Geometric Optimization of Wildlife Corridors on Grid-Based Landscapes Using Integer Programming

Chao Wang*, Jorge A. Sefair+✉, Robert J. Fletcher Jr.†
*School of Computing and Augmented Intelligence, Arizona State University, USA, cwang198@asu.edu
+✉ Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, USA, jorge.sefair@ise.ufl.edu*
†Department of Zoology, Cambridge University, UK, rf497@cam.ac.uk

Conservation corridors play a critical role in mitigating biodiversity loss by linking fragmented habitats. A central challenge in designing these corridors lies in satisfying ecological requirements through geometric constraints such as connectivity, redundancy, width, and size. In this study, we propose a mixed-integer programming model that integrates these geometric features into the corridor design process, with the objective of maximizing corridor utility over a grid-based landscape. To enhance computational performance, we develop a Branch-and-Cut algorithm supported by a bound-improving heuristic, valid inequalities, and a variable reduction preprocessing routine. We evaluate our approach on a real-world landscape aimed at protecting the Florida panther, as well as on a suite of computer-generated landscapes. Results show that the algorithm can efficiently produce near-optimal solutions for instances with up to 3,000 patches within a two-hour time limit.

Key words: OR in natural resources, wildlife corridor design, conservation planning, Branch-and-Cut, integer programming

1. Introduction

Biodiversity loss is a contemporary environmental issue facing the global community. The ongoing degradation of ecosystems has consequences for human systems, including increased disease transmission (e.g., H1N1 and COVID-19), the spread of invasive species, deterioration of water resources, and the intensification of extreme climate events, among others (Cormen 2009, Wood et al. 2014, Mace et al. 2012). While biodiversity loss is driven by multiple factors, the primary causes are habitat destruction and over-exploitation of species, both linked to human population growth and increased resource consumption (Barlow et al. 2016). Habitat reduction, predominantly induced by human activities such as agriculture and urban expansion, negatively impacts the survival and reproduction of protected and endangered species (Rodrigues et al. 2004b). As a result, habitat loss remains the principal threat to global biodiversity, affecting 85% of all species on the

*✉Corresponding author

International Union for Conservation of Nature's Red List of Threatened Species (IUCN 2021). According to the 2020 Living Planet Report by the World Wildlife Fund, the global population of birds, amphibians, mammals, fish, and reptiles has declined by an average of 68% since 1970 (WWF 2020). Although approximately 15% of the world's land area is currently under some form of protection (Belle et al. 2018), this coverage is insufficient due to significant gaps and increasing threats (Rodrigues et al. 2004a). Recent studies call for a substantial increase in protected lands by 2050 to maintain current rates of resource extraction and slow the rate of species extinction (Watson and Venter 2017, Dinerstein et al. 2019). This situation requires the attention not only of biologists and environmentalists but also of professionals from other fields who can contribute to the development of effective conservation strategies.

The design and implementation of ecological corridors are essential strategies for mitigating the negative impacts of habitat loss (Sefair 2022). Corridors play a vital role in maintaining or restoring connectivity between fragmented habitats, promoting genetic diversity by enabling species movement between protected areas. This movement may result from seasonal migration toward breeding or feeding sites, or as a response to habitat disturbances. The *effectiveness* of a corridor depends not only on the quality of the habitats it connects but also on its geometry. The *width* of a corridor is essential for isolating protected areas and creating a buffer that limits species interactions outside the corridor. However, corridors that are too narrow may provide low-quality habitats dominated by edge effects, while corridors that are too wide may fail to serve their primary purpose of facilitating movement between areas. Additionally, overly wide corridors can present significant challenges in land-use planning and implementation.

Efforts have been made to establish guidelines for corridor width (Beier 2019) and to promote corridor redundancy, defined as the presence of multiple alternative corridors connecting the same habitat patches (McRae et al. 2008). Redundancy can be achieved in various ways, such as by creating multiple distinct corridors or by splitting corridors to offer alternative routes between habitats (Pinto and Keitt 2009). Redundancy enhances the resilience of conservation plans in response to changes in land use. The *length* and *size* of a corridor are also critical for its performance. While the maximum distance an individual can travel between habitats varies by species, longer corridors may increase mortality rates. Shorter corridors are preferred due to their lower cost. However, existing land use can limit the design options, sometimes requiring the construction of longer corridors to reconnect fragmented habitats.

Our model produces land conservation recommendations for the design of a corridor connecting two fragmented habitats. The study area is discretized into equal-sized square patches, each of which can be either selected for inclusion in the corridor or excluded from it. The objective is to optimize a measure of corridor quality defined as the aggregation of individual patch utilities

(Downs and Horner 2012, Conrad et al. 2012). Patch utility can be represented by usage probability, reflecting its significance to the species (Sefair et al. 2017, Fletcher Jr. et al. 2019). Alternative approaches for assessing patch utility include the Species Threat Abatement and Recovery (STAR) metric, ecological niche models, and criteria based on patch features (Mair et al. 2021, Warren and Seifert 2011, Peterson et al. 2008). The corridor is required to satisfy the following design requirements: (i) *Connectivity*—the corridor must connect the two fragmented habitats through a connected sequence of selected patches, ensuring that no selected patch is isolated from the corridor; (ii) *Width*—the corridor must maintain a minimum width along its entire span, including at any bifurcations; (iii) *Limited size*—the total number of selected patches (i.e., corridor size) cannot exceed a pre-specified limit; and (iv) *Budget*—the total acquisition (or restoration) cost of the selected patches cannot exceed a pre-specified budget. We denote the problem of finding a corridor of maximum utility subject to the specified spatial and operational design constraints as w-MUC.

We present a mixed-integer programming (MIP) formulation for this wildlife corridor design problem and propose a tailored Branch-and-Cut (B&C) procedure to solve it. Conventional commercial solvers (e.g., CPLEX, Gurobi) are limited to small instances, so we develop an acceleration scheme that integrates cutting planes, a bound-improving heuristic, a warm-start procedure, and a pre-processing step for variable reduction. Unlike existing methods, our approach introduces a utility-based patch selection strategy, where the quality of each candidate patch reflects local spatial attributes. The model also offers parameterized control over corridor geometry, enabling the design of corridors with variable widths and sizes. Additionally, it supports corridor redundancy by allowing multi-branched configurations, with each branch satisfying the specified geometric constraints. Our method can solve instances with over 3,000 patches to near optimality within two hours on a standard desktop machine, demonstrating both its scalability and practical applicability.

The structure of this paper is as follows: Section 2 provides an overview of the relevant literature and contextualizes our contribution. Section 3 defines a corridor in our context and introduces the necessary mathematical notation. Section 4 presents the proposed MIP model, while Section 5 describes the solution methodology. Section 6 reports the results of computational experiments, including a case study on corridor design for the Florida Panther. Section 7 offers concluding remarks and suggests directions for future research.

2. Literature review

Systematic conservation planning seeks to identify, configure, and maintain ecological areas that promote long-term biodiversity persistence (Pressey et al. 2007). To support these objectives, a wide range of mathematical models has been developed to address challenges such as reserve and

corridor design, identification of ecologically vulnerable areas, mitigation of landscape fragmentation, and invasive-species management (Billionnet 2013, Costello and Polasky 2004, Sefair et al. 2017). Within this broad domain, a central research focus concerns the *prioritization* of conservation patches under limited resources. Landscape prioritization analyses leverage diverse analytical tools—including data science, mathematical optimization, and probabilistic modeling—to evaluate candidate areas based on criteria such as population frequency, species representation, presence–absence and patch-use probabilities, dynamic land prices, and related ecological and economic indicators (Burgman et al. 2005, Marianov et al. 2008, Church et al. 1996, Camm et al. 2002, Haight et al. 2000, Tóth et al. 2011, Geldmann et al. 2013, Fletcher Jr. et al. 2019).

Spatial metrics are central to systematic conservation planning because they translate ecological design principles into operational criteria for selecting and configuring protected areas under limited budgets (Pressey et al. 2007). In practice, planners account for the spatial arrangement of selected patches—such as connectivity, width, compactness, and size—since the ecological performance of a reserve network or corridor depends not only on which patches are chosen, but also on how they are configured. This view is consistent with foundational planning principles that call for explicit, measurable objectives and clearly defined design criteria for reserve networks (Margules and Pressey 2000), motivating the direct incorporation of geometric considerations into reserve and corridor design analyses.

Connectivity has become a central concept in landscape prioritization, reflecting that fragmented habitats are often insufficient to sustain dispersal, gene flow, and migration. Its ecological importance and sensitivity to environmental change and landscape configuration (Hodgson et al. 2009) have motivated diverse connectivity metrics, including structural connectivity (driven by spatial arrangement) and functional connectivity (driven by movement and behavior) (Williams et al. 2005, Kindlmann and Burel 2008, Tóth et al. 2009, Beger et al. 2010). While ensuring connectivity, exploratory analyses heuristically screen the landscape to identify paths or corridors with potential redundancy. Examples include least-cost path methods, which identify a single minimum-resistance route across a raster landscape (Kautz et al. 2006), as well as least-cost corridor and graph-theoretic approaches, which broaden this perspective by identifying wider paths and alternative routes (Pinto and Keitt 2009). More recently, probabilistic models have been used to evaluate landscape connectivity by predicting short- and long-term movement, species distribution, and mortality (Fletcher Jr et al. 2022). However, these exploratory approaches do not explicitly optimize patch selection to satisfy geometric and implementation requirements under a defined corridor performance metric (e.g., total ecological utility).

Because animals use corridors for essential behaviors such as foraging, territory patrol, and movement between feeding and breeding areas, ecological corridor models have evolved to incorporate

a broader set of realistic factors that influence connectivity. These include road network planning and the placement of wildlife crossings (Downs and Horner 2012, Yemshanov et al. 2022), the impact of prescribed burns and wildfire control on habitat structure (Matsypura et al. 2018, Rachmawati et al. 2018), species-specific mobility constraints (Phillips et al. 2008), vegetation dynamics (Perkl 2016), and multi-species spatial requirements (Lai et al. 2011). More recently, interactive conservation planning systems that integrate visual analytics and user interaction with optimization outputs have been developed to support decision-making in complex, data-rich environments (Zhang et al. 2021). Although these extensions fall outside the scope of the present paper, they represent important modeling dimensions that should be considered in practical corridor design and provide promising directions for future research building on exact optimization frameworks.

Connectivity alone does not ensure ecological suitability. Additional geometric requirements—such as minimum width (Beier 2019), bounded size (Tóth et al. 2009), and length limits—are often necessary to reduce edge effects and maintain functional habitat. To enforce connectivity and related spatial properties, exact formulations based on integer programming have been developed for reserve and corridor design. Billionnet (2013) introduced stylized mathematical models that integrate connectivity through network-flow structures with patch selection and budget constraints. Set-covering and connected-reserve formulations ensure contiguous habitat networks (Cerdeira and Pinto 2005, Önal and Briers 2006), with extensions incorporating compactness, multi-species considerations, and full connectivity requirements (Önal et al. 2016, Jafari and Hearne 2013, Shirabe 2005). Additional exact approaches include connected-subgraph formulations (Dilkina and Gomes 2010, Conrad et al. 2012), geometry-explicit path-planning models (St John et al. 2018), models for the identification of single-path corridor boundaries (Wang et al. 2022), and branch-and-cut frameworks that jointly design connected reserves and buffer zones (Álvarez-Miranda et al. 2021). Beyond conservation planning, related spatial optimization models imposing geometric constraints have been applied in harvest scheduling (Mcdill et al. 2002, Goycoolea et al. 2005, Öhman and Wikström 2008), forest management (Carvajal et al. 2013), wildfire management (Matsypura et al. 2018, Rachmawati et al. 2018), and political districting (Validi et al. 2022).

Table 1 summarizes the approaches most closely related to our work and highlights their coverage of key modeling features. In the table, ✓ indicates that the method explicitly enforces the corresponding feature; ∼ indicates partial control, heuristic enforcement, or indirect representation without formal guarantees; and — indicates that the feature is not explicitly addressed. The table shows that connectivity modeling has a well-established methodological foundation in reserve design that extends naturally to corridor applications. However, the incorporation of additional corridor-specific geometric requirements remains relatively limited. Reserve design models mostly

focus on connectivity and budget constraints, but are not designed to include explicit controls for spatial features such as minimum width or multi-branch structures.

Corridor-specific models address some of these limitations. Connected subgraph approaches (Dilkina and Gomes 2010, Conrad et al. 2012) jointly consider utility and connectivity and may allow for limited spatial branching. However, corridor width is indirectly determined by spatial resolution rather than enforced through explicit constraints, providing no guarantee of achieving a prescribed minimum width. Geometry-explicit path-planning models—the closest to the present study—(St John et al. 2018, Wang et al. 2022) directly enforce minimum width and maximum length requirements, but are mathematically structured to delineate a single continuous corridor, precluding redundancy (multi-branch) analysis. Moreover, although these models are formulated for irregular patches, their implementation entails substantial computational effort for parameter calculation and large-scale model size, even when applied to regular grid landscapes.

Table 1 Corridor design features and related literature

Method	Reference	Connectivity	Width	Length &Size	Redundancy (multi-branch)	Budget & Utility
Exploratory analysis						
Least-cost path (LCP)	Kautz et al. (2006)	✓	—	—	—	∼
Least-cost corridor (LCC)	Pinto and Keitt (2009)	✓	∼	—	∼	∼
Reserve design						
Connectivity via cuts (IP)	Carvajal et al. (2013)	✓	—	—	—	✓
Connected set covering (IP)	Cerdeira and Pinto (2005)	✓	—	—	—	✓
Connected reserve (IP)	Önal and Briers (2006)	✓	—	—	∼	✓
Compact + connected (multi-species) (IP)	Önal et al. (2016)	✓	—	✓	—	✓
Flow-based connectivity (MIP)	Jafari and Hearne (2013)	✓	—	—	—	✓
Flow-based connectivity (MIP)	Shirabe (2005)	✓	—	—	—	✓
Connectivity + buffer zone (IP)	Álvarez-Miranda et al. (2021)	✓	—	—	—	✓
Corridor design						
Flow-based and probabilistic connectivity (IP, MIP)	Billionnet (2013)	✓	—	—	—	✓
Connected subgraph (MIP)	Dilkina and Gomes (2010)	✓	—	—	∼	✓
Connected subgraph (MIP)	Conrad et al. (2012)	✓	∼	∼	∼	✓
Geometric path planning (IP)	St John et al. (2018)	✓	✓	✓	—	∼
Corridor boundary (IP)	Wang et al. (2022)	✓	✓	✓	—	✓
Flow-based Ω-width (MIP)	**This paper**	✓	✓	✓	✓	✓

To the best of our knowledge, this is the first unified integer-programming framework for grid-based landscapes that simultaneously enforces connectivity, minimum width, bounded corridor size, and budget constraints, while allowing *multi-branch corridors* that provide redundancy. The model is solved using a tailored branch-and-cut algorithm that incorporates problem-specific valid inequalities, warm starts, local search, and pre-processing procedures, resulting in scalable performance on both realistic and synthetic landscapes. Moreover, the formulation readily generalizes

to other regular tessellations (e.g., hexagonal or triangular) through an appropriate redefinition of adjacency relations.

