before for analyzing the current wildlife conservation policies (see, e.g., Drew et al. (2011), Lacher and Wilkerson (2014), Meretsky et al. (2012), Minor and Lookingbill (2010)), we are not aware of OR-oriented papers using this GAP data. For this study, we have used data from three U.S. states of varying sizes (Oregon, Pennsylvania and Vermont). For a given state encoded by a graph $G = (V, E)$, the problem parameters are obtained as follows.

- The set S consists of the terrestrial mammals living in the state according to the GAP data. All species which are classified as endangered or vulnerable (at federal or state level) by the corresponding Department of Wildlife are put into S_1 and the remaining species are put into S_2 . The complete lists of animals in set S_1 for each considered state is given in Table 1. As an interesting observation, for Vermont, two of the species (American Marten, Eastern Mountain Lion) listed by the Department of Wildlife do not occur in the GAP-data of the state.
- In the GAP-data, there are also species distribution models for each species available on a 30 meters

Table 1: Animals in S_1 for the considered states

Oregon (see (Oregon Department of Fish and Wildlife 2015))
Canadian Lynx (<i>Lynx canadensis</i>)
Gray Wolf (<i>Canis lupus</i>)
Columbian White-tailed Deer (<i>Odocoileus virginianus leucurus</i>)
Fisher (<i>Martes pennanti</i>)
Pygmy Rabbit (<i>Brachylagus idahoensis</i>)
Pennsylvania (see (US Fish and Wildlife Service 2016))
Indiana Bat (<i>Myotis sodalis</i>)
Northern long-eared Bat (<i>Myotis septentrionalis</i>)
Vermont (see (Vermont Fish & Wildlife Department 2015))
Canadian Lynx (<i>Lynx canadensis</i>)
Eastern Small-footed Bat (<i>Myotis leibii</i>)
Little Brown Bat (<i>Myotis lucifugus</i>)
Northern Bat (<i>Myotis septentrionalis</i>)
Indiana Bat (<i>Myotis sodalis</i>)
Eastern Pipistrelle (<i>Pipistrellus subflavus</i>)

$\times 30$ meters cell basis, i.e., for each such cell and species, there is an "yes/no" flag indicating if the particular cell is suitable for the species. We calculate the score w_i^s of each node i (i.e., each HU) and species s by counting the number of "yes"-cells within the HU.

- In order to avoid any arbitrary estimation, the cost of taking a land parcel $i \in V$ as part of the reserve is given by the area a_i (in squared meters) of the corresponding HU (see, e.g., Adams et al. (2016), Hermoso et al. (2012), Stewart et al. (2007)).

Table 2 gives details about the problem instances (column *source* gives the source articles used for classification of species into S_1 and S_2).

Table 2: Characteristics of the states under consideration

<i>state</i>	area (km ²)	#parcels	avg. parcel-area (km ²)	$ S_1 $	$ S_2 $	source
Oregon	254,799	3134	81.3	5	69	Oregon Department of Fish and Wildlife (2015)
Pennsylvania	119,280	1452	82.2	2	27	US Fish and Wildlife Service (2016)
Vermont	24,906	301	82.7	6	20	Vermont Fish & Wildlife Department (2015)

Spatial Analysis and Benefits of the GRSC-CB Model We now compare the four models addressed in this article for reserve set covering, namely GRSC, GRSC-B, GRSC-C and GRSC-CB. Recall that the problems GRSC and GRSC-B can be modeled as compact ILP formulations and given to a black-box ILP-solver without any additional interventions. For the models GRSC-C and GRSC-CB, the branch-and-cut implementation described in this article is used.

Table 3 gives a comparison of the results for the real-life instances for GRSC, GRSC-B, GRSC-C and GRSC-CB for $k = 3$. The table gives the number of components of the solution ($\#co.$), the number of land parcels ($\#lp.$), the best objective value (z^*), and the runtime ($t[s.]$). The time-limit for these runs was set to 3 hours and an entry *TL* indicates that the instance could not be solved to optimality within this given time-limit. In this case, the number in parentheses next to *TL* gives the optimality gap, which is calculated as $100 \cdot (z^* - LB)/z^*$, where *LB* is the obtained lower bound. In addition, Figure 4 gives a plot of $z^{GRSC-CB}/z^{GRSC}$, $z^{GRSC-CB}/z^{GRSC-B}$ and $z^{GRSC-CB}/z^{GRSC-C}$ (where $z^{\mathcal{P}}$ is the best solution value obtained for problem $\mathcal{P} \in \{\text{GRSC, GRSC-B, GRSC-C, GRSC-CB}\}$). This ratio is an indicator for the potential increase in cost incurred by using the more sophisticated reserve design strategy GRSC-CB compared to the simpler strategies (of course, a solution of a less-constraint problem may also fulfill the constraints explicitly imposed in GRSC-CB). The two compact models (GRSC, GRSC-B) were given directly to CPLEX, while setting `Basic+CPLB` was used for GRSC-C and GRSC-CB.

