Submitted to *Operations Research*
manuscript tbd

Optimizing the Geometry of Wildlife Corridors in Conservation Reserve Design

Rachel St John
Department of Industrial and Systems Engineering, University of Washington, Box 352650, Seattle, WA 98195, USA
E-mail:rkrieg@u.washington.edu; Tel.: +1-206-228-9007

Sándor F. Tóth
School of Environmental and Forest Sciences, University of Washington, Box 352100, Seattle, WA 98195, USA
E-mail:toths@u.washington.edu; Tel.: +1-206-518-4978

Zelda B. Zabinsky
Department of Industrial and Systems Engineering, University of Washington, Box 352650, Seattle, WA 98195, USA
E-mail:zelda@u.washington.edu; Tel.: +1-206-543-4607

Wildlife corridors are often used to connect critical habitat for species protection. Mixed integer programming models have been used in the past to create wildlife corridors, but they lack the capacity to control corridor geometry. We propose an approach that employs path planning techniques from artificial intelligence to account for and control corridor geometry, such as width and length. By combining path planning with network optimization, our approach allows the user to control and optimize the geometric characteristics of wildlife corridors. We illustrate our approach on two realistic landscapes and present numerical results on several computer-generated landscapes. The computational results indicate that this approach is efficient and can address problems controlling corridor geometry that were previously thought intractable. The approach has potential applications in such areas as the selection of routes or barrier construction problems, an example of which is fire break design.

Key words: network optimization, artificial intelligence, natural resources, path planning

History: This paper was first submitted on October 14, 2015.

Acknowledgments

This research was partially funded by the University of Washington's Precision Forestry Cooperative. We would like to thank the anonymous reviewers and the Associate Ediotr for their constructive comments that helped improve this paper.

1. Bios

Rachel St. John holds a Ph.D. in Industrial and Systems Engineering and an M.S. in Quantitative Ecology and Resource Management, both from the University of Washington, Seattle. Her research interests include spatial optimization, goal programming and decision making in natural resource management.

Sándor F. Tóth is Donald J. and Robert G. McLachlan Associate Professor of Natural Resource Informatics at the University of Washingtons School of Environmental and Forest Sciences in Seattle with an adjunct appointment at the Department of Industrial and Systems Engineering. Currently, he is a Fulbright Fellow at the Univeristy of Chiles Institute of Complex Systems Engineering in Santiago. He specializes in spatial optimization modeling of environmental problems, particularly those where multiple conflicting objectives are present.

Zelda B. Zabinsky is a Professor in the Industrial and Systems Engineering Department at the University of Washington, with adjunct appointments in the departments of Electrical Engineering, Mechanical Engineering, and Civil and Environmental Engineering. Her research interests include global optimization, optimization under uncertainty, analysis of algorithm performance, and many different engineering applications, including the wildlife corridor problem addressed in this paper.

2. Introduction

As human activity expands across the landscape, natural areas dwindle. From 1963 to 1997, six million hectares of forest land in the United States were lost to development, and an additional 9.4 million hectares are projected to disappear by 2050 (Alig et al. 2003). As human activity encroaches on previously undisturbed natural areas, large, contiguous patches of habitat areas shrink and become disconnected. These newly formed “islands” are often inadequate for sustaining the existing wildlife, which, in many cases, results in the loss of entire wildlife populations.

Habitat fragmentation separates critical resources and reduces the amount of available space for wildlife populations to grow, move and disperse. One method to reduce fragmentation and increase connectivity is to create wildlife corridors (Beier and Noss 1998). Wildlife corridors provide paths for species in need of protection to move freely between conserved areas that are otherwise separated by human activity. They facilitate access to resources, migration, dispersal and population mixing (Beier 1993). Major wildlife corridors include the Paso del Jaguar, a corridor for jaguars stretching from the southern United States to Argentina; the European Green Belt, a corridor for a variety of species running from Norway to Turkey; and the Siju-Rewak Corridor in India, which protects over 139 species of mammals, including Asian elephants, Bengal tigers, clouded leopards and Himalayan black bears (Panthera.org, EuropeanGreenBelt.org, WorldLandTrust.org).