3. Preliminaries

3.1. Corridor utility

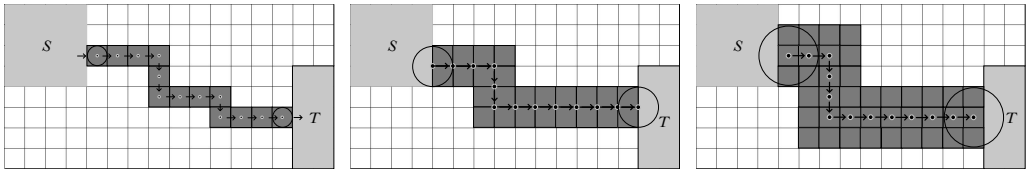
We *partition* the landscape into a set of equal-sized squares forming a grid, denoted by P (Conrad et al. 2012). Each patch represents an area through which species can move, with accessibility to and from its neighboring patches. We define a corridor $\mathcal{C}$ as a set of patches that connects a set of origin patches, S, to a set of destination patches, T, where $S \cap T = \emptyset$, ensuring that at least one path in $\mathcal{C}$ links each patch $i \in S$ to at least one patch $j \in T$. The utility of the corridor, denoted by $U_{\mathcal{C}}$, serves as a measure of its quality, representing the likelihood that individuals starting in S will use patches within $\mathcal{C}$ to reach T. Patch utility can be assessed through various metrics, such as species abundance, resource availability, the STAR index, ecological niche modeling, or usage probability (Mair et al. 2023, Préau et al. 2020, Marx et al. 2020, Fletcher Jr. et al. 2019, Sefair et al. 2017, Acevedo et al. 2015). We define $U_{\mathcal{C}} = \sum_{i \in \mathcal{C} \setminus (S \cup T)} f_i$, where $f_i > 0$ represents the utility of patch i.

3.2. Spatial properties of a corridor

We define two types of *reference* points in the discretized landscape: *centroids* and *corners*. Centroids are the geometric centers of each patch, while corners are the points where three or more patches intersect. The set P also describes the set of centroids, as there is a one-to-one correspondence between patches and their centroids. The set E represents the corners. The set of neighbors of a reference point i, denoted by $\mathcal{N}_i$, consist of the adjacent centroids (left, right, up, and down) if i is a centroid, or the neighboring corners sharing a boundary if i is a corner. Figure 2a illustrates a grid landscape and the neighborhood of centroid i. Although we focus on a square grid discretization, our models and methods can be extended to other regular tessellations, such as hexagons (Prasad and Borges 2006) and triangles (St John et al. 2018) (see Appendix A for illustrations).

We design corridors to connect two fragmented habitats with predefined origin and destination patches. Given its ecological importance, our model enforces a corridor width w, defined as the diameter of a circle that can travel from the origin to any patch in the corridor, including the destination patches, while overlapping only patches in the corridor and not intersecting any non-corridor patches. If each patch has dimensions $\alpha \times \alpha$, a corridor of width w accommodates a circle of diameter $w \times \alpha$. Figures 1a–1c illustrate this concept with corridors of widths $w = 1$, $w = 2$, and $w = 3$, respectively, connecting habitat patches S and T.

Figure 1 illustrates that a corridor of width w can be fully defined by a sequence of reference points. If w is even, the corridor is characterized by a sequence of corner points, whereas an odd

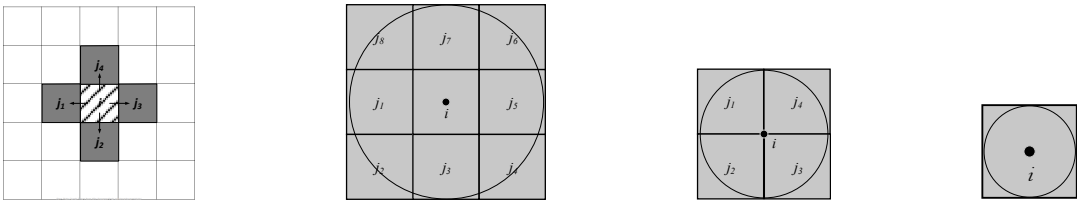


(a) $w = 1$, 13 patches (b) $w = 2$, 24 patches (c) $w = 3$, 39 patches

Figure 1 Corridors samples with different width (w)

value of w corresponds to a sequence of centroids. To enforce the corridor width, we define the set $\Omega_i(w)$ for each reference point i, which includes all patches overlapping with a circle of diameter w centered at i. Figure 2 depicts the Ω-sets for different reference points and corridor widths. For $w = 1$, the Ω-set of centroid i is $\Omega_i(1) = \{i\}$ and for $w = 3$, it expands to $\Omega_i(3) = \{j_1, \ldots, j_8\}$. When $w = 2$, the Ω-set for corner point i contains the four adjacent patches (i.e., $\Omega_i(2) = \{j_1, \ldots, j_4\}$).

The sequence of reference points defining a corridor is referred to as its *backbone*. Both the backbone and the Ω-set definitions are fundamental to our optimization model, as finding the optimal backbone is equivalent to identifying the optimal corridor for a given width. The parameter $\mathcal{L}$ denotes the maximum number of selected patches allowed in the corridor (i.e., the maximum corridor size). Limiting the number of selected patches can serve as a proxy for controlling corridor length. Figure 1 illustrates corridors with $w = 2$ and $\mathcal{L} = 13$, 24, and 39, respectively.



(a) Grid Landscape (b) $w = 3$ (c) $w = 2$ (d) $w = 1$

Figure 2 Grid landscape and $\Omega_i(w)$-set as a function of corridor width

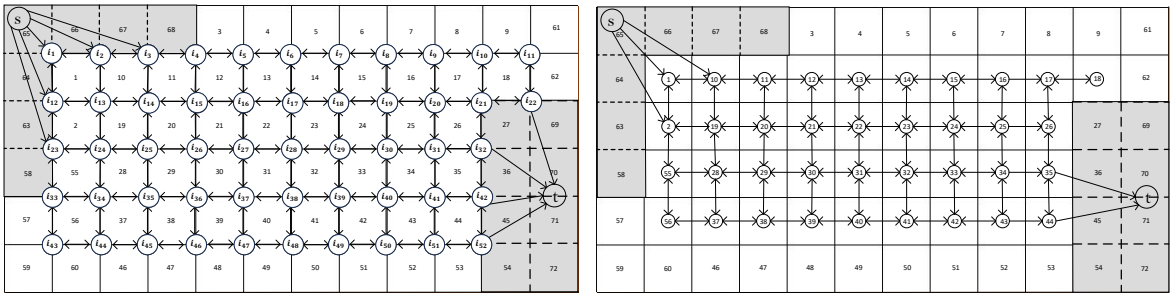
4. Optimal corridor design

4.1. Mathematical programming model

This section describes a mixed-integer program to design a maximum utility corridor that satisfies budget and spatial constraints such as connectivity, minimum width, and maximum size (w-MUC). We define a directed network $G^w = (N, A)$, where N is the set of reference points (i.e., *centroids* or *corners*) augmented with two additional nodes, s and t, to represent the origin and destination habitats S and T that need to be connected, respectively. Parameter w is the minimum required width of the corridor. If w is an odd number, then the reference points in N are the patch *centroids* (i.e., $N = P \cup \{s, t\}$) and if w is even, then they are the corner points (i.e., $N = E \cup \{s, t\}$). The

set of arcs is given by $A = \{(i,j) \in N \times N : j \in \mathcal{N}_i\}$. The set $\mathcal{N}_s$ consists of reference points whose Ω-set overlaps with (when $w \geq 2$) or is adjacent to (when $w = 1$) patches in S, whereas $\mathcal{N}_t = \emptyset$ by definition. For any given width w and $i \in N$, then we have $\Omega_i(w) \subseteq P$. That is, we eliminate reference points whose corresponding Ω-set contains patches outside the landscape.

The total budget available is given by b and the cost of acquiring (or restoring) patch $i \in P$ is denoted by c_i. Figure 3 illustrates the construction of G^w and its components when w is even and odd. Figure 3a shows a network when $w = 2$. Nodes s and t represent the two habitats in gray, and the remaining nodes are candidates for the corridor backbone. In this example, Nodes i_1, i_2, i_3, i_{12}, and i_{23} are of particular importance because their Ω-sets have sufficient overlap with the habitat to enforce the required width when leaving habitat S. For example, there is no arc from s to node i_4 because the overlap with habitat S is not enough to enforce the corridor width. We assume that the species dispersal is completed once the destination habitat T is reached through any node among i_{22}, i_{32}, i_{42} and i_{52}. Figure 3b shows a network G^w when $w = 3$. Nodes 1, 2, and 10 are connected to s because their Ω-sets allow a circle with diameter $w = 3$ to travel out of habitat S. Similar to Figure 3a, the dispersal is completed once the species enter the destination habitat T through Nodes 35 or 44.



(a) $w = 2$ (b) $w = 3$

Figure 3 Augmented networks for even and odd values of parameter w

Following the notation from Section 3, f_i represents the utility value of patch i, $\forall i \in P$. An optimal corridor connects s and t while guaranteeing a width of at least w and a maximum size $\mathcal{L}$, and using a budget of no more than b, such that the sum of the chosen patches' utility is maximized. We use the variable y_{ij} to determine whether adjacent backbone nodes i and j belong to the corridor and the binary variable x_k to indicate that patch k is selected to be part of the corridor (i.e., $x_k = 1$) or otherwise (i.e., $x_k = 0$). The mixed-integer program in (1) captures the main features of the MUC problem.

$$\max \quad z(w) = \sum_{k \in P \setminus (S \cup T)} f_k x_k \tag{1a}$$

$$\text{s.t.} \quad \sum_{j:(i,j)\in A} y_{ij} - \sum_{j:(j,i)\in A} y_{ji} = \begin{cases} 1, & \text{if } i = s \\ -1, & \text{if } i = t \\ 0, & \text{if } i \in N \setminus \{s,t\} \end{cases} \tag{1b}$$

$$y_{ij} \leq x_k, \ \forall (i,j) \in A, \ k \in \Omega_j(w), \tag{1c}$$

$$\sum_{k \in P \setminus (S \cup T)} x_k \leq \mathcal{L}, \tag{1d}$$

$$\sum_{k \in P \setminus (S \cup T)} c_k x_k \leq b, \tag{1e}$$

$$y_{ij} \in [0,1], \ \forall (i,j) \in A, \tag{1f}$$

$$x_k \in \{0,1\}, \ \forall k \in P \setminus (S \cup T). \tag{1g}$$

The objective function in (1a) maximizes the total utility value of selected patches. Constraints (1b) enforce the corridor connectivity by using a flow-based formulation that creates at least one path of consecutive reference points from s to t. The *covering* Constraints (1c) guarantee that if there is a positive flow between reference points i and j, then all patches in the Ω-set of point j must be selected for the corridor to enforce the minimum required width. Constraint (1d) indirectly controls the length of the corridor by limiting the total number of patches in the corridor, which is set to be no more than $\mathcal{L}$. The budget constraint (1e) enforces the maximum amount of resources available for patch purchases or restoration. Constraints (1f) and (1g) enforce the nature of the decision variables. Formulation (1) includes new features that are worth emphasizing:

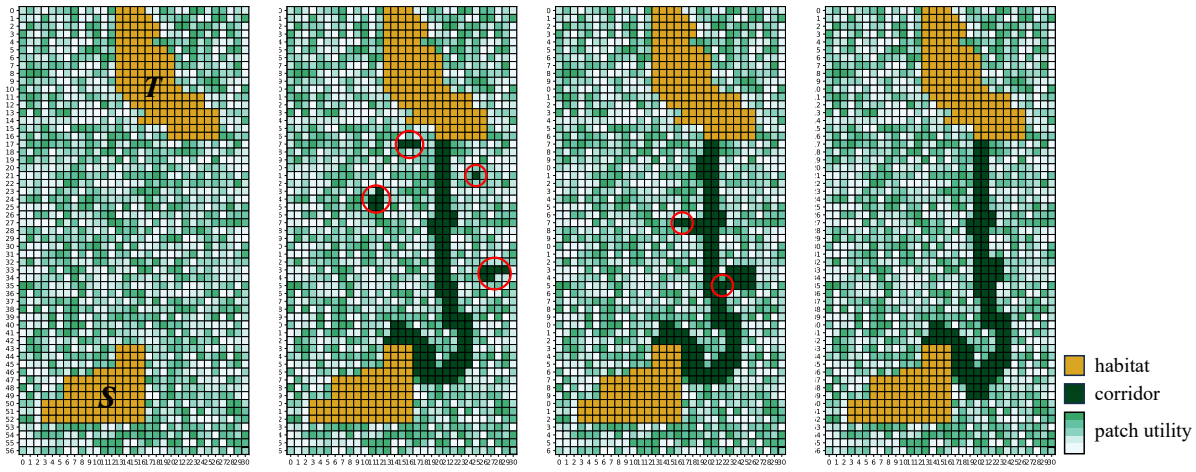
- Constraints (1b) enforce a directed flow from s to t in order to create the corridor's sequence of backbone reference points. These constraints admit loops, allowing nearby good-quality areas to be added to the main stem of the corridor.
- Flow decision variables y need not be integer, as they are only used to route flow from node s to t. Indeed, fractional flows may indicate splits in the corridor into two or more sub-corridors, each of them satisfying the width requirements from Constraints (1c).
- The formulation accommodates general tessellation patterns on regular landscapes, including triangular, square, and hexagonal grids. Appendix A provides additional illustrative examples.

Formulation (1) can be applied to any landscape. The required *input data* from the user consist of the set of patches P (i.e., grid) and the width parameter w, from which the graph G^w is constructed. The nodes s and t in G^w, representing the fragmented habitats, must be identified, along with the corresponding Ω-set for each node. For each patch $i \in P \setminus \{S \cup T\}$, the model requires an associated cost c_i and utility f_i, as well as a budget limit b and a corridor size limit $\mathcal{L}$.

The corridor design problem studied in this paper is related to reward-collecting routing problems, most closely to the *orienteering problem* (Vansteenwegen et al. 2011, Gunawan et al. 2016) and its *open* variant (Chaigneau et al. 2025), where the route is not required to return to the

starting location. However, our problem exhibits important structural differences. In the corridor problem, both costs and utilities (i.e., rewards) are node-dependent and incurred only once; that is, revisiting a node does not generate additional cost. In contrast, in the orienteering problem, costs are arc-specific, and revisiting a node incurs additional cost. A further key distinction is the width requirement, under which the cost and utility of selecting a node depend on the set of nodes "covered" by the corridor, rather than solely on the parameters of the individual node visited. Although this may appear subtle, it introduces additional complexity, as nodes may be covered multiple times depending on the selected path but costs are incurred only once (see Appendix C), reflecting patches that have already been acquired. Finally, the corridor problem allows multi-branch corridors, which may require nodes (other than the origin or destination) to be visited multiple times on different paths starting from the origin, a feature not permitted in the orienteering problem.

We emphasize that Formulation (1) does not rely on a grid-based landscape and can be extended to other discretizations, as discussed in Section 3.2 and Appendix A. Width enforcement via Ω-sets and corridor directionality via flow conservation require only a definition of centroids or corners and their corresponding adjacency relations. For hexagonal and triangular partitions, backbone nodes and Ω-sets are defined analogously—using centroids for odd widths and corners for even widths, with adjacency determined by shared boundaries—ensuring that all constraints remain valid. Appendix A provides construction details and illustrative examples, including the Ω-sets for $w = 1, 2, 3$ in hexagonal and triangular tessellations.