The obtained results reveal that the model GRSC is the easiest to solve (only one instance remains unsolved within the time-limit, and many are solved within a few seconds only), but gives very fragmented reserves, consisting of 12 to 310 connected components. The structure of the solutions is slightly better when buffer constraints are imposed: Except for OR instance with Scenario A, the solutions for GRSC-B consist of at most four connected components, and for five of the nine instances, they consist of at most 2 components, i.e., they are even feasible for GRSC-CB with $k = 3$. The cost of the solutions for GRSC-C and GRSC is very similar, i.e., just imposing connectivity of the reserve only marginally increases the cost of the reserve. On the other hand, imposing buffer constraints has a much stronger impact on the solution in terms of the overall cost: the cost for the solutions of GRSC-CB (and also GRSC-B) is up to 2.5 times higher than the cost of the solutions without buffer requirements.

Table 3: Number of components, land parcels, solution value and runtime for the best solution for the real-life instances.

<i>inst. sc.</i>	GRSC		GRSC-B		GRSC-C, $k=1$		GRSC-C, $k=3$		GRSC-CB, $k=1$		GRSC-CB, $k=3$					
	#c. #lp.	$z^* t[s.](g[\%])$	#c. #lp.	$z^* t[s.](g[\%])$	#lp.	$z^* t[s.](g[\%])$	#c. #lp.	$z^* t[s.](g[\%])$	#lp.	$z^* t[s.](g[\%])$	#c. #lp.	$z^* t[s.](g[\%])$				
VT A	25	28	10333	4.40	2	22301	9.32	3	13	10343	TL (0.04)	2	23	22301	56.98	
VT B	12	12	9192	2.18	1	21591	4.16	7	8	9377	TL (0.11)	1	19	21591	43.71	
VT C	12	13	9113	0.21	1	21591	3.80	8	8	9277	795.66	1	19	21591	32.28	
PA A	101	124	54564	2.53	5	80	72559	TL (5.12)	3	72	54623	TL (0.05)	3	83	75630	TL (15.43)
PA B	26	52	40936	614.68	2	68	56181	7649.12	3	47	43691	9135.35	2	65	56296	TL (3.04)
PA C	13	30	22351	0.11	2	56	50182	71.49	3	32	22929	91.49	2	56	50182	305.19
OR A	198	237	120894	300.58	7	169	138558	TL (13.47)	3	171	121688	TL (0.62)	3	181	155624	TL (27.47)
OR B	43	90	72369	TL (0.17)	4	151	112322	TL (17.34)	3	100	85822	TL (30.20)	3	140	112156	TL (19.40)
OR C	40	73	55244	4.39	4	134	108024	TL (7.61)	3	68	58181	TL (2.26)	3	134	108035	TL (11.60)

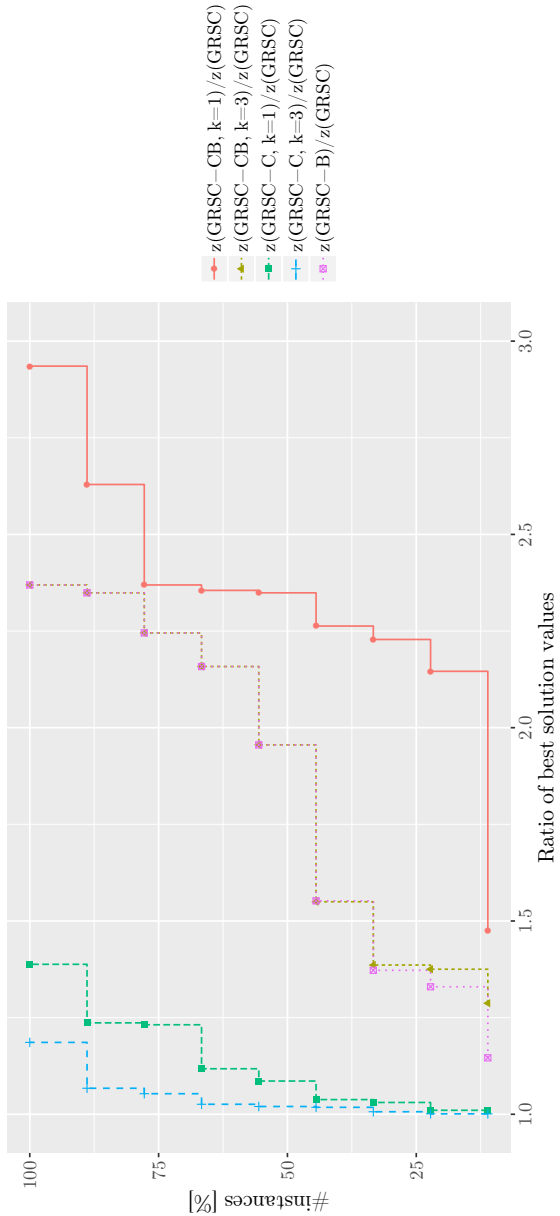


Figure 4: Ratio of the best solution values for the different problems

We now look closer into the spacial structure of the obtained solutions, when the four models are applied to the same instance. Figure 5 shows the best solutions attained for instance OR and Scenario C for all considered problem variants (note that for GRSC-C and GRSC-CB, we also compare the solutions with $k = 1$ and $k = 3$). The reported solutions show the potential of the developed framework to design reserves that respond to different ecological needs. It is not surprising that GRSC solutions produce very fragmented reserves as the one depicted in Figure 5a. Such solutions may be suitable when existing protected areas need to be reinforced by intensifying the protection policy by means of cost-efficient actions. On the contrary, if the conservation planners seek to design reserves that are comprised by less scattered units, but do not explicitly require the areas to be connected in the strong sense, the solutions obtained by using GRSC-B (see e.g., Figure 5b) seem quite suitable. This solution is comprised of only four components, and the components are spatially *compact*, which are both desirable characteristics in many planning contexts. The solutions reported in Figures 5c and 5e represent optimal solutions for GRSC-C with $k = 1$ and $k = 3$, respectively. Comparing the spatial structure of these two solutions, we observe that different values of k result in spatially completely different solutions. This result demonstrates how important for the decision makers is to determine the right choice of the cardinality bound k when reserves with connectivity requirements have to be designed. Moreover, the same effect of k holds for the solutions of the GRSC-CB.