Wildlife corridors are often designed on a two-dimensional landscape. Property lines and natural boundaries (such as rivers or cliffs) typically partition the landscape into polygon-shaped parcels (see Figure 1). The core reserves (dark parcels in Figure 1) may be connected via a collection of other parcels (lightly shaded parcels in Figure 1) - a wildlife corridor. To be successful, wildlife corridors must be conducive to travel. In particular, corridors cannot be too narrow or too long (Soule and Gilpin 1991). Narrow corridors do not provide adequate buffer from the surrounding unsuitable habitat, and long, winding corridors are difficult to negotiate for some species. Given the spatial configuration of the landscape and corridor width and length requirements, it is not obvious which set of parcels will create an ideal corridor. The combinatorial nature of the problem makes corridor selection difficult for even small landscapes. Mixed integer programming (MIP) models have been used for corridor design in the past, but lack the capacity to explicitly control corridor geometry (e.g. width or length) without making assumptions about parcel shape and size.

We introduce a new approach that combines techniques from path planning (artificial intelligence) with network optimization models in order to calculate and control geometric characteristics of corridors, such as width and length. This approach allows the user to optimize corridor construction by selecting parcels in landscapes of any spatial configuration. By borrowing concepts from path planning, we introduce models that can explicitly control geometric characteristics that were not incorporated in previous spatial optimization studies. Our method can extend beyond wildlife corridors to other problems involving connectivity in spatial optimization, such as constructing fire

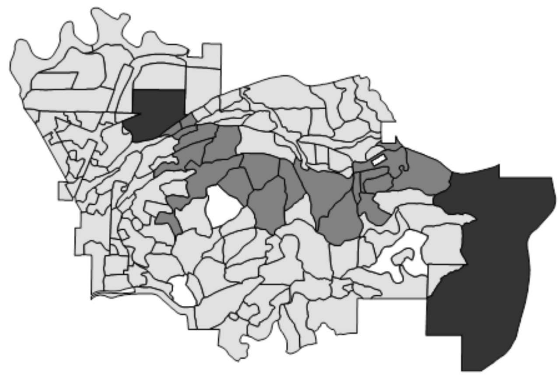

Figure 1 A landscape with a corridor.

breaks and planning emergency escape routes subject to budget constraints. In this paper, we first discuss the state-of-the art in corridor design and provide a brief introduction to the field of path planning to motivate its use in spatial optimization. We then describe our proposed approach and present an illustrative case study. We also conduct numerical experiments on computer-generated landscapes to demonstrate the computational efficiency of the approach. Lastly, we discuss the use of the method in potential applications beyond reserve design.

3. Background

3.1. Wildlife Reserves and Corridors

In land management, decision makers can create wildlife reserves by preserving or restoring parcels on the landscape that contain critical habitat. Along with habitat conservation, habitat connectivity is crucial to the vitality of some wildlife populations. One method for increasing the connectivity of wildlife reserves is to create spatial linkages called corridors (Beier and Noss 1998). A corridor should have certain spatial attributes to be successful. Some of these attributes are specific to the target population (such as width), while others are universally important, such as habitat area.

To be of use to wildlife, corridors must contain suitable habitat. Many species have specific needs as to what type of terrain and vegetation they need to survive and disperse. For prey species and birds that prefer interior forest habitat, a patchy corridor may not offer sufficient protection against predation (Soule and Gilpin 1991). Grizzly bears in British Columbia show preference for high elevation Douglas fir forests and abundant vegetation (Proctor et al. 2008), while transient chipmunks like to travel along fence rows through farmland (Bennett et al. 1994). Lastly, the structure of habitat such as tree or shrub density within corridors can also pose obstacles to movement, as was the case with migrating reindeer in Sweden. Such structural obstacles can be modeled with "resistance" coefficients, as in Le Bras et al. (2013), whereby the shortest resistance-weighted paths are sought between designated habitat patches.

Corridors cannot be too narrow either (Beier and Loe 1992, Williams et al. 2005). Animals that benefit from wildlife corridors prefer to avoid human interaction and areas of exposure. Anderson et al. (1977) studied birds in corridors of various widths, and found strong positive correlation between corridor width and the abundance of several species. In addition to birds, some mammals are also sensitive to corridor width. Simulation studies conducted by Soule and Gilpin (1991) show that the probability of corridor success increases with width until it reaches an asymptote. Harrison (1992) reviewed studies on seven different mammal populations and compiled a list of minimum corridor widths. The author highlights the critical research need to determine "the minimum width of effective corridors." Coster et al. (2014) proposed a method for calculating habitat width requirements and demonstrated the use of the method with migrating amphibians in Maine.

Lastly, corridors cannot be too long. If the distance between two resources is too great, or the corridor itself winds around the landscape, making an unnecessarily long path, animals will be less likely to use it. For slow-moving prey species, long corridors can be counter-productive as they can give fast-moving predators an advantage (Soule and Gilpin 1991). Brodie et al. (2015) identify length as a critical attribute of wildlife corridors. Long and narrow corridors are poor connectors.