(a) Input landscape (b) Corridor with disconnected patches, $w = 2$ (c) Corridor with disconnected flow and backbone , $w = 2$ (d) Feasible corridor, $w = 2$

Figure 4 Constraint visualization

4.2. Imposing patch connectivity

Formulation (1) identifies a set of patches producing the maximum possible utility while connecting habitats S and T. However, its solution may have disconnected components in addition to a corridor connecting habitats S and T. In Figure 4a, the highlighted patches (in yellow) represent S and T, respectively. Figure 4b shows a possible solution to Formulation (1) that includes a corridor connecting S and T (dark green color), and additional disconnected patches outside of the corridor (in red circles). Disconnected patches are not desirable because they consume resources and the utility they provide is unrealistic as they are not part of the corridor. To prevent this situation, we define $\bar{\mathcal{N}}_i$ as the set of patches that share a boundary with patch $i \in P$. For $\mathcal{H} \subset P$, we define $\partial\mathcal{H} = \bigcup_{i\in\mathcal{H}} \bar{\mathcal{N}}_i \setminus \mathcal{H}$ as the set of patches adjacent (by boundary) to at least one patch in $\mathcal{H}$ but not belonging to $\mathcal{H}$, i.e., $\partial\mathcal{H}$ is the *boundary* of $\mathcal{H}$. To ensure that every selected patch is connected to the corridor, we enforce the following *ring* constraints (adapted from Carvajal et al. (2013)),

$$\sum_{j\in\partial\mathcal{H}\setminus\{S\cup T\}} x_j \geq x_k,\ \forall k \in P \setminus \{S\cup T\}, \mathcal{H} \in \mathscr{C}(k), \tag{2}$$

where $\mathscr{C}(k) = \{\mathcal{H} \subset P \setminus \{S\cup T\} : k \in \mathcal{H}\}$ is the set of all patch sets containing non-habitat patch k. Because there is an exponential number of Constraints (2), we impose them in a B&C fashion. At any integer solution encountered during the B&B exploration, we search for disconnected components, i.e., sets of patches not connected to the corridor as illustrated in Figure 4a, by running a set of Breadth-First Search (BFS) procedures. This allows us to detect not only patches that are not connected to the corridor linking s and t, but also the corresponding connected components (i.e., k and $\mathcal{H}$ in Constraint (2)).

4.3. Imposing w-width

Formulation (1) with Constraints (2) produces a corridor that allows a circle of diameter $w \times \alpha$ to move from S to T. However, there may be areas stemming from the corridor that are impassable for such circle, as shown in the red circled area on the left side of the corridor in Figure 4c. This area is connected to the corridor but fails to satisfy the width requirement because there is no flow through the nodes in G^w associated with the circled patches. To enforce the width requirement across the corridor, we impose the constraints

$$\sum_{j:k\in\Omega_j(w)} \sum_{i:(i,j)\in A, k\notin\Omega_i(w)} y_{ij} \geq \frac{1}{|\mathcal{L}|} x_k, \forall k \in P \setminus (S\cup T), \tag{3}$$

that ensure that if patch k is selected, then there must be positive flow into at least one of the backbone nodes whose Ω-set covers k. We use the coefficient $\frac{1}{|\mathcal{L}|}$ because the maximum number of patches allowed to be selected is $|\mathcal{L}|$. Solution $(\mathbf{x}, \mathbf{y})$ is removed after imposing Constraints (3) because the corresponding left-hand side is equal to zero and $x_k = 1$. Because there is a limited number of Constraints (3), we add them all to Formulation (1)-(2) since the beginning of the B&B algorithm. This change results in Model (1′).

4.4. Imposing backbone connectivity

Model (1′) ensures that every selected patch in a feasible solution $(\mathbf{x}, \mathbf{y})$ is connected to the corridor and enforces the width requirement locally. However, the corridor may still have narrow sections that violate the width requirement due to the presence of flow subtours disconnected from s. This is the case of the area to the right side of the corridor (highlighted in a red circle) in Figure 4c, which is connected to the corridor, satisfies Constraints (2) and (3), but does not permit passage of a circle of diameter $w \times \alpha$. To enforce the width w in such cases, we define $G^w(\mathcal{H}) = (N(\mathcal{H}), A(\mathcal{H}))$ as the backbone network induced by the set of patches $\mathcal{H} \subset P \setminus \{S \cup T\}$, such that $\Omega_i(w) \subseteq \mathcal{H}$ for each $i \in N(\mathcal{H})$, and $(i,j) \in A(\mathcal{H})$ when $\Omega_j(w) \subset \mathcal{H}$ and $j \in \mathcal{N}_i$. Note that $G^w(\mathcal{H})$ is *connected* when $\mathcal{H}$ consists of a set of connected patches satisfying the width constraint. For any $\mathcal{H}, \mathcal{H}' \subset P \setminus \{S \cup T\}$ such that $G^w(\mathcal{H})$ and $G^w(\mathcal{H}')$ are connected, $G^w(\mathcal{H} \cup \mathcal{H}')$ is disconnected, and $\mathcal{H}$ is not a valid corridor connecting S and T, the following constraints enforce the width requirement between the two sets of connected patches.

$$\sum_{j \in \partial\mathcal{H} \setminus \mathcal{H}'} x_j \geq x_k, \ \forall k \in \mathcal{H} \tag{4}$$

Similar to Constraints (2), we impose Constraints (4) using a B&C scheme. The sets $\mathcal{H}$ and $\mathcal{H}'$ can be identified by running a BFS algorithm on the backbone network induced by any integer solution. In this case, $\mathcal{H}'$ represents the set of patches inducing a connected backbone network linking S and T (i.e., $\mathcal{H}'$ is a valid corridor), while $\mathcal{H}$ consists of patches such that $G^w(\mathcal{H})$ is disconnected from $G^w(\mathcal{H}')$ (as illustrated in Figure 4c). Because $\partial\mathcal{H} \setminus \mathcal{H}'$ is not necessarily a complete *ring* around $\mathcal{H}$, we refer to Constraints (4) as *semi-rings*.

We define Model (1″) as Model (1) augmented with Constraints (2), (3), and (4). Model (1″) provides a corridor that satisfies all the geometrical demands in w-MUC. An example of a feasible corridor is given in Figure 4d.

Proposition 4.1 establishes the computational complexity of the corridor problem studied in this paper, whose decision version we denote by Dw-MUC. A complete proof is provided in Appendix B.

PROPOSITION 4.1. Dw-MUC is NP-complete.

5. Solution algorithm and acceleration techniques

In this section, we describe a B&C algorithm and a set of acceleration techniques to improve time to solve w-MUC. Section 5.1 presents a warm-up strategy based on a polynomial-time algorithm for solving a minimum-cost corridor design problem with width w, a preprocessing procedure to identify ineligible patches for a given maximum allowable corridor size, and a heuristic algorithm to enhance the utility of a given corridor. Section 5.4 presents two families of valid inequalities to strengthen Model (1″). Section 5.5 describes how to obtain lower and upper bounds for the optimal value of w-MUC, while Section 5.6 describes a complete procedure for the solution of w-MUC.

5.1. Warm-start heuristic

To improve the performance of the B&B search, we develop a warm-up strategy to quickly obtain an initial feasible corridor with a corresponding lower bound on the optimal corridor utility. For any given width w, size $\mathcal{L}$, and budget b, we sample backbone paths connecting s and t in G^w using various criteria and then use the corresponding Ω-sets to construct a candidate corridor.

Algorithm 1 produces a corridor of width w by running a shortest path algorithm on $G^w = (N, A)$ with modified arc costs. The inputs are the target width w and a vector $\mathbf{c} = [c_i]_{i \in P}$ that captures the patch costs (e.g., purchasing or restoration).

Lines 1–3 build G^w and compute the arc cost between adjacent backbone nodes i and j as the total cost of patches in $\Omega_j(w) \setminus \Omega_i(w)$. We assume that all patches in habitats S and T have zero cost (i.e., they are already protected) and $\Omega_i(w) = \varnothing$ for each $i \in S$. Line 4 finds an st-shortest path of backbone nodes, $\mathcal{P}$. Line 5 constructs the corresponding corridor of width w from s to t, $\mathcal{C}^w$, and calculates its total cost, z^w. Algorithm 1 produces a corridor from s to t of width w with a time complexity given by the shortest-path algorithm used in Line 4 (e.g., $\mathcal{O}(|A| + |N| \log |N|)$ (Fredman and Tarjan 1987)).

PROPOSITION 5.1. Algorithm 1 returns a corridor of width w connecting S and T.

Proof. Line 4 produces a sequence of backbone nodes connecting habitat nodes s and t. To prove that the width of the resulting corridor is at least w, we divide the analysis into two parts. First, by the construction of G^w, any outgoing arc from s or incoming arc to t guarantees that the corridor width is at least w at that segment (see Figure 3). This follows from the enforced overlap of the corresponding Ω-sets with S and T. Next, consider any arc (i, j) that belongs to $\mathcal{P}$, with $i, j \notin \{s, t\}$. By construction, this arc represents either a horizontal or vertical movement in the grid. In the horizontal case, $\Omega_j \cap \Omega_i$ forms a $w \times (w-1)$ area, while in the vertical case, it forms a $(w-1) \times w$ area. This guarantees that the Ω-sets of nodes connected by an arc in $\mathcal{P}$ share at least w patches along one of their sides, enforcing the target width of w at every segment. By combining these two cases, we conclude that the resulting corridor has a width of at least w throughout. ■

5.2. Pre-processing procedure

Algorithm 1 can verify whether a corridor of width w and size $\mathcal{L}$ exists between habitats S and T. The following proposition formalizes this result.

PROPOSITION 5.2. Consider a given set of candidate patches P, with habitat patches defined by S and T, size parameter $\mathcal{L}$, and a target width w. Define z^w as the cost of the corridor obtained by invoking *min-cost*$(w, \tilde{\mathbf{1}})$, where $\tilde{\mathbf{1}}$ is a $|P|$-dimensional vector, with its i-th entry set to 1 if $i \in P \setminus (S \cup T)$ and 0 otherwise. Then, the w-MUC problem is infeasible whenever $\mathcal{L} < z^w$.

Algorithm 1 : *min-cost*$(w, \mathbf{c})$

Input: w, $\mathbf{c}$

Output: $\mathcal{C}^w$, z^w

1: Build network $G^w = (N, A)$ (see Section 4.1)
2: **for** $(i, j) \in A$ **do**
3: Compute the cost of arc (i, j) as $\sum_{k \in \Omega_j(w) \setminus \Omega_i(w)} c_k$
4: Find a shortest path $\mathcal{P}$ from s to t with node set $N_{\mathcal{P}}$
5: Compute $\mathcal{C}^w = \bigcup_{k \in N_{\mathcal{P}}} \Omega_k(w)$ and $z^w = \sum_{i \in \mathcal{C}^w \setminus (S \cup T)} c_i$
6: Return $\mathcal{C}^w$, z^w

Note that in landscapes where all patches are eligible, *min-cost*$(w, \tilde{\mathbf{1}})$ returns the minimum number of patches needed to produce a corridor of width w from s to t. However, if not all patches in the landscape are available or the cost vector $\mathbf{c}$ is generically defined, Algorithm 1 provides a cost proxy rather than a guaranteed lower bound on the corridor cost due to the possible overlap of Ω-sets in an optimal path. Appendix C illustrates this situation with an example.

We extend Algorithm 1 into a pre-processing procedure to reduce the number of decision variables by removing patches that are not reachable by a corridor of width w and maximum size $\mathcal{L}$. To this end, Algorithm 2 finds a corridor of width w connecting backbone nodes x and y and assuming that a set of patches $\Lambda \subset P$ is already selected (i.e., $c_i = 0, \forall i \in \Lambda$). Algorithm 2 is similar to Algorithm 1, except for the generic origin and destination (x and y in this case instead of s and t) and the existence of Λ. In Algorithm 2, the computation of the cost of arc (i, j) in Line 3 discounts the cost of patches in Λ.

Algorithm 2 : *min-cost*$(w, \mathbf{c}, x, y, \Lambda)$

Input: w, $\mathbf{c}$, x, y, Λ

Output: $\mathcal{P}_{xy}^{w,\Lambda}$, $N_{xy}^{w,\Lambda}$, $\mathcal{C}_{xy}^{w,\Lambda}$, $z_{xy}^{w,\Lambda}$

1: Build network $G^w = (N, A)$ (see Section 4.1)
2: **for** $(i, j) \in A$ **do**
3: Compute the cost of arc (i, j) as $\sum_{k \in \Omega_j(w) \setminus (\Omega_i(w) \cup \Lambda)} c_k$
4: Find a shortest path $\mathcal{P}_{xy}^{w,\Lambda}$ from x to y with node set $N_{xy}^{w,\Lambda}$
5: Compute $\mathcal{C}_{xy}^{w,\Lambda} = \bigcup_{k \in N_{xy}^{w,\Lambda}} \Omega_k(w)$ and $z_{xy}^{w,\Lambda} = \sum_{i \in \mathcal{C}_{xy}^{w,\Lambda} \setminus (S \cup T \cup \Lambda)} c_i$
6: Return $\mathcal{P}_{xy}^{w,\Lambda}$, $N_{xy}^{w,\Lambda}$, $\mathcal{C}_{xy}^{w,\Lambda}$, $z_{xy}^{w,\Lambda}$

Definition 5.1 determines whether a backbone node can be reached based on the maximum allowable corridor size $\mathcal{L}$.

Definition 5.1 *A backbone node x in $G^w = (N, A)$ is reachable by a corridor of maximum size $\mathcal{L}$ and width w if it satisfies at least one of the following conditions:*

(1) There exists a path $\mathcal{P}_{st}$ from s to t in G^w, with corresponding corridor size $\mathcal{L}$ and width w, that passes through node x, i.e., $x \in N_{st}$, where N_{st} denotes the set of nodes in $\mathcal{P}_{st}$.

(2) There exist a path $\mathcal{P}_{st}$ from s to t in G^w and a path $\tilde{\mathcal{P}}_x$ connecting node $x \notin N_{st}$ to a node $y \in N_{st} \setminus \{s,t\}$, resulting in a corridor of size $\mathcal{L}$ and width w.

Based on Definition 5.1, Algorithm 3 outlines a shortest-path preprocessing (SPP) procedure applied to G^w for given values of w and $\mathcal{L}$. The output, $N^{w,\mathcal{L}}$, represents the set of backbone nodes that are unreachable and will never be part of an optimal solution for the specified $\mathcal{L}$ and w. The algorithm checks whether each node satisfies the conditions in Definition 5.1 by solving $\mathcal{O}(|N|^2)$ shortest-path problems on modified networks. Nodes that meet the definition are labeled as reachable and removed from $N^{w,\mathcal{L}}$.