Finally, Figures 5d and 5f depict the optimal GRSC-CB solutions for $k = 1$ and $k = 3$, respectively. Due to the presence of the buffer layer, these solutions are by far more compact than their counterparts obtained by the alternative three models, GRSC, GRSC-B or GRSC-C. This balance of connectivity and compactness makes clear that the GRSC-CB formulation is capable of successfully embodying these two ecologically functional characteristics to the obtained solutions.

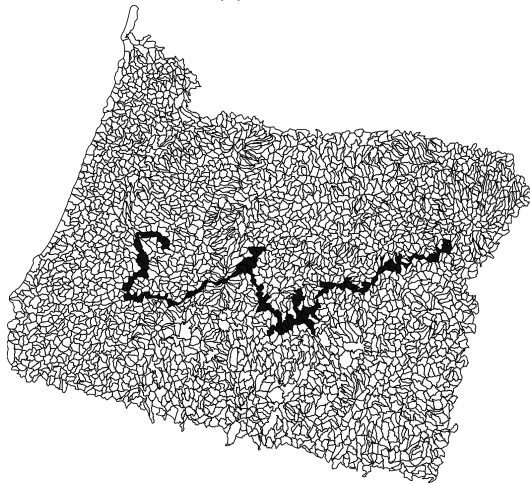
The diversity of the produced solutions are an evidence that connectivity and buffer zones are important features that greatly influence the spatial layout of the obtained reserves. Imposing these features allows to define different spatial arrangements, leading to different ecological profiles that address different conservation needs. For instance, the solutions depicted in Figures 5c and 5d resemble wildlife corridors, which are conservation plans that are suitable when reserves need be spatially and functionally compatible with other human activities. Therefore, the GRSC-CB model and its variants, along with the corresponding algorithmic framework given in this paper, provide a powerful tool for conservation planning. They deliver a flexible decision-aid framework that addresses many different modeling requirements for designing conservation reserves.



(a) GRSC



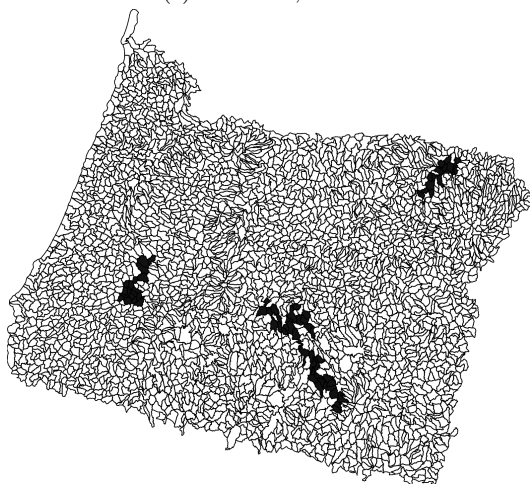
(b) GRSC-B



(c) GRSC-C, $k = 1$



(d) GRSC-CB, $k = 1$



(e) GRSC-C, $k = 3$



(f) GRSC-CB, $k = 3$

Figure 5: Solutions obtained for instance OR and Scenario C

4.5 Case Study: Mitchell river catchment

As well as U.S., other countries have devoted enormous financial and human efforts for gathering and analyzing biodiversity information, and for designing and implementing conservation plans. One important example corresponds to the Northern Australia Water Futures Assessment program Department of Agriculture & Water Resources (2012), which aims at providing information needed to inform the development and protection of northern Australia’s water resources, so that development is ecologically, culturally and economically sustainable. Within this program, one can find the Northern Australia Aquatic Ecological Assets initiative Griffith University (2012) (NAAE), which focuses on conducting fine scale assessments of particular catchments in northern Australia.

One of the most important catchment areas approached by the NAAE initiative is the Mitchell river catchment, located in Queensland, northern Australia. Within this catchment area, several freshwater fish species were classified as threatened, and their main threats were spatially and functionally identified. The studied area (71,630 km²) was divided into 2,316 land parcels (i.e., sub-catchments), and the connections among them corresponded to the river stream network (for further details, see Cattarino et al. (2015)).

In the considered region, 46 species (all of them were freshwater fishes) were classified as threatened (the list of species, and their area of presence, can be found in Table 5 in the Appendix). In Figure 6a we show the spatial distribution density of the species; darker colors are associated to parcels hosting many species, while lighter colors are associated to parcels hosting few species. These species were endangered, due to the presence of four main threats in the catchment: water buffalo (*Bubalis bubalis*), cane toad (*Bufo marinus*), river flow alteration (caused by impoundments, channels for water extractions and levee banks), and grazing land use; however, a given species was not necessarily menaced by all four threats. In Figure 6b we show how the number of threats spatially distributes in the considered area.