3.2. Mathematical Programming for Reserve Selection

Reserve selection models select a subset of parcels that best meet certain conservation goals. In spatial reserve design, some of these goals are best met if the spatial configuration of the resulting reserve networks are as conducive to the survival of species in need of protection as possible. These spatial attributes include area, compactness, contiguity and shape. See Williams et al. (2005), or more recently, Billionnet (2013) for detailed reviews on spatial reserve design. In some projects, including wildlife corridors, the subset of sites to be selected must be spatially connected to encourage the movement and immigration of species in need of protection. Sessions (1992) was among the first authors to model these problems as graphs, where each parcel is represented by a node, and the adjacencies among the parcels are represented by edges (Fig. 2). Sessions posed the connected reserve selection problem as a Steiner network problem in which one must find the minimum-cost path that connects a given set of nodes. Williams (1998) was the first to use integer programming to solve the Steiner network problem for wildlife corridors and later used primal-dual graphs to represent continguous land acquisition problems as minimum spanning trees of a predetermined size (Williams 2002).

Shirabe (2005)'s network flow model, in which a single parcel is preselected, and all other parcels in the reserve must be spatially linked to it, is another example of a mathematical programming formulation for full connectivity. Önal and Briers (2006) introduced the first exact, integer programming model to create an "unrooted" connected reserve where the number of selected parcels is minimized and none of the parcels need to be preselected. The authors use linear inequalities and a tail function to break network cycles thereby ensuring full connectivity. In order to improve computational performance, a two step preprocessing technique was introduced to reduce problem size. While addressing the unrooted connectivity problem, their model can be extended to the case of rooted corridors as well.

More recently, Conrad et al. (2012) built an exact, integer programming-based network flow model to determine fully connected reserves. They introduced a two-phase method for improving solution time where a minimum-cost Steiner tree is found first (if one exists), and then it is used as an initial solution in the MIP-based optimization model. As in Önal and Briers (2006), extra constraints can be added to accommodate corridors with specific sinks. Jafari and Hearne (2013) present yet another example of unrooted connectivity optimization. Dilkina and Gomes (2010) compare a Steiner network formulation to single and multi-commodity network flow formulations.

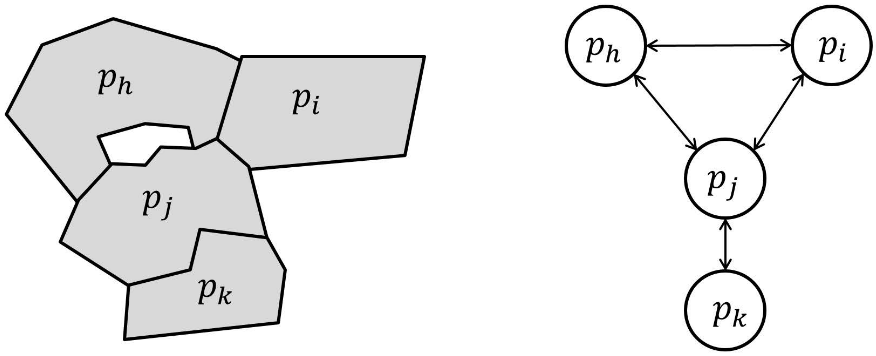

Figure 2 A landscape and its graph-theoretical representation.

They report that using their algorithm, the Steiner network model could be solved two orders of magnitude faster than the network flow formulations. Lastly, Álvarez-Miranda et al. (2013) present three new integer programming models for connectivity using only node variables. The authors compare their models and theoretically show that one of the formulations guarantees LP bounds as tight as those in Dilkina and Gomes (2010).

The need for continuous canopy corridors, or corridors of forest stands above a certain minimum age often arises in managed forest ecosystems as well. For example, Carvajal et al. (2013) introduced a model that uses only node variables and implemented a cutting-plane approach to achieve computational tractability on large scale harvest scheduling problems requiring connected reserves.

Several closely related reserve design studies proposed alternative approaches that relaxed either the exact optimality or absolute connectivity requirements. Heuristic approaches for connectivity optimization include Brás et al. (2013), that finds a minimum Steiner Tree approximation, and Cerdeira et al. (2005), that finds an approximately optimum solution for the connected set covering problem. Some models minimize spatial discontinuities within the reserve by minimizing the gaps between reserve fragments (Önal and Briers 2005, Önal and Wang 2008). Others maximized the size of contiguous habitat patches (Tóth et al. 2009).