Line 1 initializes $N^{w,\mathcal{L}}$ to $N \setminus \{s,t\}$. Lines 2 examines all non-habitat nodes in $N \setminus \{s,t\}$ while searching for a path from s to t that passes through y and contains a segment from x to y. Line 3 verifies whether $N^{w,\mathcal{L}}$ is empty. If it is, all nodes are reachable, and the algorithm terminates returning an empty set. Else, Line 4 calls *min-cost*$(w, \tilde{\mathbf{1}}, s, y, \varnothing)$ to compute a shortest-path from s to y with corresponding node set $N_{sy}^{w,\varnothing}$, corridor $\mathcal{C}_{sy}^{w,\varnothing}$, and cost $z_{sy}^{w,\varnothing}$. Using a network with the cost of patches in $\mathcal{C}_{sy}^{w,\varnothing}$ equal to zero, Line 5 invokes *min-cost*$(w, \tilde{\mathbf{1}}, y, t, \Lambda_{sy})$ to compute a shortest path from node y to t, with corresponding node set $N_{yt}^{w,\Lambda_{sy}}$, corridor $\mathcal{C}_{yt}^{w,\Lambda_{sy}}$, and cost $z_{yt}^{w,\Lambda_{sy}}$, where $\Lambda_{sy} = \mathcal{C}_{sy}^{w,\varnothing}$. Combining the corridors $\mathcal{C}_{sy}^{w,\varnothing}$ and $\mathcal{C}_{yt}^{w,\Lambda_{sy}}$ results in corridor Λ_{st}^{y} that connects s to t with width w and passes through node y. The for-loop in Line 6 examines each node in $N^{w,\mathcal{L}}$ to find a shortest-path from x to y. After updating the costs for patches in Λ_{st}^{y} to zero, Line 7 calls *min-cost*$(w, \tilde{\mathbf{1}}, x, y, \Lambda_{st}^{y})$ to obtain a shortest path from x to y with corresponding node set $N_{xy}^{w,\Lambda_{st}^{y}}$, corridor $\mathcal{C}_{xy}^{w,\Lambda_{st}^{y}}$, and cost $z_{xy}^{w,\Lambda_{st}^{y}}$. If the corridor from s to t that passes through y and contains a segment from x to y does not exceed the maximum size $\mathcal{L}$, then Lines 8–9 remove *all* nodes in $N_{sy}^{w,\varnothing} \cup N_{yt}^{w,\Lambda_{sy}} \cup N_{xy}^{w,\Lambda_{st}^{y}}$ from $N^{w,\mathcal{L}}$, as they are considered reachable. Line 10 returns the set of unreachable nodes, $N^{w,\mathcal{L}}$.

Algorithm 3 terminates because, in the worst-case, $\mathcal{O}(|N|^2)$ shortest-path problems are solved to find the set of unreachable nodes. Algorithm 3 can be used recursively to determine reachable patches in a given landscape for multiple values of w. This is important as decisions to conserve corridors require multiple designs before reaching a final decision. We take advantage of the nested structure of real decisions by implementing Algorithm 3 as follows. This is illustrated in more detail in Section 6.

(1) For each width w and set of patches P, we produce corridor designs with different shapes by sequentially using the size parameters $\mathcal{L}_1 < \mathcal{L}_2 < \cdots < \mathcal{L}_n$

Algorithm 3 : $SPP(w, \mathcal{L}, G^w)$

Input: w, $\mathcal{L}$

Output: $N^{w,\mathcal{L}}$

1: Initialize $N^{w,\mathcal{L}} = N \setminus \{s,t\}$
2: **for** $y \in N \setminus \{s,t\}$ **do**
3: **if** $N^{w,\mathcal{L}} \neq \varnothing$ **then**
4: Compute $N_{sy}^{w,\varnothing}$, $\mathcal{C}_{sy}^{w,\varnothing}$, and $z_{sy}^{w,\varnothing}$ by calling *min-cost*$(w, \tilde{\mathbf{1}}, s, y, \varnothing)$. Set $\Lambda_{sy} = \mathcal{C}_{sy}^{w,\varnothing}$
5: Compute $N_{yt}^{w,\Lambda_{sy}}$, $\mathcal{C}_{yt}^{w,\Lambda_{sy}}$, and $z_{yt}^{w,\Lambda_{sy}}$ by calling *min-cost*$(w, \tilde{\mathbf{1}}, y, t, \Lambda_{sy})$. Set $\Lambda_{st}^{y} = \mathcal{C}_{sy}^{w,\varnothing} \cup \mathcal{C}_{yt}^{w,\Lambda_{sy}}$
6: **for** $x \in N^{w,\mathcal{L}}$ **do**
7: Compute $N_{xy}^{w,\Lambda_{st}^{y}}$, $\mathcal{C}_{xy}^{w,\Lambda_{st}^{y}}$, and $z_{xy}^{w,\Lambda_{st}^{y}}$ by calling *min-cost*$(w, \tilde{\mathbf{1}}, x, y, \Lambda_{st}^{y})$
8: **if** $z_{sy}^{w,\varnothing} + z_{yt}^{w,\Lambda_{sy}} + z_{xe}^{w,\Lambda_{st}^{y}} \leq \mathcal{L}$ **then**
9: Update $N^{w,\mathcal{L}} \leftarrow N^{w,\mathcal{L}} \setminus (N_{sy}^{w,\varnothing} \cup N_{yt}^{w,\Lambda_{sy}} \cup N_{xy}^{w,\Lambda_{st}^{y}})$
10: Return $N^{w,\mathcal{L}}$

(2) For each size $\mathcal{L}_i$, $i \in \{1, \ldots, n-1\}$, we implement Algorithm 3 to obtain the unreachable set $N^{w,\mathcal{L}_i}$ and use it to initialize $N^{w,\mathcal{L}_{i+1}}$ instead of $N \setminus \{s,t\}$ in Line 1. Proposition 5.3 justifies this strategy.

PROPOSITION 5.3. Consider a corridor problem with target width w and maximum size $\mathcal{L}$. The following properties hold as a result of Definition 5.1.

- If node x is reachable, then it remains reachable for a corridor of the same width w and any size $\mathcal{L}' > \mathcal{L}$.
- Node x is unreachable if there exists a node x' that is unreachable by a corridor of the same size $\mathcal{L}$ and width $w' < w$ such that $\Omega_{x'}(w') \subset \Omega_x(w)$.

Proof. To prove the first property, suppose for a contradiction that node x is reachable by a corridor of width w and size $\mathcal{L}$, but unreachable by a corridor of the same width and size $\mathcal{L}' > \mathcal{L}$. However, the corridor of size $\mathcal{L}$ is feasible for the w-MUC problem with target size $\mathcal{L}'$ according to Constraint (1d), thus node x is reachable, which is a contradiction. The proof of the second property follows similar arguments. ■

Step (2) can be further accelerated by replacing $N^{w,\mathcal{L}_i}$ in Lines 1 and 10 of Algorithm 3 with $N^{w,\mathcal{L}_{i-1}} \setminus \bigcup_{x' \in N^{w',\mathcal{L}_i}} \{x \in N : \Omega_{x'}(w') \subset \Omega_x(w)\}$ and $N^{w,\mathcal{L}_i} \cup \bigcup_{x' \in N^{w',\mathcal{L}_i}} \{x \in N : \Omega_{x'}(w') \subset \Omega_x(w)\}$, respectively, for any w' such that $w' < w$. To remove unreachable patches in a corridor with target width w and size $\mathcal{L}$, we Compute $N^{w,\mathcal{L}}$ by calling Algorithm 3 and update $P \leftarrow P \setminus \cup_{k \in N^{w,\mathcal{L}}} \Omega_k(w)$ as input to Model 1. Algorithm 3 is correct, as it exhaustively explores all possible ways to connect node x to a valid corridor from s to t while satisfying the conditions in Definition 5.1.

5.3. Local-search heuristic

We also implement a corridor improvement heuristic that leverages the structure of the objective function (1a) and the nonnegativity of the f-parameters, as adding additional patches to a feasible

corridor $\mathcal{C}$ increases its total utility. Based on this observation, Algorithm 4 seeks to enhance the total utility of a given feasible corridor $\mathcal{C}$ by adding more patches while respecting the maximum allowable size and budget constraints. Algorithm 4 examines the backbone nodes in $\bar{N} = \{i \in N : \Omega_i(w) \not\subset \mathcal{C}\}$ by layers, based on their distance to $\mathcal{C}$, and sequentially attempts to add them to the corridor in nondecreasing order of their utility-cost ratio. The inputs of Algorithm 4 include a feasible corridor $\mathcal{C}$, a set of already selected patches $\bar{P}$, the maximum size $\mathcal{L}$, the available budget b, and the utility and cost vectors $\mathbf{f}$ and $\mathbf{c}$, respectively. The algorithm returns an improved corridor, if one is found, along with its corresponding utility value.

Line 1 creates a copy of the corridor and initializes $\tilde{b}$ and $\tilde{\mathcal{L}}$ as the leftover budget and the remaining number of patches that can be selected, respectively. Patches that are already protected or purchased, if any, do not count toward the budget b or the size constraint $\mathcal{L}$. Formally, let $\bar{P}$ denote the set of patches that are already protected or purchased, ensuring that $c_k = 0, \forall k \in \bar{P}$. If $\tilde{b} \leq 0$ or $\tilde{\mathcal{L}} \leq 0$, the algorithm terminates, as no additional patches can be added to the corridor. Line 2 constructs a network, G_e^w, where the node set consists of $\bar{N}$ along with an additional node, e, representing $\mathcal{C}$. The network G_e^w includes an arc (i, j) for each pair $i, j \in \bar{N}$ such that $(i, j) \in A$. Additionally, it contains an arc (e, j) for each $j \in \bar{N}$ such that $j \in \mathcal{N}_i$ with $i \in N \setminus \bar{N}$. The cost of every arc in G_e^w is set to 1. Lines 3–5 calculate a utility ratio $\frac{\sum_{k \in \Omega_i(w)} f_k}{\max\{\sum_{k \in \Omega_i(w)} c_k, 1\}}$ for each node $i \in \bar{N}$ and the shortest distance from node e to each node $i \in \bar{N}$, denoted by d_{ei}, which is equivalent to the shortest distance from node i to corridor $\mathcal{C}$. Note that we use $\max\{\cdot, 1\}$ to avoid division by 0 in case all patches in $\Omega_i(w)$ are already protected (i.e., their cost is equal to 0). Line 6 stores the maximum distance between nodes in $\bar{N}$ to e in parameter ℓ, which represents the number of layers around corridor $\mathcal{C}$ to be processed. Line 7 builds the first layer such that L_1 contains all the nodes located at a distance of 1 from e in nonincreasing order of the utility-cost ratios calculated in Line 4. Line 8 initializes k and extracts the first node from L_1. The While-loop in Lines 9–17 sequentially adds nodes to the corridor while satisfying the budget and size constraints. After adding the additional patches from the currently processed node to the corridor in Line 10, Line 11 updates the leftover budget and size available and removes the current node from its layer. Once all nodes in a layer have been processed and there are still layers to explore, Lines 12–17 update the current layer index, construct the next layer, and extract the next node for evaluation. In Line 18, Algorithm 4 returns the obtained corridor, $\mathcal{C}'$, along with its corresponding utility, $U_{\mathcal{C}'}$.

We generate multiple initial feasible corridors by combining additional strategies with Algorithm 4. First, we set the cost of every arc in G^w to 1 and obtain K paths from s to t using a K-shortest path algorithm (Yen 1971). For each of these paths, we construct a corridor using the Ω-sets of the nodes along the path. Additionally, we run *min-cost*$(w, \mathbf{c})$, using inverse functions of patch utility as the cost, to generate corridors for potential improvement. Each obtained corridor is then enhanced

Algorithm 4 : *stripe*($\mathcal{C}$, $\mathcal{L}$, $\bar{P}$, b, $\mathbf{f}$, $\mathbf{c}$)

Input: $\mathcal{C}$, $\mathcal{L}$, $\bar{P}$, b, $\mathbf{f}$, $\mathbf{c}$

Output: $\mathcal{C}'$, $U_{\mathcal{C}'}$

1: Initialize $\mathcal{C}' = \mathcal{C}$, $\tilde{b} = b - \sum_{j \in \mathcal{C}'} c_j$ and $\tilde{\mathcal{L}} = \mathcal{L} - \left|\mathcal{C}' \setminus \bar{P}\right|$. If $\tilde{b} \leq 0$ or $\tilde{\mathcal{L}} \leq 0$ go to Line 18
2: Build auxiliary network G_e^w
3: **for** $i \in \bar{N}$ **do**
4: Calculate ratio $r_i = \frac{\sum_{k \in \Omega_i(w)} f_k}{\max\{\sum_{k \in \Omega_i(w)} c_k, 1\}}$
5: Calculate d_{ei} on G_e^w
6: Compute $\ell = \max_{i \in \bar{N}} d_{ei}$
7: Build set $L_1 = \{i \in \bar{N} : d_{ei} = 1\}$ and sort elements of L_1 in nonincreasing order of r-ratios
8: Set $k = 1$ and $i \leftarrow$ first node of L_k
9: **while** $\tilde{b} - \sum_{j \in \Omega_i(w) \setminus \mathcal{C}'} c_j > 0$ and $\tilde{\mathcal{L}} - |\Omega_i(w) \setminus \mathcal{C}'| > 0$ **do**
10: Update $\mathcal{C}' \leftarrow \mathcal{C}' \cup \Omega_i(w)$
11: Update $\tilde{b} \leftarrow \tilde{b} - \sum_{j \in \Omega_i(w) \setminus \mathcal{C}'} c_j$, $\tilde{\mathcal{L}} \leftarrow \tilde{\mathcal{L}} - |\Omega_i(w) \setminus \mathcal{C}'|$, and $L_k \leftarrow L_k \setminus \{i\}$
12: **if** $|L_k| = 0$ and $k < \ell$ **then**
13: Set $k \leftarrow k + 1$
14: Build set $L_k = \{i \in \bar{N} : d_{ei} = k\}$ and sort elements of L_k in nonincreasing order of r-ratios
15: Set $i \leftarrow$ first node of L_{k+1}
16: **else**
17: Set $i \leftarrow$ next node in L_k
18: Return $\mathcal{C}'$, $U_{\mathcal{C}'} = \sum_{j \in \mathcal{C}'} f_j$

using *stripe*($\mathcal{C}, \mathcal{L}, \bar{P}, b, \mathbf{f}, \mathbf{c}$) (Algorithm 4), resulting in a pool of candidate corridors. We select the corridor with the highest utility, denoted as $\hat{\mathcal{C}}$, with corresponding utility $U_{\hat{\mathcal{C}}}$. To warm-start the B&B exploration, we impose the constraint $z(w) \geq U_{\hat{\mathcal{C}}}$ and initialize the incumbent solution by setting $x_i = 1$ for each $i \in \hat{\mathcal{C}}$, and $x_i = 0$ otherwise.