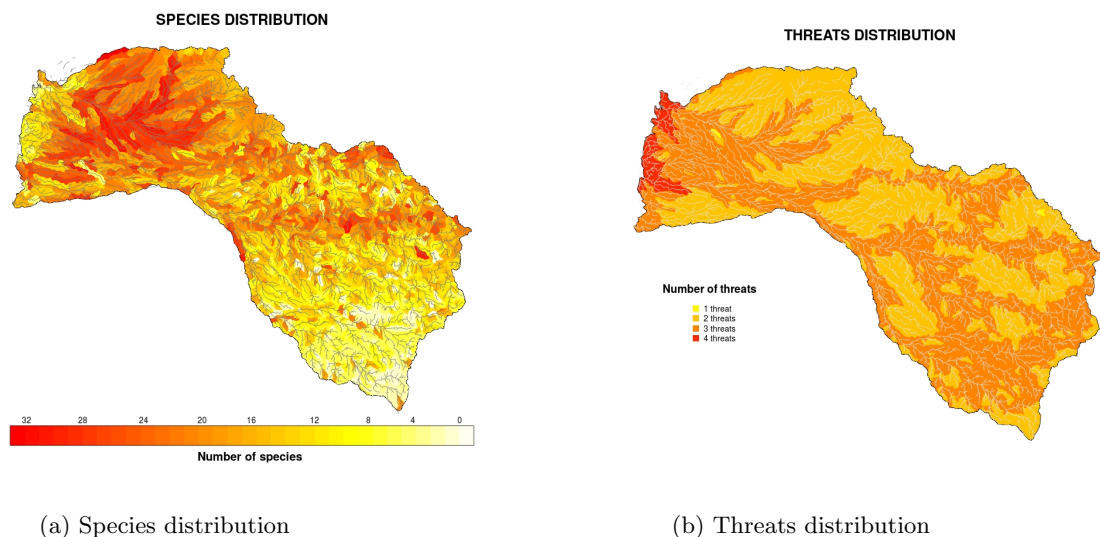


Figure 6: Spatial distribution of species and threats density in the Mitchell river catchment area

Since we had explicit information of which species and which threats co-occurred in each land parcel, the suitability score for a given species $s \in S$ in a given parcel $i \in I$ is given by

$$w_i^s = \alpha_i^s \left(1 - \frac{\#threats_i^s}{\#threats_s + 1} \right)^3,$$

where $\#threats_i^s$ corresponds to the number of threats in i that affect s , $\#threats_s$ corresponds to total number of threats that affect species s , and α_i^s is a binary parameter taking value 1 if species s is hosted

in land parcel i , and 0 otherwise. This expression was defined following the model proposed in Cattarino et al. (2015). Since in this dataset species were not classified according to their level of vulnerability, we considered that all of them were part of S_1 (i.e., $|S_1| = 46$).

Obtained results For the Mitchell river catchment dataset, we solved models GRSC, GRSC-B, GRSC-C and GRSC-CB, using the same algorithmic setting used for the U.S. wildlife dataset. Since $S_2 = \emptyset$, we imposed the buffer area to be a *boundary* of thickness given by $d = 1$ (Scenario C). Due to the structure of the stream network, the underlying graph was comprised by several connected components; this implied that both, the GRSC-C and the GRSC-CB were infeasible for $k = 1$ and $k = 2$, so we considered $k = 3$ in both cases.

Table 4: Number of components, land parcels, solution value and runtime for the best solution for the Mitchel river catchment instance.

<i>inst.</i>	GRSC				GRSC-B				GRSC-C, $k = 3$				GRSC-CB, $k = 3$			
	#c.	#lp.	z^*	$t[s.](g[\%])$	#c.	#lp.	z^*	$t[s.](g[\%])$	#c.	#lp.	z^*	$t[s.](g[\%])$	#c.	#lp.	z^*	$t[s.](g[\%])$
australia	255	402	2866	0.66	5	154	5527	TL (24.37)	3	192	3013	TL (1.28)	3	156	5850	TL (36.37)

In Table 4 (equivalent to Table 3) we report a summary of the results obtained when solving the four models on the described instances; the corresponding solutions are shown in Figure 7. From the spatial point of view, we can observe that the GRSC solution (Figure 7a) is highly fragmented (255 components, and 402 land parcels), which is basically due to the presence of many species distributed along the whole studied area. When requiring the presence of a buffer (GRSC-B), the solution changes substantially (Figure 7b): the designed reserve is comprised by only 5 components (and 154 land parcels in total). The solution obtained when solving the GRSC-C with $k = 3$ (Figure 7c), seems to properly address the fact that species live along water flows, encompassing three relatively long catchment segments. Finally, the solution obtained when solving the GRSC-CB with $k = 3$ (Figure 7d), spans over similar areas as those spanned by the solution obtained GRSC-C with $k = 3$, and it is comprised by almost the same number of land parcels (156 compared to 154). From the conservation point of view, the solutions provided by the GRSC-C and the GRSC-CB (with $k = 3$ in both cases), allow a more effective implementation of conservation strategies (i.e., implementation of measures against the corresponding threats) due to their connectivity and compactness.