While the problem of connectivity has been addressed with integer programming, little has been done to explicitly incorporate width and length of wildlife corridors into the models. The issue is either not addressed, or it is assumed that corridors that are one parcel wide are sufficient (Williams 1998). Although the model proposed by Conrad et al. (2012) allows corridors to be multiple parcels wide, there is no guarantee that the actual width of these corridors would meet a given threshold because the parcels can take various shapes and sizes. Unless the landscape is partitioned into a square grid, as was done for the Grizzly bear corridors in Conrad et al. (2012), the actual width of multiple adjacent parcels put together may be too narrow. In practice, the landscape might not be easily partitioned into a grid. Boundaries such as mountains, coastline and property lines result in parcels that can be highly irregular in shape, and may even have holes due to lakes or developed areas (e.g., Figure 1). On landscapes such as these, it is not clear how to create optimal corridors with geometric requirements or goals. When the landscape is translated to a network graph, such as in Figure 2, much of the information on parcel geometry is lost. Thus, analysts must either ignore characteristics such as width and length, or superimpose grid partitioning on the irregularly partitioned landscape, leading to a suboptimal solution.

Current tools in mathematical programming cannot control geometric characteristics on an irregularly partitioned landscape. Without constraints on spatial configuration, characteristics such as width and length cannot be measured. Also, the current graph-theoretical representation of the landscape does not allow for incorporation of such characteristics into an integer program. We require a means to calculate geometric characteristics of landscapes, as well as a new graph-theoretical interpretation of the landscape. For measuring geometric characteristics, we adopt techniques from an area of study arising in artificial intelligence called path planning, discussed in the next section. We use these techniques to account for the width and length of corridors on landscapes of any spatial configuration.

3.3. Path Planning

Path planning is an area of study within the field of artificial intelligence. It has applications in areas such as robotic surgery (Kiraly et al. 2004), unmanned bomb disposal (Jian-Jun et al. 2007), video game artificial intelligence (Demyen 2007) and molecular motion (Cortés et al. 2005). The classic problem of path planning is called the "piano mover's problem" (LaValle 2006), where an agent (the piano) is to be placed in an enclosed area (a room) filled with obstacles such as furniture. One must determine an optimal path that could move the piano without colliding with the obstacles. The optimal path may be the shortest path, the widest path or the path with fewest turns. Geometric characteristics of the path such as width, length (Demyen 2007), turn angle (Pinter 2001), and steepness (Roles and ElAarag 2013) are explicitly controlled by partitioning the area into a set of convex polygons (e.g., triangles) called a *navigation mesh*. A search algorithm is then used to determine the optimal path. Here, we adapt a technique proposed by Demyen (2007) who used a triangular navigational mesh to find the maximum width of an agent that can travel collision-free from a starting to an ending point. We integrate Demyen's method with optimization to calculate the width and length of wildlife corridors.

4. Overview of the Optimal Corridor Construction Approach

We introduce the Optimal Corridor Construction Approach (OCCA) that allows analysts to create optimally connected reserves by explicitly controlling geometric characteristics on a landscape of any spatial configuration. We demonstrate the OCCA by controlling the width and length of a wildlife corridor. The concepts and the framework can be applied beyond wildlife corridors.

The approach involves five steps, as outlined in Table 1. First, the corridor objectives and constraints are to be defined (Step 1). For example, analysts may wish to find the minimum-length corridor that satisfies a minimum width requirement. Alternatively, they may want to maximize the width of the corridor while limiting the length. In Step 2, the building blocks of the corridor, called polygons, must be defined. Polygons consist of one or more contiguous parcels. For more detail, see Section 5.2.

In Step 3, the polygons are used to create a new graphical interpretation of the landscape (see Figure 3 for an example). Instead of defining parcels as nodes and adjacencies between parcels as edges, the nodes now represent transitions between polygons called *gates*, and edges represent optimal routes (for example, the route of maximal width or minimal length) through polygons from one gate to the next.

Table 1 Proposed approach.

The Optimal Corridor Construction Approach
1. Specify corridor objectives and constraints
2. Select eligible polygons for corridor construction
3. Find gate pairs for each triplet of non-overlapping contiguous polygons
4. For each triplet and each of its gate pairs, find the optimal route and associated width and length
5. Create and solve an optimization problem for optimal corridor construction