5.4. Strengthening the formulation

We strengthen Model (1″) using multiple valid inequalities, which we impose either directly to the root node or during the B&B exploration. Whenever we obtain an integer solution during the B&B exploration that induces a disconnected corridor, we impose Constraints (2) and (4) to ensure the patch and backbone connectivity requirements. In addition, we utilize Proposition 5.4 to eliminate solutions that violate the maximum allowable size when disconnected patches are re-connected to a corridor. In this case, $\mathcal{C}$ is the set of patches in a corridor from s to t satisfying the width requirement. The parameter $\tilde{d}_k^{\mathcal{C}}$ represents the minimum number of patches required to connect patch $k \in P \setminus (\mathcal{C} \cup S \cup T)$ to the corridor $\mathcal{C}$. We use G_e^w to obtain $\tilde{d}_k^{\mathcal{C}}$. To calculate the $\tilde{d}$-parameters, we set the cost of arc (i, j) to $|\Omega_j(w) \setminus \Omega_i(w)|$ when $i, j \in \bar{N}$ as in Algorithm 1, whereas the cost of arc (e, j) is given by $|\Omega_j(w) \setminus \mathcal{C}|$, for each $j \in \mathcal{N}_i$ such that $i \in N \setminus \bar{N}$. Using this construction,

we calculate $\tilde{d}_k^{\mathcal{C}}$ by finding the shortest path from e to the nodes in $\bar{N}$ whose Ω-sets cover patch k. The corresponding cost is given by $\tilde{d}_k^{\mathcal{C}} = \min_{\{i \in \bar{N}: k \in \Omega_i(w)\}} d_{ei}$.

PROPOSITION 5.4. Consider any feasible corridor $\mathcal{C}$ and patch $k \notin \mathcal{C}$ such that $\tilde{d}_k^{\mathcal{C}} > \mathcal{L} - |\mathcal{C}|$, then

$$\sum_{j \in \mathcal{C}} x_j + x_k \leq |\mathcal{C}| \tag{5}$$

in any optimal solution.

Proof. Suppose that $x_j = 1$ for all $j \in \mathcal{C}$, and that $x_k = 1$. The minimum number of additional patches required to connect k to $\mathcal{C}$ is $\tilde{d}_k^{\mathcal{C}}$, resulting in a corridor of size $|\mathcal{C}| + \tilde{d}_k^{\mathcal{C}} > \mathcal{L}$, which is infeasible. Consequently, patch k and the patches in $\mathcal{C}$ cannot be selected simultaneously. ■

Constraints (5) eliminate fractional solutions, including those with $x_i = 1$ for all $i \in \mathcal{C}$ and $x_k > 0$ whenever $\tilde{d}_k^{\mathcal{C}} > \mathcal{L} - |\mathcal{C}|$.

When the landscape is an $n \times m$ grid, we strengthen Model (1″) by introducing Constraints (6) and (7). These constraints also work for the hexagon discretization (see Figure 9 in Appendix A) by turning the coordinate system 60 degrees clockwise. In these constraints, R_i and C_j denote the set of candidate patches in the i-th row and j-th column of the grid, respectively, for values of i and j that will surely contain corridor patches. The use of Constraints (6) and (7) is formalized in Proposition 5.5.

PROPOSITION 5.5. Consider any feasible corridor $\mathcal{C}$ in a $n \times m$ grid landscape. Define coordinates $(\underline{i}, \underline{j})$ and $(\bar{i}, \bar{j})$ as the top-left and bottom-right corner patches in the smallest box containing both S and T. Then any optimal solution satisfies

$$\sum_{k \in R_i} x_k \geq w, \quad \forall i \in \{\underline{i}, \ldots, \bar{i}\} : R_i \cap (S \cup T) = \varnothing, \tag{6}$$

$$\sum_{k \in C_j} x_k \geq w, \quad \forall j \in \{\underline{j}, \ldots, \bar{j}\} : C_j \cap (S \cup T) = \varnothing. \tag{7}$$

Proof. By construction, if there exists an index $i \in \{\underline{i}, \ldots, \bar{i}\}$ such that $R_i \cap (S \cup T) = \varnothing$, then the patches in R_i form an st-cut. As a result, if $\sum_{k \in R_i} x_k = 0$, then habitats S and T become disconnected. Due to the geometry of the landscape, if $0 < \sum_{k \in R_i} x_k < w$, the corridor connecting S and T will fail to meet the width requirement. A similar analysis applies to patches in C_j whenever there is an index $j \in \{\underline{j}, \ldots, \bar{j}\}$ such that $C_j \cap (S \cup T) = \varnothing$. To guarantee connectivity and required width, any feasible corridor must satisfy the conditions in (6) and (7). ■

5.5. Bounds on the objective function value

5.5.1. Lower bound. We establish lower bounds for w-MUC by enhancing integer solutions during the B&B exploration whenever possible. When an integer solution with disconnected patches

from the corridor is obtained, we trigger Algorithm 4 with the call *stripe*$(\tilde{\mathcal{C}}, \mathcal{L}, \bar{P}, b, \mathbf{f}, \mathbf{c})$, where $\tilde{\mathcal{C}}$ is the corridor induced by the integer solution that connects s and t and satisfies the minimum width. If the utility of the corridor returned by *stripe*$(\cdot)$, $\tilde{\mathcal{C}}'$, is larger than the current incumbent value, we tighten the formulation by enforcing an objective function value of at least $U_{\tilde{\mathcal{C}}'}$ and update the incumbent solution to $\tilde{\mathcal{C}}'$.

We emphasize that in order to produce a lower bound on the optimal solution of an instance with maximum size $\mathcal{L}$ and target width w, it is also possible to solve more restrictive problems with maximum size $\mathcal{L}' < \mathcal{L}$ or target width $w' > w$.

5.5.2. Upper bound. Our preliminary computational results indicated that the linear relaxation objective value in w-MUC is very weak, motivating the search for a tighter upper bound. To this end, we introduce Proposition 5.6.

PROPOSITION 5.6. If $\mathcal{C}_{w^*}$ is an optimal corridor with utility $U_{\mathcal{C}_{w^*}}$ to an instance with budget b, maximum size $\mathcal{L}$, and target width w^*, then $U_{\mathcal{C}_{w^*}} \geq U_{\mathcal{C}_w}$ for any $w \geq w^*$, where $\mathcal{C}_w$ is any feasible corridor of width w.

Proof. Consider any corridor $\mathcal{C}_w$ with width $w \geq w^*$ that is feasible for budget b and maximum size $\mathcal{L}$. This means that $\mathcal{C}_w$ is also feasible to the problem with target width w^*, thus $U_{\mathcal{C}_w} \leq U_{\mathcal{C}_{w^*}}$. ■

According to Proposition 5.6, solving problems with smaller widths than required produces upper bounds for problems with larger width w, under the same maximum size $\mathcal{L}$ and budget b. We use this result to produce upper bounds.

5.6. Solution Strategy

Algorithm 5 describes a B&C strategy to find a corridor of maximum utility and target width w. The algorithm is rooted in Model (1″) from Section 4, and uses the warm-start heuristic and *SPP* procedures described in Sections 5.1 and 5.2, and valid inequalities (5) –(7), along with iterative calls to *stripe*$(\cdot)$ as described in Section 5.5.1. Algorithm 5 takes as input the target width w, budget b, maximum size $\mathcal{L}$, tolerance parameter ϵ, time limit parameter T, and the parameter K (described in Section 5.3), as well as the utility and cost vectors, $\mathbf{f}$ and $\mathbf{c}$, respectively. Its output is a corridor, $\mathcal{C}^*$, along with its corresponding optimality gap, Δz.

The algorithm maintains upper and lower bounds, which are updated throughout the iterations. Line 1 initializes the lower and upper bounds as well as the output corridor and optimality gap. If available, these bounds can be set to the values described in Section 5.5. Line 2 determines the feasibility of the problem instance by calculating a corridor of minimum size. Line 3 stops the algorithm if the instance is infeasible. When the instance is feasible with respect to the corridor size, Line 4 obtains an initial corridor and its corresponding utility by calling the warm-start heuristic

from Section 5.1. Line 5 updates the incumbent using corridor $\mathcal{C}_0$. Line 6 triggers the solution of Model (1″), augmented with Inequalities (6) and (7) if in a grid landscape, initialized with the warm-start solution from Line 4, and enhanced with the pre-processing procedure described in Section 5.2. The B&B is paused after finding the first integer solution $(\mathbf{x}, \mathbf{y})$ with corresponding corridor $\tilde{\mathcal{C}}$. Line 7 verifies the tolerance and time limit conditions. The while-loop in Lines 7–15 updates the incumbent solution and gap by imposing violated constraints and iterative calls to *stripe*$(\cdot)$ from Section 5.1. The gap is updated using the incumbent solution and the LP-relaxation upper bound from the B&B.

Lines 8–9 verify whether integer solution $(\mathbf{x}, \mathbf{y})$ satisfies Constraints (2), (4), and (5). Sections 4.2, 4.4, and 5.4 provide details on the constraint separation strategy. If $(\mathbf{x}, \mathbf{y})$ is feasible and yields a better lower bound, then Line 9 updates the incumbent solution, lower bound value, and relative gap. The corridor $\tilde{\mathcal{C}}$, which connects S and T, is found by tracking the continuous variables y in the solution $(\mathbf{x}, \mathbf{y})$. Line 11 sends $\tilde{\mathcal{C}}$ to *stripe*$(\cdot)$ for possible improvement and returns corridor $\tilde{\mathcal{C}}'$. If $\tilde{\mathcal{C}}'$ yields a better lower bound, then Line 13 updates the incumbent, lower bound value, and relative gap. Line 14 adds the violated constraints to be enforced in the remaining active nodes of the B&B. Line 15 resumes the B&B exploration with the added inequalities. The algorithm re-visits Line 7 when the next integer solution is found. Line 16 returns corridor $\mathcal{C}^*$ and its optimality gap Δz.

Algorithm 5 : B&C-F(w, b, $\mathcal{L}$, ϵ, T, K, $\mathbf{f}$, $\mathbf{c}$)

Input: w, b, $\mathcal{L}$, ϵ, T, $\mathbf{f}$, $\mathbf{c}$

Output: $\mathcal{C}^*$

1: Initialize $UB = \infty$, $LB = 0$, $\mathcal{C}^* = \varnothing$, and $\triangle z = \infty$
2: Obtain $\mathcal{C}(\mathbf{1}) \leftarrow$ *min-cost*$(w, \mathbf{1})$ (See Section 5.1)
3: **if** $|\mathcal{C}(\mathbf{1}) \setminus (S \cup T)| \leq \mathcal{L}$ **then**
4: Obtain initial corridor $\mathcal{C}_0$ with utility $U_{\mathcal{C}_0}$ using warm-start heuristic (See Section 5.1)
5: Set $LB = U_{\mathcal{C}_0}$
6: Solve Model (1″) to find integer solution $(\mathbf{x}, \mathbf{y})$ and corridor $\tilde{\mathcal{C}}$
7: **while** $\triangle z > \epsilon$ and time $< T$ **do**
8: **if** $(\mathbf{x}, \mathbf{y})$ satisfies (2), (4), and (5), and $U_{\tilde{\mathcal{C}}} > LB$ **then** (See Sections 4.2, 4.4, and 5.4)
9: Update $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}$, $LB \leftarrow U_{\tilde{\mathcal{C}}}$, and $\triangle z$
10: **else**
11: Call *stripe*$(\tilde{\mathcal{C}}, \mathcal{L}, \bar{P}, b, \mathbf{f}, \mathbf{c})$ to obtain $\tilde{\mathcal{C}}'$ (See Section 5.5.1)
12: **if** $U_{\tilde{\mathcal{C}}'} > LB$ **then**
13: Update $\mathcal{C}^* \leftarrow \tilde{\mathcal{C}}'$, $LB \leftarrow U_{\tilde{\mathcal{C}}'}$, and $\triangle z$
14: Add violated inequalities
15: Continue B&B exploration. Go to Line 7 after finding next integer solution $(\mathbf{x}, \mathbf{y})$ and corridor $\tilde{\mathcal{C}}$
16: Return $\mathcal{C}^*$, Δz

6. Computational experiments

This section describes the instances we use for testing the performance of our solution approaches. We use the terms B&C-B, B&C-W, and B&C-F to represent the B&C baseline solution procedure for Model (1″) (B&C-B), the B&C-B procedure enhanced with a warm-start (B&C-W), and the B&C-W procedure with the full set of enhancements (B&C-F), i.e., the bound-improving heuristic from Section 5.5.1 as well as the valid inequalities. In addition, we separately report the performance of B&C-F when the shortest-path pre-processing procedure (i.e., SPP) is used in conjunction with all other enhancements. Unless explicitly noted, the reported performance for B&C-F excludes SPP. Our algorithms are coded in Python and implemented on a Windows 10 Pro 64-bit desktop with an Intel Core i7-7700U CPU (3.60GHz) and 32 GB of RAM. All mathematical programs are solved using CPLEX 12.7.15 with a time limit of 7200 seconds and relative optimality gap tolerance of $\epsilon = 0.001$. For each run, we report the solution time t (in seconds) and the relative gap $\triangle z$ (in %). Section 6.1 describes a case study on corridor design for the endangered Florida panther. Section 6.2 describes a set of randomly generated instances and reports the results for landscapes of different sizes to demonstrate the performance of the proposed solution approaches. Section 6.3 and Appendix I present a sensitivity analysis of utility gains as the budget and allowed corridor size increase.

6.1. Corridor design for the *Florida Panther*

We use the w-MUC model to design a corridor for the protection of the endangered *Florida Panther*. Since the 1970s, conservation efforts have aimed to protect the Florida panther through land conservation and highway underpasses, yet habitat fragmentation, loss, and degradation remain major threats. The landscape in Figure 5 (rotated 90° clockwise) includes two key habitats: S (Florida Panther Wildlife Refuge) and T (northern Big Cypress Preserve) (Marx et al. 2020). Habitat S is a stronghold for the species, with conservation efforts focused on establishing corridors for range expansion and population growth (Kautz et al. 2006). In this case study, site selection for protection is not limited to land acquisition but can also involve alternative conservation mechanisms, such as land easement contracts with landowners. Our model aims to strategically connect these habitats, expanding the Florida panther's range and reducing habitat fragmentation.

The landscape consists of a 57×31 grid, resulting in $|P| = 1767$ patches, which we refer to as the *panther* landscape. Following the approach in Marx et al. (2020) and Fletcher Jr. and Fortin (2018), we assess the relative variation in patch use across different land-cover types in the region. Using this method, we compute the visitation probability for each patch, assuming an individual starts from any patch in S, resulting in a spatially correlated utility surface. This probability, ranging from 0.36 to 0.82, serves as a proxy for patch utility (i.e., $\mathbf{f}$-vector). We define the total corridor utility as the sum of the visiting probabilities of the selected patches.

Due to the challenges in obtaining real land and restoration cost data, we generate the cost vector $\mathbf{c}$ by randomly sampling from a uniform distribution $U(100, 100000)$. To induce a real estate cost structure, we select a patch with the highest cost and assign costs to other patches such that their values decrease with distance from this central patch. A formal procedure for this approach, along with other cost structures, is detailed in Section 6.2.

We generate corridors with widths $w = 2$, 3, and 4, setting b to 10% of the total cost of all non-habitat patches in all instances. For reference, the total costs of corridors obtained using *min-cost*$(w, \mathbf{c})$ are 3.04%, 5.08%, and 7.23% of the total landscape cost for $w = 2$, 3, and 4, respectively. To control corridor geometry, we also adjust the size, starting from the minimum number of patches required for feasibility: $\mathcal{L} = 59$, 92, and 124 for $w = 2, 3$, and 4, respectively. These values are determined using *min-cost*$(w, \mathbf{1})$.