From the computational point of view, it is clear that this dataset is much harder than the ones considered before. The two models requiring a buffer boundary (GRSC-B and the GRSC-CB) are considerably more difficult than those that do not require it. As can be seen from Table 4, in these two cases, is not possible to prove optimality within the running time, and the reached gaps are 24.34% and 36.67%, respectively. Although the GRSC-C optimal solution was not found, the solution computed within the time-limit reached a 1.28% gap. Despite the lack of optimality proof, the solutions provided for the different models represent the first attempt to address from a mathematical programming point of view, the conservation planning challenges arising in the Mitchell catchment area (see Cattarino et al. (2016) for a recent reference on the use of heuristic algorithms for addressing a conservation planning problem in the considered region).

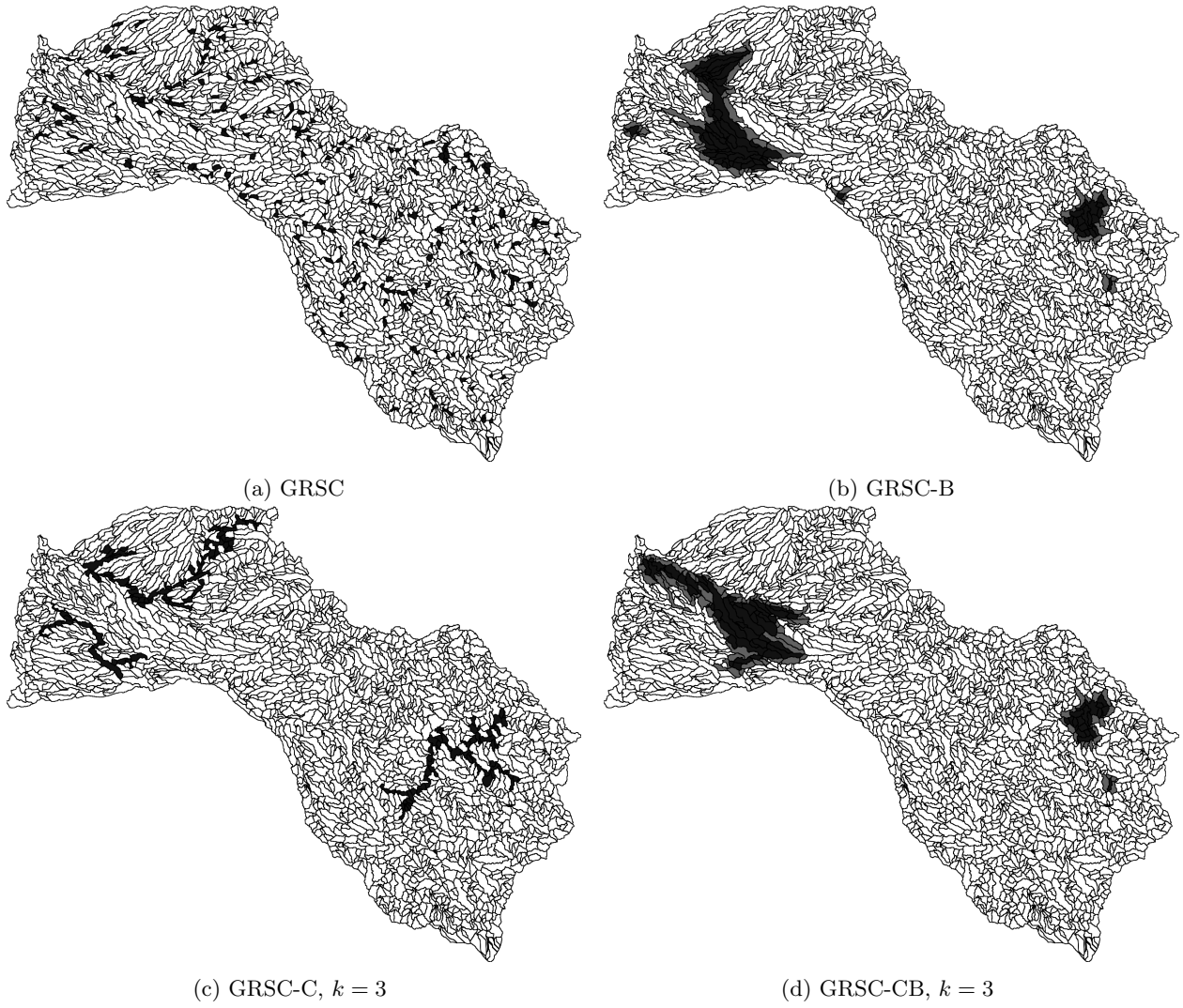


Figure 7: Solutions obtained for the Mitchell river catchment instance considering Scenario C

5 Conclusions

Demographic expansion, natural resource exploitation, and the consequences of climate change, are among the processes that had resulted in a dramatic loss of biodiversity in the last decades. The maintenance of biodiversity is crucial for the survival of humankind (Cardinale et al. 2012). Thus, immense efforts have been devoted in the last decades by international organizations, governments, and foundations, for the establishment of protected areas aiming at ensuring a sustainable landscape for wildlife. In this paper, we introduced the *Generalized Reserve Set Covering Problem with Connectivity and Buffer Requirements (GRSC-CB)*. This problem is an extension of previous modeling approaches for the design of nature reserves. The problem simultaneously considers, for the first time *connectivity requirements*, construction of *buffer zones* and *suitability quotes* for species. All these constraints have been identified as being crucial to the design of useful nature reserves. The GRSC-CB allows to design a reserve comprised of one or more connected components; each of them consisting of a core surrounded by a buffer zone, *satisfying minimum suitability requirements for each species*. We also consider intermediate problems in which only buffer or connectivity constraints are imposed, denoted by GRSC-B and GRSC-C, respectively, and the Generalized Reserve Set Covering Problem (GRSC) in which both, buffer and connectivity constraints are dropped.