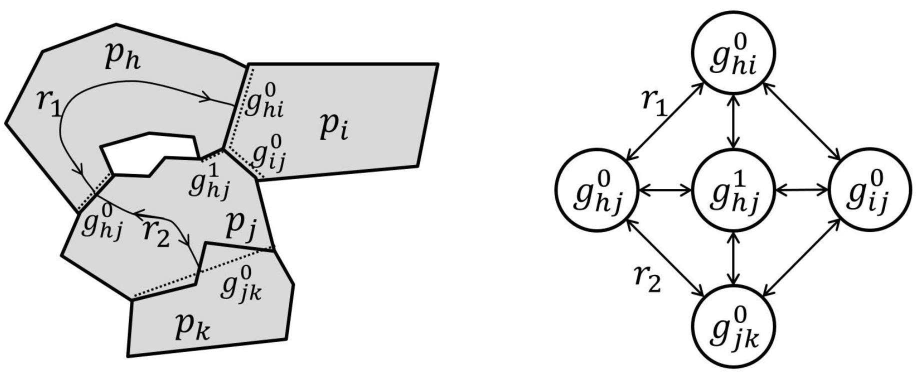

Figure 3 New graph-theoretical representation.

In Step 4, path planning techniques (Demyen 2007) are applied to calculate the geometric characteristics of the routes between gates to be used as edge weights in the graph (Fig. 3). This step involves solving a computationally trivial optimization problem for each triplet of polygons. Finally, Step 5 solves a network flow model on the new graph to generate the optimal corridor on the landscape. Each step of the approach is described in detail in Section 5.

The OCCA allows analysts to obtain a set of parcels from the landscape that form a corridor of maximum width subject to length restrictions, or of minimum length subject to minimum allowable width constraints. This novel integration of path planning techniques and mathematical programming allows for control over geometric aspects (such as path width, length, steepness, angle) that were previously beyond the capacity of spatial optimization.

5. Methodology of the Optimal Corridor Construction Approach

Terminology

For consistency, upper case letters denote sets, objective function values and bounds, while lower case letters denote decision variables, and objective- and constraint coefficients. We consider a landscape Ω that is composed of a set of polygons. A polygon may consist of a single parcel, or a cluster of contiguous parcels. Let $V = \{v_i\}$ be the set of *vertices* on the Cartesian plane, and an *edge* $e = (v_i, v_j)$ be a straight, closed line segment connecting two vertices. A *polygon* p is a continuous region enclosed by a set of at least three edges $\varepsilon(p) = \{e\}$ which we call the *boundary of* p. Every edge $e \in \varepsilon(p)$ shares each endpoint with exactly one other edge in $\varepsilon(p)$ and does not intersect any other edge in $\varepsilon(p)$. In this study, we consider *polygons with holes*. Let p_s be a polygon, and let $p_0, \ldots, p_k$ be a set of non-intersecting polygons within the boundary of p_s. Then, the polygon $p_* = p_s \setminus p_0 \setminus \ldots \setminus p_k$ is a polygon with holes, with boundary $\varepsilon(p_*) = \varepsilon(p_s) \cup \varepsilon(p_0) \cup \ldots \cup \varepsilon(p_k)$. Two polygons p_i and p_j are *non-overlapping* if $(p_i \setminus \varepsilon(p_i)) \cap (p_j \setminus \varepsilon(p_j)) = \emptyset$. Two polygons p_i and p_j are *adjacent* if they are non-overlapping and $\varepsilon(p_i) \cap \varepsilon(p_j) \neq \emptyset$. A vertex, edge or polygon a is *contained* in a polygon p if $a \cap p = a$. An edge or polygon b is *partially contained* in a polygon p if $b \cap p \neq b$, $b \cap p \neq p$ and $b \cap p \neq \emptyset$.

In this study we consider only *contiguous* landscapes, that is, landscapes such that $\Omega = \bigcup_{i=0}^{n} p_i$ is a polygon. Given a landscape Ω, a *corridor* $C = (p_0, p_1, \ldots, p_\omega)$ is a sequence of non-overlapping polygons in Ω where p_i, p_{i+1} are adjacent for all $i < \omega$, and polygons p_0, p_ω represent the pre-existing reserves or specific landscape features such as lakes or seashores that we wish to connect.

In order to give "width" and "length" formal definitions, we first introduce the concept of "agent". An *agent* is a circular region of a given diameter whose location is defined by its center-point. An agent moves through a corridor C by beginning in p_0, travelling on a continuous curve called a *path* through $p_1, p_2, \ldots, p_{\omega-1}$ without intersecting $\varepsilon(\bigcup_{p_i \in C} p_i) \setminus \{\varepsilon(p_0) \cap \varepsilon(p_1)\} \setminus \{\varepsilon(p_{\omega-1}) \cap \varepsilon(p_\omega)\}$, and ending in p_ω. Note that the possible number of paths through a corridor is infinite. The *width* of a path is the maximum diameter an agent can have and still follow the path. The *length* of a path is the distance the agent travels following the path.