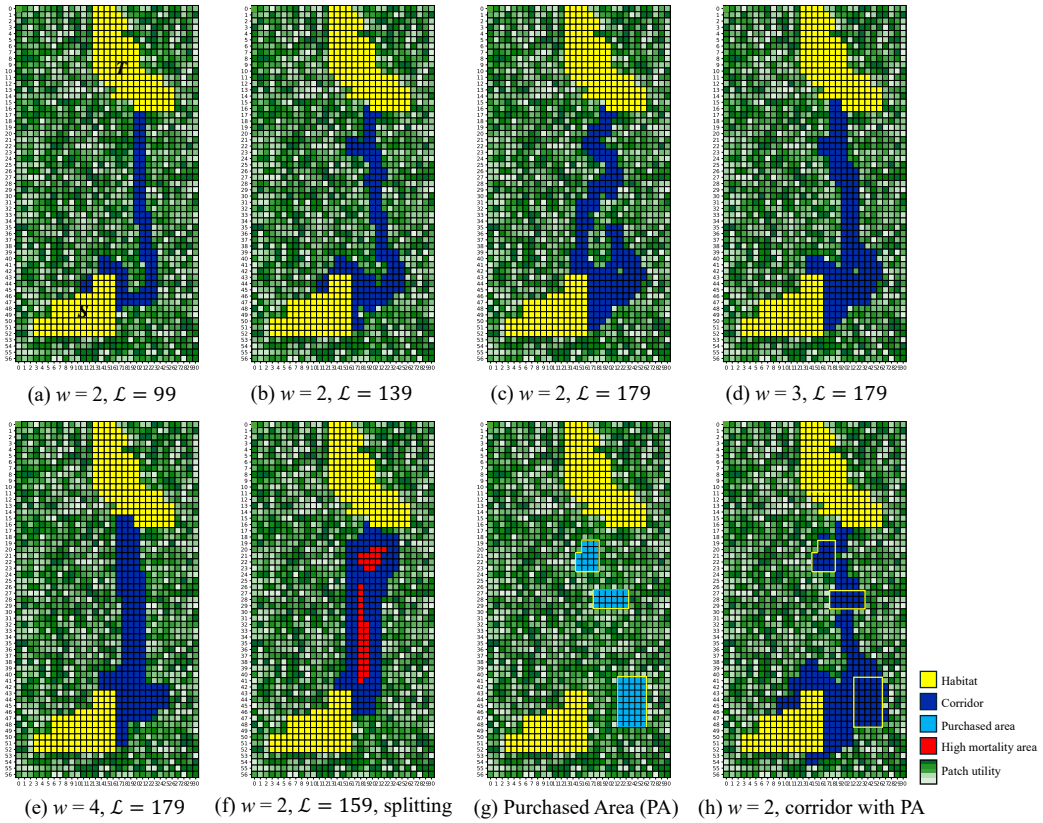


Figure 5 **Florida panther corridor designs**

Figure 5 illustrates optimal corridors with multiple spatial features that can be produced with our model through different parameter configurations. Habitat patches are depicted in yellow,

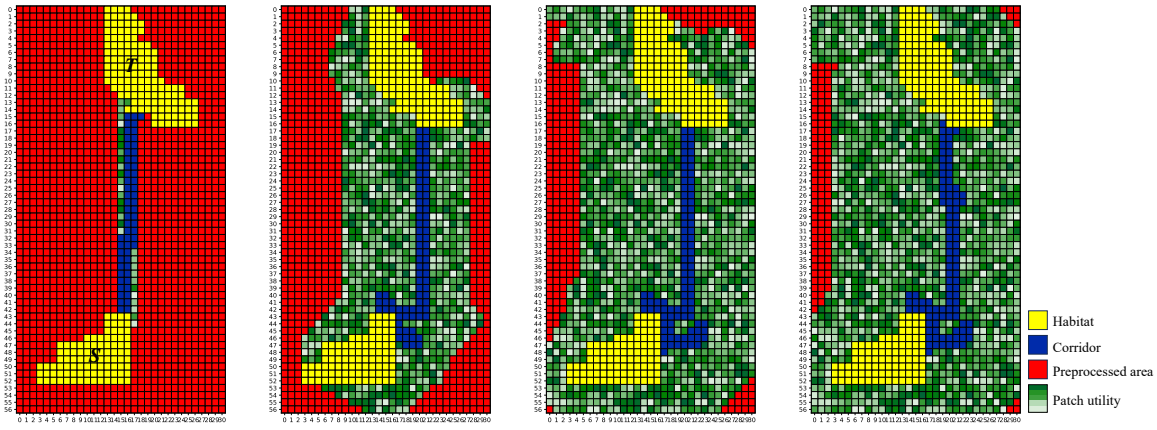
and darker shades in non-habitat patches indicate higher patch utility. The optimal corridors are highlighted in blue. Figures 5a-5c show the corridor design after increasing $\mathcal{L}$ for a fixed w. Figures 5c-5e, show the effect of increasing w for a given value of $\mathcal{L}$, which results in thicker corridors. Figure 5e shows the resulting corridor when the width is increased from $w = 3$ to $w = 4$ with $\mathcal{L} = 179$. This change removes the unselected patch enclosed by the corridor (i.e., the *hole*) in Figure 5d. If undesirable, this configuration can be eliminated by adding a cut during the B&B exploration. Figure 5f illustrates a splitting corridor, a key feature emerging from Model (1″). To induce this corridor, we reduce the utility of the red patches (e.g., areas with high human presence) while increasing the utilities of nearby patches. The resulting corridor splits and merges to bypass low-quality patches while ensuring that each branch meets the width requirement. This capability is unique to our model and is not achievable with existing methods in the literature, which focus on single-branch corridors. Appendix D provides an illustration of the backbone nodes and the associated flow that induce the splitting corridor shown in Figure 5f. Figures 5g and 5h illustrate scenarios where certain areas are already conserved (e.g., previously acquired or protected), and the objective is to find an optimal corridor that incorporates these areas. To enforce this, we introduce the constraints $x_i = 1, \forall i \in \bar{P}$, where $\bar{P}$ represents the pre-selected patches. These patches are shown in light blue in Figure 5g and enclosed within yellow rectangles in Figure 5h.

We pre-process the landscape using SPP to determine the reachability of patches under a given size constraint. As $\mathcal{L}$ increases, the set of reachable patches expands. In our tests, we run SPP starting with the smallest feasible $\mathcal{L}$ and incrementally increase it until all patches are reachable. Figure 6 illustrates optimal corridors (blue), habitats (yellow), and the pre-processed area (red) for different values of w and $\mathcal{L}$ in the *panther* landscape. As $\mathcal{L}$ increases, the reduction in the set of patches becomes less significant.

Corridor protrusions, such as the one shown on the right side of the corridor in Figure 5c, may arise when the optimization model attempts to capture utility from high-value patches. Regulating the corridor size (via $\mathcal{L}$), width (via w), or the available budget (via b) can help mitigate these effects. We recommend using these parameters to explore alternative design trade-offs, as illustrated by the corridor progression in Figures 5c, 5b, and 5a with respect to $\mathcal{L}$, and in Figures 5c, 5d, and 5e with respect to w. To further control corridor shape, Model (1″) could be extended to impose a maximum width constraint or to limit the number of arcs used in G^w. However, such extensions would likely require additional binary variables, further increasing the model's complexity and potentially degrading computational performance.

6.2. Corridor design for randomly generated instances

In addition to the panther landscape, we generate a set of randomly constructed landscapes of varying sizes. We create grid landscapes of dimensions 16×32, 21×42, and 40×80, referred to



(a) $w = 2$, $\mathcal{L} = 59$ (b) $w = 2$, $\mathcal{L} = 79$ (c) $w = 2$, $\mathcal{L} = 92$ (d) $w = 2$, $\mathcal{L} = 99$

Figure 6 Patch reduction through pre-processing using SPP

as *small*, *medium*, and *large*, respectively. In each landscape, the top-left 5×5 region is designated as habitat S, while the bottom-right 5×10 region serves as habitat T. Patch utilities are assigned by sampling the uniform distribution $U(1000, 20000)$. These values are sampled independently to capture the potential for sharp spatial heterogeneity in habitat characteristics and to preserve generality for future extensions, such as multi-species indices, where composite utility measures may exhibit limited spatial smoothness.

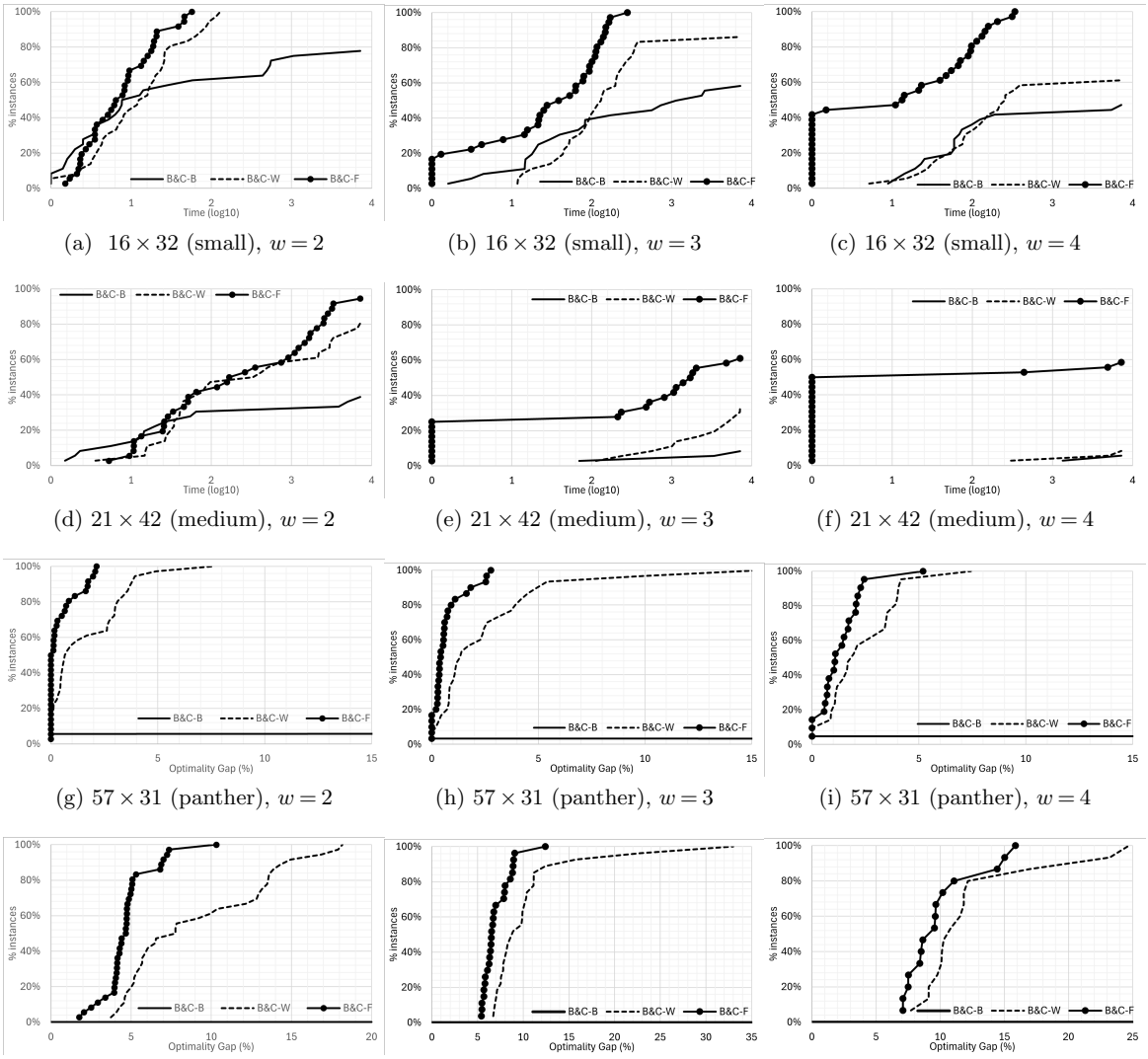
Patch costs are generated using two patterns: *flat* and *spatial kriging*. In the flat pattern, all patches have the same cost. The spatial kriging pattern captures land cost clustering, where costs are similar within nearby areas but vary across more distant locations. This pattern includes one or more high-cost regions (*peaks*) that influence the surrounding cost distribution. The procedure for generating the spatial kriging cost pattern is described in Appendix E. For every landscape, we generate cost patterns in three ways: flat, kriging with one peak, and kriging with three peaks. We set the available budget to 1.5 times the proxy of the minimum-cost corridor that is obtained by *min-cost*$(4, \mathbf{c})$. For each cost pattern in each landscape, we test all the proposed solution approaches for $w = 2$, 3, and 4 and 12 different problem-specific values of parameter $\mathcal{L}$. Parameter $\mathcal{L}$ is problem-specific as the minimum feasible number of patches depends on S, T, and w.

Figure 7 presents the performance profiles across randomly generated landscapes of varying sizes (small, medium, and large) and the panther landscape, for corridor width values $w = 2$, 3, and 4. Figures 7a-7f focus on runtime to optimality, while Figures 7g-7l highlight the optimality gap. Each plot compares the performance of B&C-B, B&C-W, and B&C-F across different $\mathcal{L}$ values and cost structures (flat, one-peak kriging, and three-peak kriging). As expected, increasing landscape size

and corridor width makes instances more challenging to solve. Figures 7a and 7d show that in some easy-to-solve cases, B&C-B outperforms B&C-W and B&C-F due to the additional time required for enhancements. However, in all other instances, B&C-F demonstrates superior performance by detecting infeasibility faster, solving instances to optimality in less time, or achieving a tighter optimality gap. B&C-F solves the most instances to optimality within the time limit across all settings, except for the small landscape with $w = 2$, where B&C-W also solves all instances. For instances that time out, B&C-F achieves better optimality gaps across all settings. Notably, B&C-B fails to find a feasible solution within the time limit for any instance in the large landscape and for nearly all instances in the panther landscape. Appendix F provides detailed results for different instance sizes, cost structures, $\mathcal{L}$ values, and corridor widths, demonstrating that B&C-F achieves significant speedups—often by several orders of magnitude—and substantially reduces the optimality gap compared to B&C-W. Appendix H summarizes the contribution of the VIs from Proposition 5.5 on the root node quality in B&C-F. The effect of the VIs is modest for small instances (less than a 0.3% improvement in root node quality) but becomes more important for larger problems. For the large (40×80) and Panther landscapes, the average improvement in the root node quality ranges between 1.2% and 8%. Appendix H also presents an ablation analysis on a representative instance to isolate the contribution of the additional enhancements.