We proposed a branch-and-cut framework to solve the GRSC-CB and the remaining three problem variants. The solution framework is enhanced by the use of valid inequalities and also contains a construction and a primal heuristic, and utilizes a local branching scheme to create feasible solutions of good quality. To assess the suitability of our approach, we presented a computational study considering grid graphs, as well as real-life instances representing three different states of the U.S. and a region in northern Australia. The U.S. instances were constructed using data from the National Gap Analysis Program, an initiative of the U.S. Geological Survey (U.S. Geological Survey 2016a,b), with the focus on the protection of mammal species. The Australia instance was constructed using information obtained by the Northern Australia Water Futures Assessment program Department of Agriculture & Water Resources (2012), and corresponds to a conservation setting of fresh water fishes.

In our study, we compared the solutions obtained by using GRSC-CB model against the solutions obtained by using less restrictive models (i.e., GRSC, GRSC-B, and GRSC-C). On the one hand, we showed the effectiveness of the proposed algorithmic framework on synthetic and real-world instances, by providing optimal or high-quality solutions for many instances of realistic size. On the other hand, and more importantly, our study demonstrated the practical versatility of the GRSC-CB (and its variants). The spatial diversity of solutions produced by using the different problem variants showed that connectivity and buffer zones are important features to consider when designing reserves. Thanks to the flexibility of our models, the developed algorithmic framework provides a powerful tool which allows decision-makers to create desired spatial arrangements while responding to different ecological needs.

Acknowledgements

E. Álvarez-Miranda acknowledges the support of the Chilean Council of Scientific and Technological Research (FONDECYT grant N.1180670 and Complex Engineering Systems Institute ICM:P-05-004-F/CONICYT:FB0816), and of the European Union’s HORIZON 2020 research and innovation programme under the Marie Skłodowska-Curie Actions (grant agreement N.691149, SuFoRun Project). The research of M. Sinnl was supported by the Austrian Research Fund (FWF, Project P 26755-N19)

References

- V. Adams, R. Pressey, and J. Álvarez-Romero. Using optimal land-use scenarios to assess trade-offs between conservation, development, and social values. *PLOS ONE*, 11(6):1–20, 06 2016.
- E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The maximum weight connected subgraph problem. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization*, pages 245–270. Springer, 2013a.

- E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The rooted maximum node-weight connected subgraph problem. In C. Gomes and M. Sellmann, editors, *Proceedings of CPAIOR 2013*, volume 7874 of *LNCS*, pages 300–315. Springer, 2013b.
- M. Batisse. The biosphere reserve: A tool for environmental conservation and management. *Environmental Conservation*, 9:101–111, 6 1982.
- M. Batisse. Development and implementation of the biosphere reserve concept and its applicability to coastal regions. *Environmental Conservation*, 17:111–116, 1990.
- P. Beier and R. Noss. Do habitat corridors provide connectivity? *Conservation Biology*, 12(6):1241–1252, 1998.
- H. Beyer, Y. Dujardin, M. Watts, and H. Possingham. Solving conservation planning problems with integer linear programming. *Ecological Modelling*, 328:14–22, 2016.
- A. Billionnet. Designing an optimal connected nature reserve. *Applied Mathematical Modelling*, 36(5):2213–2223, 2012.
- A. Billionnet. Mathematical optimization ideas for biodiversity conservation. *European Journal of Operational Research*, 231(3):514–534, 2013.
- B. Cardinale, J. Duffy, A. Gonzalez, D. Hooper, C. Perrings, P. Venail, A. Narwani, G. Mace, D. Tilman, D. Wardle, A. Kinzig, G. Daily, M. Loreau, J. Grace, A. Larigauderie, D. Srivastava, and S. Naeem. Biodiversity loss and its impact on humanity. *Nature*, 486(7401):59–67, 2012.
- R. Carvajal, M. Constantino, M. Goycoolea, J.P. Vielma, and A. Weintraub. Imposing connectivity constraints in forest planning models. *Operations Research*, 61(4):824–836, 2013.
- L. Cattarino, V. Hermoso, J. Carwardine, M. Kennard, and S Linke. Multi-action planning for threat management: A novel approach for the spatial prioritization of conservation actions. *PLOS ONE*, 10(5):1–18, 05 2015.
- L. Cattarino, V. Hermoso, L. Bradford, J. Carwardine, K. Wilson, M. Kennard, and S. Linke. Accounting for continuous species’ responses to management effort enhances cost-effectiveness of conservation decisions. *Biological Conservation*, 197:116–123, 2016.
- H. Cayton, N. Haddad, N. McCoy, et al. Conservation Corridor, 2017a. URL <http://conservationcorridor.org/>. accessed at 31.01.2017.
- H. Cayton, N. Haddad, N. McCoy, et al. Conservation Corridor: Technical Papers and Methods, 2017b. URL <http://conservationcorridor.org/corridor-toolbox/technical-papers-and-methods/>. accessed at 31.01.2017.
- R. Church, D. Stoms, and F. Davis. Reserve selection as a maximal covering location problem. *Biological Conservation*, 76(2):105–112, 1996.
- M. Clemens, C. ReVelle, and J. Williams. Reserve design for species preservation. *European Journal of Operational Research*, 112(2):273–283, 1999.
- Microsoft Corporation. Conservation at Microsoft, 2016. URL <http://research.microsoft.com/en-us/projects/conservation/>. accessed at 10.06.2016.
- D. Debinski and R. Holt. A survey and overview of habitat fragmentation experiments. *Conservation Biology*, 14(2):342–355, 2000.
- Department of Agriculture & Water Resources. Northern Australia Water Futures Assessment 2009–2012, 2012. URL <http://www.agriculture.gov.au/water/national/northern-australia/northern-australia-water-futures-assessment>. accessed at 10.05.2017.
- Department of Biosciences (U. of Helsinki). C-BIG Conservation Biology Informatics Group, 2016. URL <http://cbig.it.helsinki.fi/research/topics/>. accessed at 08.08.2016.
- B. Dilkina and C. Gomes. Solving connected subgraph problems in wildlife conservation. In A. Lodi, M. Milano, and P. Toth, editors, *Proceedings of CPAIOR 2010*, volume 6140 of *LNCS*, pages 102–116. Springer, 2010.
- C. Drew, Y. Wiersma, and F. Huettmann, editors. *Predictive Species and Habitat Modeling in Landscape Ecology: Concepts and Applications*. Springer, 1st edition, 2011.
- M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1–3):23–47, 2003.
- M. Fischetti, M. Leitner, I. Ljubić, M. Luipersbeck, M. Monaci, M. Resch, D. Salvagnin, and M. Sinnl. Thinning out steiner trees a node-based model for uniform edge costs. *Mathematical Programming Computations*, 9(2):203–229, 2017.
- S. Gergel and M. Turner, editors. *Learning landscape ecology: a practical guide to concepts and techniques*. Springer, 1st edition, 2002.