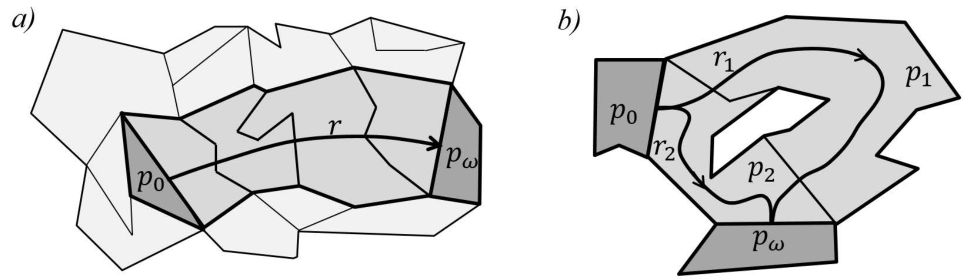

Figure 4 a) A landscape, a corridor and a route; b) a corridor with two routes.

In this study, we are interested in centered paths which we call *routes*. We assume the populations of interest are edge averse in that they typically travel through the interior of the corridor (Soule and Gilpin 1991). The route is a proxy for the average path that the population is likely to follow. Note that the route may not necessarily be the shortest path (Figure 4a).

Given a corridor, we calculate the width and length of each route in a corridor to determine whether it is usable, and if it is, whether it is optimal for wildlife. A corridor with holes has many potential routes. For example, in Figure 4b, the corridor has two routes. Depending on the needs of the populations of interest, the best route through a corridor may be different. In Figure 4b, r_1 is the optimal route if we want a route of maximum width, but if we wish to minimize length, the optimal route is r_2. For a given landscape, a corridor that contains the optimal route is called the *optimal corridor*.

Given a landscape Ω, and polygons p_0 and p_ω, our objective is to find an optimal corridor that connects these polygons containing a route of maximum width or shortest length subject to a budget constraint.

5.1. Specifying Corridor Objectives and Constraints

The first step of the Optimal Corridor Construction Approach is to specify the corridor objectives and constraints (see Table 1). Given a landscape partitioned into parcels available for corridor use, a wildlife corridor is to be created by selecting parcels that will connect two areas of habitat (e.g., Figure 1). Often, not only must corridors be connected, but they must also be comprised of suitable habitat, meet budget requirements, and contain routes that are neither too narrow nor too long.

An optimal corridor may be a corridor with the widest route or a corridor with the shortest route. If the analyst chooses to maximize corridor width, a constraint with an upper bound on length may be added. Alternatively, if the analyst minimizes corridor length, a minimum width threshold can be specified.

5.2. Selecting Eligible Polygons for Corridor Construction

Once we have determined the characteristics of an optimal corridor, we create a set of eligible polygons. Typically in corridor selection MIPs, the corridor is one parcel wide. These corridors may be suboptimal if width is of concern. We allow the corridor to be multiple parcels wide by defining *polygons*, sets of one or more contiguous parcels. These potentially overlapping polygons are analogous to clusters in area restriction models, introduced by McDill et al. (2002), and also used by Goycoolea et al. (2005), Könnyű and Tóth (2013), and Tóth et al. (2013). In these models, constraints ensure that overlapping and adjacent clusters cannot be selected, thus ensuring that contiguous areas selected for harvest do not exceed a predefined threshold. For wildlife corridors, using polygons rather than single parcels allows us to control geometric characteristics of corridors that are several parcels wide.

Theoretically, the number of possible combinations of contiguous parcels can be large and unwieldy. For large landscapes of thousands of parcels, there would be a combinatorial explosion in the number of potential polygons leading to computationally expensive, or even intractable, models. Thus, our approach requires that the set of *e*ligible polygons is restricted based on the landscape and/or the computational resources at hand. For example, the set of eligible polygons may only include contiguous clusters of parcels whose total area is less than 50 hectares.

5.3. Find Valid Gate Pairs for Each Triplet of Polygons

The width and length of a corridor depends on the width and length of the route through each polygon in the corridor. In order to determine the width and length of a route through a polygon, we determine how the route travels through the polygon. We use *gate pairs* to specify where a route

Table 2 Procedure for finding gates given adjacent polygons.

Gate Finding Procedure

Let p_i, p_j be adjacent polygons.