Table 2 reports the performance of B&C-F before and after pre-processing the panther landscape using SPP for the three proposed cost patterns. Appendix G reports the same for the small, medium, and large landscapes. The **bold** value in Column $\mathcal{L}$ is the required number of patches to ensure a feasible corridor for each width. The maximum reported value of $\mathcal{L}$ is the point beyond which all patches in the landscape are reachable. Columns $t(s)$ and $t_{\text{SPP}}(s)$ show the solution time achieved by B&C-F before and after applying SPP, respectively. The time reduction is computed in column t-diff (%) as $100(t - t_{\text{SPP}})/t$. When $t > 7200$, we use 7200 as a reference time in the calculations. If both t and t_{SPP} exceed 7200, we set t-diff$= 0$. Columns $\triangle z(\%)$ and #DV record the optimality gap and the number of decision variables before applying SPP, respectively. Columns $\triangle z_{\text{SPP}}(\%)$ and $\#\text{DV}_{\text{SPP}}$ present these metrics after applying SPP. The percentage reductions in the optimality gap and the number of decision variables due to SPP are reported in columns $\triangle z$-diff(%) and #DV-diff(%), computed as $\triangle z$-diff$= 100(\triangle z - \triangle z_{\text{SPP}})/\triangle z$ and #DV-diff$= 100$ (#DV - $\#\text{DV}_{\text{SPP}}$)/#DV. If the use of SPP fully closes the optimality gap, we report $\triangle z$-diff$=\downarrow 100\%$. When both gaps are "opt.", we set $\triangle z$-diff$= 0$.

Table 2 shows that applying SPP reduces the number of decision variables across instances in the panther landscape. As expected, the performance gains diminish as $\mathcal{L}$ increases. For values of $\mathcal{L}$ beyond those reported in the table, SPP has no effect, as every patch becomes eligible. The combination of SPP and B&C-F helps close the optimality gap in some instances that would



(a) 16×32 (small), $w = 2$ (b) 16×32 (small), $w = 3$ (c) 16×32 (small), $w = 4$

(d) 21×42 (medium), $w = 2$ (e) 21×42 (medium), $w = 3$ (f) 21×42 (medium), $w = 4$

(g) 57×31 (panther), $w = 2$ (h) 57×31 (panther), $w = 3$ (i) 57×31 (panther), $w = 4$

(j) 40×80 (large), $w = 2$ (k) 40×80 (large), $w = 3$ (l) 40×80 (large), $w = 4$

Figure 7 Performance Profiles for the Panther and Randomly Generated Landscapes

otherwise time out without pre-processing. Additionally, solution times decrease significantly, in some cases by at least an order of magnitude. For instances that still time out, SPP leads to substantial reductions in the optimality gap.

The instances reported in Table 2, as well as the results presented in Appendix G, are solved sequentially to leverage the results of Propositions 5.3 and 5.6. Specifically, the values reported in column t_{SPP} exclude the runtime of SPP, as it is computed only once for the fewest number of patches associated with each landscape and width combination, and then reused for all subsequent corridor size values. This reflects practical scenarios, where practitioners typically explore multiple corridor designs derived from optimal solutions across varying widths, sizes, and budget levels.

w	$\mathcal{L}$	$t(s)$	$t_{\text{SPP}}(s)$	t-diff (%)	$\triangle z$(%)	$\triangle z_{\text{SPP}}$(%)	$\triangle z$-diff (%)	#DV	$\#\text{DV}_{\text{SPP}}$	#DV-diff (%)
					***Flat* cost pattern**					
2	**59**	>7200	0.4	↓99.99	2.07	opt.	↓100	1543	90	↓94.17
	79	5188.8	1235.9	↓76.18	opt.	opt.	0	1543	820	↓46.86
	92	1980.6	1573.3	↓20.56	opt.	opt.	0	1543	1299	↓15.81
	99	1052.2	815.5	↓22.5	opt.	opt.	0	1543	1432	↓7.19
3	**92**	>7200	0.8	↓99.99	2.53	opt.	↓100	1543	156	↓89.89
	99	>7200	1811.7	↓74.84	1.09	opt.	↓100	1543	382	↓75.24
	119	>7200	>7200	0	0.35	0.22	↓37.14	1543	810	↓47.5
	124	>7200	>7200	0	0.41	0.11	↓73.17	1543	943	↓38.89
	139	>7200	>7200	0	0.59	0.15	↓74.58	1543	1252	↓18.86
	159	>7200	>7200	0	0.29	0.16	↓44.83	1543	1503	↓2.59
4	**124**	>7200	3.1	↓99.96	2.27	opt.	↓100	1543	218	↓85.87
	139	>7200	>7200	0	2.04	0.42	↓79.41	1543	496	↓67.85
	159	>7200	>7200	0	1.4	0.97	↓30.71	1543	894	↓42.06
	179	>7200	>7200	0	1.07	0.82	↓23.36	1543	1260	↓18.34
	199	>7200	>7200	0	0.72	0.58	↓19.44	1543	1464	↓5.12
					***Kriging* cost pattern with one peak**					
2	**59**	>7200	0.5	↓99.99	2.13	opt.	↓100	1543	90	↓94.17
	79	6861.1	612.1	↓91.08	opt.	opt.	0	1543	820	↓46.86
	92	>7200	3522.7	↓51.07	0.13	opt.	↓100	1543	1299	↓15.81
	99	>7200	7023.1	↓2.46	0.16	opt.	↓100	1543	1432	↓7.19
	119	>7200	4616.4	↓35.88	0.22	opt.	↓100	1543	1511	↓2.07
3	**92**	>7200	1	↓99.99	2.57	opt.	↓100	1543	156	↓89.89
	99	>7200	3779.2	↓47.51	1.62	opt.	↓100	1543	382	↓75.24
	119	>7200	>7200	0	0.72	0.25	↓65.28	1543	810	↓47.5
	124	>7200	>7200	0	0.42	0.27	↓35.71	1543	943	↓38.89
	139	>7200	>7200	0	0.36	0.2	↓44.44	1543	1252	↓18.86
	159	>7200	>7200	0	0.31	0.2	↓35.48	1543	1503	↓2.59
4	**124**	>7200	3.2	↓99.96	2.14	opt.	↓95.33	1543	218	↓85.87
	139	>7200	>7200	0	2.06	0.34	↓83.5	1543	496	↓67.85
	159	>7200	>7200	0	1.02	0.66	↓35.29	1543	894	↓42.06
	179	>7200	>7200	0	0.79	0.76	↓3.8	1543	1260	↓18.34
	199	3011.2	2303.2	↓23.51	opt.	opt.	0	1543	1464	↓5.12
					***Kriging* cost pattern with three peaks**					
2	**59**	>7200	0.5	↓99.99	1.97	opt.	↓100	1543	90	↓94.17
	79	2174.1	997.7	↓54.11	opt.	opt.	0	1543	820	↓46.86
	92	3011.5	1213.3	↓59.71	opt.	opt.	0	1543	1299	↓15.81
	99	2312.6	2019.7	↓12.67	opt.	opt.	0	1543	1432	↓7.19
3	**92**	>7200	1	↓99.99	2.77	opt.	↓100	1543	156	↓89.89
	99	>7200	2432.5	↓66.22	1.82	opt.	↓100	1543	382	↓75.24
	119	>7200	>7200	0	0.6	0.15	↓75.0	1543	810	↓47.5
	124	>7200	>7200	0	0.75	0.17	↓77.33	1543	943	↓38.89
	139	>7200	>7200	0	0.48	0.29	↓76.61	1543	1252	↓18.86
	159	>7200	>7200	0	0.53	0.33	↓37.74	1543	1503	↓2.59
4	**124**	>7200	5	↓99.93	2.44	opt.	↓100	1543	218	↓85.87
	139	>7200	>7200	0	1.72	0.62	↓63.95	1543	496	↓67.85
	159	>7200	>7200	0	1.19	1.09	↓8.4	1543	894	↓42.06
	179	>7200	>7200	0	0.61	0.51	↓16.39	1543	1260	↓18.34
	199	>7200	>7200	0	6.82	5.19	↓23.90	1543	1464	↓5.12

Table 2 B&C-F Performance with and without SPP pre-processing for 57×31 (panther) grid landscape

Table 3 summarizes the SPP runtime. For small and medium landscapes, the SPP pre-processing time is minimal and yields performance improvements across all instances, even when added to the B&C-F solution time. For the panther landscape, the SPP procedure is more computationally intensive, but its cost is amortized across all applicable instances, resulting in overall performance gains. However, this is not true for the large landscape, where the SPP pre-processing time is substantial and does not yield a viable acceleration strategy.

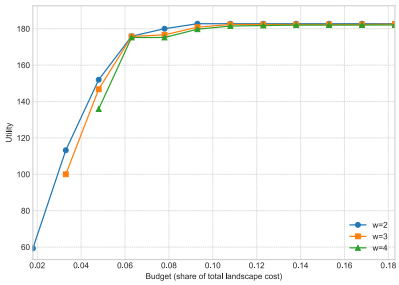
We emphasize that SPP is executed only once per landscape and can subsequently be reused for all cost scenarios associated with that landscape. This feature is particularly valuable in practice: when practitioners evaluate multiple financial scenarios or stakeholder-specific cost structures, SPP needs to be performed only once. Moreover, SPP can be executed in a nested manner: patches that are not reachable for a given maximum allowable corridor size will remain unreachable for any smaller allowable size. Each additional scenario then benefits from the reduced problem size without incurring further pre-processing overhead. Appendix G provides the complete set of results and a more in-depth analysis of this situation.

Landscape size	w	$t(s)$
16×32 (small)	2	44.2
	3	31.7
	4	23.1
21×42 (medium)	2	365.7
	3	269.7
	4	214.1
57×31 (panther)	2	1672.4
	3	1345.2
	4	1227.4
40×80 (large)	2	27979.1
	3	24113.3
	4	21510.1

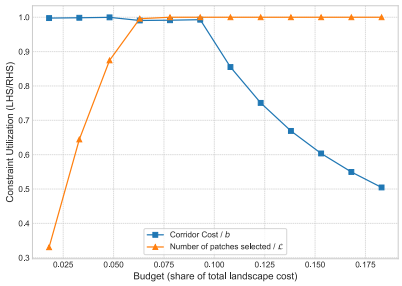
Table 3 Performance of SSP for different landscapes and w

6.3. Sensitivity Analysis

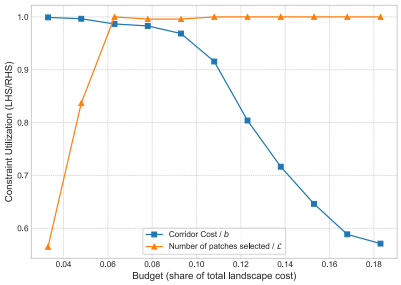
We perform a sensitivity analysis to evaluate how corridor utility responds to changes in the available budget (b) and the allowed corridor size ($\mathcal{L}$), using the Florida panther landscape. Figure 8 reports the optimal utility for varying budgets under the one-peak kriging cost pattern with $\mathcal{L} = 239$. For a fixed budget, the utilities satisfy $U_{\mathcal{C}_2} \geq U_{\mathcal{C}_3} \geq U_{\mathcal{C}_4}$, consistent with Proposition 5.6, as narrower corridors allow greater flexibility in patch selection. The figure also illustrates diminishing returns, with large gains at low budget levels and marginal or no improvements in utility thereafter. Figures 8b–d illustrate the constraint dynamics by showing how the binding constraint between the budget (i.e., Constraint (1e)) and the allowable number of patches (i.e., Constraint (1d)) changes with increasing budget for $w = 2, 3, 4$. At low budget levels, the budget constraint is binding; at intermediate budgets, the two constraints may bind jointly (or nearly so); and at high budget levels, the allowable number of patches becomes the binding constraint. For all widths, the marginal utility gain from increasing the budget drops sharply once the corridor size constraint becomes binding, as expected. This analysis highlights the tradeoff between utility, financial constraints, and geometric requirements. Appendix I presents additional sensitivity analyses, including the relationships between utility and budget and between utility and corridor size for the flat and multi-peak cost patterns.



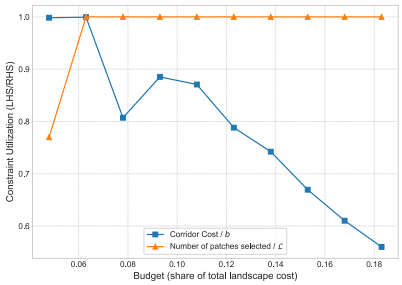
(a) Utility vs. budget for $w = 2, 3, 4$.



(b) Constraint utilization vs. budget for $w = 2$.



(c) Constraint utilization vs. budget for $w = 3$.



(d) Constraint utilization vs. budget for $w = 4$.

Figure 8 **Utility vs. budget and constraint utilization under one-peak kriging cost pattern, $w = 2, 3, 4$, and $\mathcal{L} = 239$.**

7. Concluding Remarks

We introduce a mathematical formulation, solution procedure, and algorithmic enhancements to address a practical challenge in conservation planning. The proposed model incorporates key geometric features—patch connectivity, corridor size (number of patches), and width—into the design of ecological corridors connecting fragmented habitats. It also supports the integration of existing protected areas into the corridor network and promotes redundancy by identifying multi-branch corridors that satisfy the specified geometric criteria. These capabilities are essential for leveraging existing conservation investments and for identifying routes that avoid high-mortality areas (e.g., human-wildlife conflict). The proposed algorithmic enhancements substantially increase the size of instances that can be solved to optimality or yield improved optimality gaps within a two-hour time limit.

The proposed approach also accommodates alternative shapes for the Ω-sets used to enforce width. These sets can be tailored to specific areas—for example, to account for predator zones or to require wider corridor segments in regions where human–wildlife conflict mitigation is necessary. Our approach supports the design of multi-branch corridors, which can enhance redundancy and ecological resilience. Our model not only provides a formal framework to identify optimal corridors

under such constraints but also allows for easy integration of additional design features (e.g., pre-protected patches), offering a flexible and scalable tool for real-world conservation planning.

Although the proposed methods enable the solution of large instances, there remains room for improvement at the pre-processing stage. Although the SPP approach is designed to be generic for regularly partitioned landscapes, its computational performance can be prohibitive for certain instances, limiting its practical usability. Further improvements could be achieved by exploiting the structural properties inherent in grid-based landscapes. Additionally, alternative strategies for reducing the number of candidate patches—such as cost-based filtering—may be more effective in settings where the budget constraint is more binding than the limit on the allowable number of patches in the corridor.

A limitation of this study is its focus on a single-species analysis. In principle, the model can capture patch utilities defined as weighted aggregations of species-specific suitability scores, reflecting the multi-objective nature of maximizing habitat value across species while enforcing shared geometric constraints. However, accommodating multiple species with distinct ecological requirements affecting the corridor geometry would likely require substantial modifications to the formulation. We view this as an important direction for future research. Additional promising extensions include the use of irregular landscape discretizations (e.g., based on property boundaries), the design of multiple corridors connecting several fragmented habitats, and the incorporation of edge-sensitivity mitigation strategies along corridor boundaries, enabling the model to dynamically determine whether a given patch requires edge mitigation and to assign costs accordingly.

An interesting direction for future work is the development of online constraints elicited by the user during visual inspection of the corridor. Such a mechanism could guide the design process by directly incorporating shape preferences, rather than relying solely on closed form descriptions or automated separation procedures.

8. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 2308687 (Sefair) and 1655555 (Fletcher). The authors gratefully acknowledge Lindsay E. Carrillo, an undergraduate researcher at the University of Florida, for her valuable contributions to the development of the Florida Panther case study. The authors would also like to thank the Editor and the four anonymous reviewers for their valuable suggestions and constructive feedback, which helped improve the manuscript.