- Griffith University. Northern Australia Aquatic Ecological Assets 2009-2012, 2012. URL <https://www.griffith.edu.au/environment-planning-architecture/australian-rivers-institute/research/projects2/northern-australia-aquatic-ecological-assets>. accessed at 10.05.2017.
- V. Hermoso, D. Ward, and M. Kennard. Using water residency time to enhance spatio-temporal connectivity for conservation planning in seasonally dynamic freshwater ecosystems. *Journal of Applied Ecology*, 49(5):1028–1035, 2012.
- IUCN. International Union for Conservation of Nature, 2016. URL <http://www.iucn.org/>. accessed at 05.10.2016.
- N. Jafari and J. Hearne. A new method to solve the fully connected reserve network design problem. *European Journal of Operational Research*, 231(1):202–209, 2013.
- N Jafari, B. L. Nuse, C. T. Moore, B. Dilkina, and J. Hepinstall-Cymerman. Achieving full connectivity of sites in the multiperiod reserve network design problem. *Computers & Operations Research*, 81:119–127, 2017.
- K. Kaparis and A. Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical Programming, Series B*(124):69–91, 2010.
- I. Lacher and M. Wilkerson. Wildlife connectivity approaches and best practices in u.s. state wildlife action plans. *Conservation Biology*, 28(1):13–21, 2014.
- D. Lindenmayer and J. Franklin, editors. *Conserving Forest Biodiversity A Comprehensive Multiscaled Approach*. Island Press, 1st edition, 2002.
- V. Meretsky, L. Maguire, F. Davis, D. Stoms, J. Scott, D. Figg, D. Goble, B. Griffith, S. Henke, J. Vaughn, and S. Yaffee. A state-based national network for effective wildlife conservation. *BioScience*, 62(11):970–976, 2012.
- J. Millsbaugh and F. Thompson, editors. *Models for Planning Wildlife Conservation in Large Landscape*. Elsevier, 1st edition, 2008.
- E. Minor and T. Lookingbill. A multiscale network analysis of protected-area connectivity for mammals in the united states. *Conservation Biology*, 24(6):1549–1558, 2010.
- K. Öhman and T. Lämås. Reducing forest fragmentation in long-term forest planning by using the shape index. *Forest Ecology and Management*, 212(1–3):346–357, 2005.
- H. Önal and R. Briers. Incorporating spatial criteria in optimum reserve network selection. *Proceedings of the Royal Society of London B: Biological Sciences*, 269(1508):2437–2441, 2002.
- H. Önal and R. Briers. Selection of a minimum boundary reserve network using integer programming. *Proceedings of the Royal Society of Londn B: Biological Sciences*, 270(1523):1487–1491, 2003.
- H. Önal, Y. Wang, S. Dissanayake, and J. D. Westervelt. Optimal design of compact and functionally contiguous conservation management areas. *European Journal of Operational Research*, 251(3):957–968, 2016.
- Oregon Department of Fish and Wildlife. Threatened, Endangered, and Candidate Fish and Wildlife Species, 2015. URL http://www.dfw.state.or.us/wildlife/diversity/species/threatened_endangered_candidate_list.asp. accessed at 08.02.2017.
- S. Polasky, J. Camm, and B. Garber-Yonts. Selecting biological reserves cost-effectively: An application to terrestrial vertebrate conservation in oregon. *Land Economics*, 77(1):68–78, 2001.
- R. Pressey, H. Possingham, and J. Day. Effectiveness of alternative heuristic algorithms for identifying indicative minimum requirements for conservation reserves. *Biological Conservation*, 80(2):207–219, 1997.
- M. Schwartz. Choosing the appropriate scale of reserves for conservation. *Annual Review of Ecology and Systematics*, 30:83–108, 6 1999.
- J. Scott, F. Davis, B. Csuti, R. Noss, B. Butterfield, C. Groves, H. Anderson, S. Caicco, F. D’erchia, T. Edwards, T. Ulliman, and R. Wright. Gap analysis: A geographic approach to protection of biological diversity. *Wildlife Monographs*, (123):3–41, 1993.
- R. St. John, S. Tóth, and Z. Zabinsky. Optimizing the geometry of wildlife corridors in conservation reserve design. *Forthcoming in Operations Research*, 2018.
- R. Stewart, I. Ball, and H. Possingham. The effect of incremental reserve design and changing reservation goals on the long-term efficiency of reserve systems. *Conservation Biology*, 21, 2007.
- H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Math. Japonica*, 24(6):573–577, 1980.
- US Fish and Wildlife Service. Federally Listed and Proposed Species in Pennsylvania, 2016. URL https://www.fws.gov/northeast/pafo/pdf/endspecies/T&E_list_02172016.pdf. accessed at 08.02.2017.