1. Let E_c be a set of contiguous edges in $\varepsilon(p_i) \cap \varepsilon(p_j)$ such that no other edge in $\varepsilon(p_i) \cap \varepsilon(p_j)$ is contiguous to the edges in E_c.
2. For a given E_c, let v_1, v_2 be the endpoints of the contiguous edges in E_c, and construct a pseudo-edge $\tilde{e}^c_{ij} = (v_1, v_2)$.
   (a) If $\tilde{e}^c_{ij}$ is contained in $p_i \cup p_j$, $\tilde{e}^c_{ij} \in G_{ij}$,
   (b) else, if $\tilde{e}^c_{ij}$ is partially contained in $p_i \cup p_j$, define $\{\tilde{e}^{c0}_{ij}, \tilde{e}^{c1}_{ij}, \ldots, \tilde{e}^{ct}_{ij}\}$ as the segments of $\tilde{e}^c_{ij}$ partitioned by $\varepsilon(p_i \cup p_j)$. Then $\tilde{e}^{c0}_{ij}, \tilde{e}^{c1}_{ij}, \ldots, \tilde{e}^{ct}_{ij} \in G_{ij}$ and all contiguous combinations of $\{\tilde{e}^{c0}_{ij}, \tilde{e}^{c1}_{ij}, \ldots, \tilde{e}^{ct}_{ij}\}$ are in G_{ij},
   (c) else, if $\tilde{e}^c_{ij} \cap (p_i \cup p_j) = \emptyset$, so $\tilde{e}^c_{ij}$ is not a gate.
3. Repeat steps 1 and 2 for all E_c.

enters and exits the polygon. Valid gate pairs also depend on the previous and subsequent polygons in the corridor. For example, in Figure 5a, the width and length of an optimal route through corridor (p_2, p_3, p_4) is different than that of the optimal route through corridor (p_1, p_3, p_4). We let a *triplet* be a sequence of three non-overlapping polygons (p_i, p_j, p_k) such that p_i and p_j are adjacent and p_j and p_k are adjacent. Note that p_i and p_k may be the same polygon (for example, in Figure 4b, route r_1 crosses triplet (p_2, p_1, p_2)). In Figure 5a, there are nine triplets with p_3 as the middle polygon: $(p_1, p_3, p_1), (p_1, p_3, p_2), (p_1, p_3, p_4), (p_2, p_3, p_1), (p_2, p_3, p_2), (p_2, p_3, p_4), (p_4, p_3, p_1), (p_4, p_3, p_2), (p_4, p_3, p_4)$. Note the width and length of corridor (p_i, p_j, p_k) is equal to the width and length of its reversed corridor (p_k, p_j, p_i).

Given a triplet (p_i, p_j, p_k), a valid gate pair consists of two gates, an entering gate representing the transition from p_i to p_j and an exiting gate, representing the transition from p_j to p_k. We define G_{ij} as the set of gates between p_i and p_j, where p_i and p_j are adjacent. Note that the set

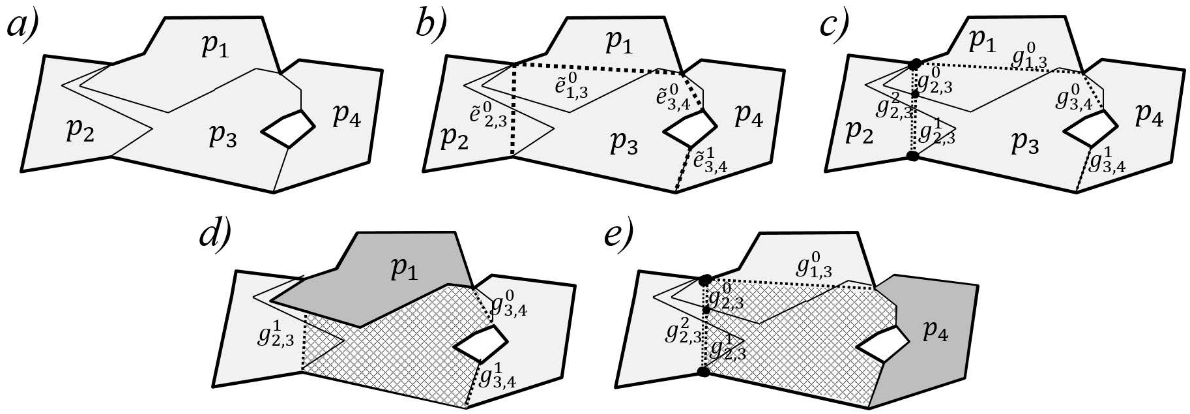

Figure 5 a) A set of polygons, b) their associated pseudo-edges, c) gates, d) gates and core polygon for (p_2, p_3, p_4) and e) gates and core polygon for (p_1, p_3, p_2).