References

Acevedo, M., J. A. Sefair, J. C. Smith, B. Reichert, R. J. Fletcher Jr. 2015. Conservation under uncertainty: Optimal network protection strategies for worst-case disturbance events. *Journal of Applied Ecology* **52**(6) 1588–1597.

Álvarez-Miranda, E., M. Goycoolea, I. Ljubić, M. Sinnl. 2021. The generalized reserve set covering problem with connectivity and buffer requirements. *European Journal of Operational Research* **289**(3) 1013–1029.

Barlow, J., G. D. Lennox, J. Ferreira, E. Berenguer, A. C. Lees, R. M. Nally, J. R. Thomson, S. F. D. B. Ferraz, J. Louzada, V. H. F. Oliveira, et al. 2016. Anthropogenic disturbance in tropical forests can double biodiversity loss from deforestation. *Nature* **535**(7610) 144–147.

Beger, M., H. S. Grantham, R. L. Pressey, K. A. Wilson, E. L. Peterson, D. Dorfman, P. J. Mumby, R. Lourival, D. R. Brumbaugh, H. P. Possingham. 2010. Conservation planning for connectivity across marine, freshwater, and terrestrial realms. *Biological Conservation* **143**(3) 565–575.

Beier, P. 2019. A rule of thumb for widths of conservation corridors. *Conservation Biology* **33**(4) 976–978.

Belle, E., N. Kingston, N. Burgess, T. Sandwith, N. Ali, K. MacKinnon. 2018. Protected planet report 2018. *UNEP-WCMC, IUCN and NGS: Cambridge UK, Gland Switzerland and Washington, DC, USA* .

Billionnet, A. 2013. Mathematical optimization ideas for biodiversity conservation. *European Journal of Operational Research* **231**(3) 514–534.

Burgman, M. A., D. B. Lindenmayer, J. Elith. 2005. Managing Landscapes for Conservation Under Uncertainty. *Ecology* **86**(8) 2007–2017.

Camm, J. D., S. K. Norman, S. Polasky, A. R. Solow. 2002. Nature reserve site selection to maximize expected species covered. *Operations Research* **50**(6) 946–955.

Carvajal, R., M. Constantino, M. Goycoolea, J. P. Vielma, A. Weintraub. 2013. Imposing Connectivity Constraints in Forest Planning Models. *Operations Research* **61**(4) 824–836.

Cerdeira, J. O., L. S. Pinto. 2005. Requiring connectivity in the set covering problem. *Journal of combinatorial optimization* **9**(1) 35–47.

Chaigneau, Charly, Nathalie Bostel, Axel Grimault. 2025. A large neighborhood search-based approach to tackle the very large scale team orienteering problem in industrial context. *Computers & Operations Research* **176** 106954.

Church, R. L., D. M. Stoms, F. W. Davis. 1996. Reserve selection as a maximal covering location problem. *Biological Conservation* **76**(2) 105–112.

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

Cormen, T. H. 2009. *Introduction to algorithms*. Third edition ed. MIT Press, Cambridge, Massachusetts.

Costello, C., S. Polasky. 2004. Dynamic reserve site selection. *Resource and Energy Economics* **26**(2) 157–174.

Dilkina, B., C. P. Gomes. 2010. Solving connected subgraph problems in wildlife conservation. *International conference on integration of artificial intelligence and operations research techniques in constraint programming*. Springer, 102–116.

Dinerstein, E., C. Vynne, E. Sala, A. R. Joshi, S. Fernando, T. E. Lovejoy, J. Mayorga, D. Olson, G. P. Asner, J. E. Baillie, et al. 2019. A global deal for nature: guiding principles, milestones, and targets. *Science Advances* **5**(4) eaaw2869.

Downs, J. A., M. W. Horner. 2012. Enhancing habitat connectivity in fragmented landscapes: Spatial modeling of wildlife crossing structures in transportation networks. *Annals of the Association of American Geographers* **102**(1) 17–34.

Fletcher Jr., R. J., M. J. Fortin. 2018. *Spatial ecology and conservation modeling: applications with R*. Springer Publishing, Cham, Switzerland.

Fletcher Jr., R. J., J. A. Sefair, C. Wang, C. L. Poli, T. A. Smith, E. M . Bruna, R. D. Holt, M. Barfield, A. J. Marx, M. A. Acevedo. 2019. Towards a unified framework for connectivity that disentangles movement and mortality in space and time. *Ecology Letters* **22**(10) 1680–1689.

Fletcher Jr, Robert J, Jorge A Sefair, Nicholas Kortessis, Roldolfo Jaffe, Robert D Holt, Ellen P Robertson, Sarah I Duncan, Andrew J Marx, James D Austin. 2022. Extending isolation by resistance to predict genetic connectivity. *Methods in Ecology and Evolution* **13**(11) 2463–2477.

Fredman, M. L., R. E. Tarjan. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)* **34**(3) 596–615.

Garey, M. R., D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman & Co., Princeton, NJ.

Geldmann, J., M. Barnes, L. Coad, I. D. Craigie, M. Hockings, N. D. Burgess. 2013. Effectiveness of terrestrial protected areas in reducing habitat loss and population declines. *Biological Conservation* **161**(C) 230–238.

Goycoolea, M., A. Murray, F. Barahona, R. Epstein, A. Weintraub. 2005. Harvest scheduling subject to maximum area restrictions: Exploring exact approaches. *Operations Research* **53**(3) 490–500.

Gunawan, Aldy, Hoong Chuin Lau, Pieter Vansteenwegen. 2016. Orienteering problem: A survey of recent variants, solution approaches and applications. *European journal of operational research* **255**(2) 315–332.

Haight, R. G., C. S. Revelle, S. A. Snyder. 2000. An integer optimization approach to a probabilistic reserve site selection problem. *Operations Research* **48**(5) 697–708.

Hodgson, J. A., C. D. Thomas, B. A. Wintle, A. Moilanen. 2009. Climate change, connectivity and conservation decision making: back to basics. *Journal of Applied Ecology* **46**(5) 964–969.

IUCN. 2021. The international union for conservation of nature's red list of threatened species. URL `https://www.iucnredlist.org/`.

Jafari, Nahid, John Hearne. 2013. A new method to solve the fully connected reserve network design problem. *European Journal of Operational Research* **231**(1) 202–209.

Kautz, R., R. Kawula, T. Hoctor, J. Comiskey, D. Jansen, D. Jennings, J. Kasbohm, F. Mazzotti, R. McBride, L. Richardson, K. Root. 2006. How much is enough? landscape-scale conservation for the florida panther. *Biological conservation* **130**(1) 118–133.

Kindlmann, P., F. Burel. 2008. Connectivity measures: A review. *Landscape Ecology* **23**(8) 879–890.

Lai, K., C. Gomes, M. Schwartz, K. McKelvey, D. Calkin, C. Montgomery. 2011. The steiner multigraph problem: wildlife corridor design for multiple species. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25. 1357–1364.

Mace, G. M., K. Norris, A. H. Fitter. 2012. Biodiversity and ecosystem services: a multilayered relationship. *Trends in Ecology and Evolution* **27**(1) 19–26.

Mair, L., E. Amorim, M. Bicalho, T. M. Brooks, V. Calfo, R. D. T. Capellão, C. Clubbe, M. Evju, E. P. Fernandez, G. C. Ferreira, et al. 2023. Quantifying and mapping species threat abatement opportunities to support national target setting. *Conservation Biology* **37**(1) e14046.

Mair, L., L. A. Bennun, T. M. Brooks, S. H. Butchart, F. C. Bolam, N. D. Burgess, J. M. Ekstrom, E. Milner-Gulland, M. Hoffmann, K. Ma, et al. 2021. A metric for spatially explicit contributions to science-based species targets. *Nature Ecology and Evolution* **5**(6) 836–844.

Margules, C. R., R. L. Pressey. 2000. Systematic conservation planning. *Nature* **405**(6783) 243–253. doi:10.1038/35012251.

Marianov, V., C. Revelle, S. Snyder. 2008. Selecting compact habitat reserves for species with differential habitat size needs. *Computers and Operations Research* **35**(2) 475–487.

Marx, A. J., C. Wang, J. A. Sefair, M. A. Acevedo, R. J. Fletcher Jr. 2020. SAMC: an R package for connectivity modeling with spatial absorbing markov chains. *Ecography* **43**(4) 518–527.

Matsypura, D., O. A. Prokopyev, A. Zahar. 2018. Wildfire fuel management: Network-based models and optimization of prescribed burning. *European Journal of Operational Research* **264**(2) 774–796.

Mcdill, M., S. Rebain, J. Braze. 2002. Harvest scheduling with area-based adjacency constraints. *Forest Science* **48**(4) 631–642.

McRae, B. H., B. G. Dickson, T. H. Keitt, V. B. Shah. 2008. Using circuit theory to model connectivity in ecology, evolution, and conservation. *Ecology* **89**(10) 2712–2724.

Öhman, K., P. Wikström. 2008. Incorporating aspects of habitat fragmentation into long-term forest planning using mixed integer programming. *Forest Ecology and Management* **255**(3) 440–446.

Önal, H., R. A. Briers. 2006. Optimal selection of a connected reserve network. *Operations Research* **54**(2) 379–388.

Önal, H., Y. Wang, S. TM Dissanayake, J. D. Westervelt. 2016. Optimal design of compact and functionally contiguous conservation management areas. *European Journal of Operational Research* **251**(3) 957–968.

Perkl, R. M. 2016. Geodesigning landscape linkages: Coupling gis with wildlife corridor design in conservation planning. *Landscape and Urban Planning* **156**(D) 44–58.

Peterson, A. T., M. Papeş, J. Soberón. 2008. Rethinking receiver operating characteristic analysis applications in ecological niche modeling. *Ecological Modelling* **213**(1) 63–72.

Phillips, S. J., P. Williams, G. Midgley, A. Archer. 2008. Optimizing dispersal corridors for the cape proteaceae using network flow. *Ecological Applications* **18**(5) 1200–1211.

Pinto, N., T. H. Keitt. 2009. Beyond the least-cost path: evaluating corridor redundancy using a graph-theoretic approach. *Landscape Ecology* **24**(2) 253–266.

Prasad, B. R. G., R. M. Borges. 2006. Searching on patch networks using correlated random walks: Space usage and optimal foraging predictions using Markov chain models. *Journal of Theoretical Biology* **240**(2) 241–249.

Préau, C., F. Grandjean, Y. Sellier, M. Gailledrat, R. Bertrand, F. Isselin-Nondedeu. 2020. Habitat patches for newts in the face of climate change: local scale assessment combining niche modelling and graph theory. *Scientific Reports* **10**(1) 3570.

Pressey, R. L., M. Cabeza, M. E. Watts, R. M. Cowling, K. A. Wilson. 2007. Conservation planning in a changing world. *Trends in Ecology and Evolution* **22**(11) 583–592.

Rachmawati, R., M. Ozlen, J. Hearne, K. Reinke. 2018. Fuel treatment planning: Fragmenting high fuel load areas while maintaining availability and connectivity of faunal habitat. *Applied Mathematical Modelling* **54**(C) 298–310.

Rodrigues, A. S. L., H. R. Akcakaya, S. J. Andelman, M. I. Bakarr, L. Boitani, T. M. Brooks, J. S. Chanson, L. D. C. Fishpool, G. A. B. Da Fonseca, K. J. Gaston, et al. 2004a. Global gap analysis: priority regions for expanding the global protected-area network. *BioScience* **54**(12) 1092–1100.

Rodrigues, A. S. L., S. J. Andelman, M. I. Bakarr, L. Boitani, T. M. Brooks, R. M. Cowling, L. D. C. Fishpool, G. A. B. D. Fonseca, K. J. Gaston, M. Hoffmann, J. S. Long, P. A. Marquet, J. D. Pilgrim, R. L. Pressey, J. Schipper, W. Sechrest, S. N. Stuart, L. G. Underhill, R. W. Waller, M. E. J. Watts, X. Yan. 2004b. Effectiveness of the global protected area network in representing species diversity. *Nature* **428**(6983).

Sefair, J. A. 2022. Optimization for conservation planning. *Encyclopedia of Optimization*. Springer, 1–9.

Sefair, J. A., J. C. Smith, M. A. Acevedo, R. J. Fletcher JR. 2017. A defender-attacker model and algorithm for maximizing weighted expected hitting time with application to conservation planning. *IISE Transactions* **49**(12) 1112–1128.

Shirabe, T. 2005. A model of contiguity for spatial unit allocation. *Geographical Analysis* **37**(1) 2–16.

St John, R., S. Tóth, Z. Zabinsky. 2018. Optimizing the geometry of wildlife corridors in conservation reserve design. *Operations Research* **66**(6) 1471–1485.

Tóth, S. F., R. G. Haight, L. W. Rogers. 2011. Dynamic reserve selection: Optimal land retention with land-price feedbacks. *Operations Research* **59**(5) 1059–1078.

Tóth, S. F., R. G. Haight, S. A. Snyder, S. George, J. R. Miller, M. S. Gregory, A. M. Skibbe. 2009. Reserve selection with minimum contiguous area restrictions: An application to open space protection planning in suburban chicago. *Biological Conservation* **142**(8) 1617–1627.

Validi, H., A. Buchanan, E. Lykhovyd. 2022. Imposing contiguity constraints in political districting models. *Operations Research* **70**(2) 867–892.

Vansteenwegen, Pieter, Wouter Souffriau, Dirk Van Oudheusden. 2011. The orienteering problem: A survey. *European journal of operational research* **209**(1) 1–10.

Wang, Y., P. Qin, H. Önal. 2022. An optimisation approach for designing wildlife corridors with ecological and spatial considerations. *Methods in Ecology and Evolution* **13**(5) 1042–1051.

Warren, D. L., S. N. Seifert. 2011. Ecological niche modeling in maxent: the importance of model complexity and the performance of model selection criteria. *Ecological applications* **21**(2) 335–342.

Watson, J. E., O. Venter. 2017. A global plan for nature conservation. *Nature* **550**(7674) 48–49.

Williams, J., C. ReVelle, S. Levin. 2005. Spatial attributes and reserve design models: A review. *Environmental Modeling and Assessment* **10**(3) 163–181.

Wood, C. L., K. D. Lafferty, G. DeLeo, H. S. Young, P. J. Hudson, A. M. Kuris. 2014. Does biodiversity protect humans against infectious disease? *Ecology* **95**(4) 817–832.

WWF. 2020. *Living Planet Report 2020: Bending the curve of biodiversity loss*. World Wildlife Fund.

Yemshanov, D., R. G. Haight, N. Liu, R. Rempel, F. H. Koch, A. Rodgers. 2022. Exploring the tradeoffs among forest planning, roads and wildlife corridors: a new approach. *Optimization Letters* **16**(3) 747–788.

Yen, J. Y. 1971. Finding the k shortest loopless paths in a network. *Management Science* **17**(11) 712–716.

Zhang, Rui, Yafeng Lu, Katherine Adams, Jorge A Sefair, Haley Mellin, Miguel A Acevedo, Ross Maciejewski. 2021. A visual analytics framework for conservation planning optimization. *Environmental Modelling & Software* **145** 105178.