- U.S. Geological Survey. National Gap Analysis Program, 2016a. URL <http://gapanalysis.usgs.gov/>. accessed at 08.02.2017.
- U.S. Geological Survey. Gap Factsheet, 2016b. URL <http://pubs.usgs.gov/fs/2013/3087/pdf/fs2013-3087.pdf>. accessed at 08.02.2017.
- Vermont Fish & Wildlife Department. Endangered and Threatened Animals of Vermont, 2015. URL <http://www.vtfishandwildlife.com/common/pages/DisplayFile.aspx?itemId=268519f>. accessed at 08.02.2017.
- Y. Wang and H. Önal. Designing connected nature reserve networks using a graph theory approach. *Acta Ecologica Sinica*, 31(5):235–240, 2011.
- Y. Wang and H. Önal. Optimal design of compact and connected nature reserves for multiple species. *Conservation Biology*, 30(2):413–424, 4 2016.
- J. Williams. Optimal reserve site selection with distance requirements. *Computers & Operations Research*, 35(2):488–498, 2008.
- J. Williams and C. ReVelle. A 0-1 programming approach to delineating protected reserves. *Environment and Planning B: Planning and Design*, 23(5):607–624, 1996.
- J. Williams and C. ReVelle. Reserve assemblage of critical areas: A zero-one programming approach. *European Journal of Operational Research*, 104(3):497–509, 1998.
- J. Williams, C. ReVelle, and S. Levin. Using mathematical optimization models to design nature reserves. *Frontiers in Ecology and the Environment*, 2:98–105, 2004.
- J. Williams, C. ReVelle, and A. Levin. Spatial attributes and reserve design models: A review. *Environmental Modeling & Assessment*, 10(3):163–181, 2005.

Appendix 1: Additional information of Mitchell river catchment

Table 5: Detailed information of the endangered species included in the Mitchell river catchment area.

Name of species	Area (km ²)
Scleropages jardinii	26130.2
Nematalosa erebi	34153.3
Thryssa scratchleyi	17161
Neoarius berneyi	21077
Neoarius graeffei	8832.2
Neoarius leptaspis	10920.3
Neoarius paucus	45154.8
Anodontiglanis dahli	22921.2
Neosilurus ater	32947.6
Neosilurus hyrtlii	26560.3
Porochilus rendahli	17874.5
Arramphus sclerolepis	18386.5
Zenarchopterus spp.	10130.7
Strongylura krefftii	25112.6
Craterocephalus stercusmuscarum	54071.5
Iriatherina weneri	1639.4
Melanotaenia splendida inornata	70157.5
Pseudomugil tennellus	2118.9
Ophisternon spp.	26898.5
Ambassis sp.	778.9
Ambassis agrammus	8789.8
Ambassis macleayi	51412
Denarius bandata	11330
Lates calcarifer	22966.9
Amniataba percoides	64519
Hephaestus carbo	10098.4
Hephaestus fuliginosus	64041.6
Variichthys lacustris	365.7
Leiopotherapon unicolor	65926.9
Scortum ogilbyi	60007.9
Glossamia aprion	52607.2
Toxotes chatareus	45386.6
Glossogobius aureus	40946.1
Glossogobius giuris	950.8
Glossogobius sp. 2	24460
Hypseleotris compressa	370.7
Mogurnda mogurnda	14594.7
Oxyeleotris lineolatus	64179.9
Oxyeleotris selheimi	60793.9
Synaptura salinarum	3218.8
Synaptura selheimi	12046.5
Megalops cyprinoides	10908.8