G_{ij} has the same elements as the set G_{ji}. We generate gates between p_i and p_j by constructing *pseudo-edges* that represent where a route may enter p_j from p_i. The midpoint of a gate serves as a transition point for a route moving from polygon p_i to polygon p_j. To find G_{ij}, we introduce a gate finding procedure (see Table 2).

The gate finding procedure starts by identifying contiguous sets of shared edges between adjacent polygons p_i and p_j. For example, in Figure 5a, p_2 and p_3 share one set of contiguous edges and p_3 and p_4 share two sets of contiguous edges. For each set of contiguous edges, construct a pseudo-edge $\tilde{e}^c_{ij}$ that connects the endpoints of the contiguous edge set, where c indexes the contiguous edge sets. For example, in Figure 5b, the pseudo-edge between p_1 and p_3 is $\tilde{e}^0_{1,3}$, the pseudo-edge between p_2 and p_3 is $\tilde{e}^0_{2,3}$ and the pseudo-edges between p_3 and p_4 are $\tilde{e}^0_{3,4}$ and $\tilde{e}^1_{3,4}$.

If $\tilde{e}^c_{ij}$ is contained in $p_i \cup p_j$, then it is a gate. Since $\tilde{e}^0_{1,3}$ is contained in $p_1 \cup p_3$, it is a gate, denoted $g^0_{1,3}$, shown in Figure 5c. Similarly, $\tilde{e}^0_{3,4}$ and $\tilde{e}^1_{3,4}$ are gates, denoted by $g^0_{3,4}$ and $g^1_{3,4}$, respectively. If no part of $\tilde{e}^c_{ij}$ is contained in $p_i \cup p_j$, then it is not a gate. A route using such an edge as a gate may not be contained in the corridor.

If $\tilde{e}^c_{ij}$ is partially contained in $p_i \cup p_j$, we partition $\tilde{e}^c_{ij}$ at each point it crosses $\varepsilon(p_i \cup p_j)$. Each partition of $\tilde{e}^c_{ij}$ is a gate, as well as any contiguous combination of partitions. In Figure 5b, $\tilde{e}^0_{2,3}$ is partially contained in $p_2 \cup p_3$, so we partition it into two gates, $g^0_{2,3}, g^1_{2,3}$, and we also have the combination gate $g^2_{2,3} = g^0_{2,3} \cup g^1_{2,3}$. In Figure 5c, we have gate sets $G_{1,3} = \{g^0_{1,3}\}$, $G_{2,3} = \{g^0_{2,3}, g^1_{2,3}, g^2_{2,3}\}$ and $G_{3,4} = \{g^0_{3,4}, g^1_{3,4}\}$. Notice that $g^0_{2,3}$ is not contained in $p_2 \cup p_3$, however it is a possible gate for triplet (p_1, p_3, p_2). Note that gates do not have direction, so $g_{i,j} = g_{j,i}$.

For each triplet (p_i, p_j, p_k), we use its corresponding gate sets, G_{ij}, G_{jk} to create the set of gate pairs, Φ_{ijk}. Each gate pair (g^m_{ij}, g^n_{jk}) is comprised of one gate from G_{ij} that is contained in $p_i \cup p_j \cup p_k$ and one gate from G_{jk} that is also contained in $p_i \cup p_j \cup p_k$. In Figure 5d, the gate pairs for triplet (p_2, p_3, p_4) are $\Phi_{2,3,4} = \{(g^1_{2,3}, g^0_{3,4}), (g^1_{2,3}, g^1_{3,4})\}$ and in Figure 5e, the gate pairs for triplet (p_1, p_3, p_2) are $\Phi_{1,3,2} = \{(g^0_{1,3}, g^0_{3,2}), (g^0_{1,3}, g^1_{3,2}), (g^0_{1,3}, g^2_{3,2})\}$.

Next, for every gate pair, we find the optimal route through its core polygon by embedding pathfinding techniques into a network optimization model.

5.4. For Each Triplet and Each of its Valid Gate Pairs, Find the Optimal Route and Associated Width and Length

Given a triplet (p_i, p_j, p_k) and one of its gate pairs, (g^m_{ij}, g^n_{jk}), Step 4 of the OCCA is to find the optimal route from g^m_{ij} to g^n_{jk}. The optimal route from g^m_{ij} to g^n_{jk} must remain within $p_i \cup p_j \cup p_k$. We define the *core polygon*, $\tilde{p}_{ijk}$, as the polygon contained in $p_i \cup p_j \cup p_k$ where $\varepsilon(\tilde{p}_{ijk})$ includes all gates from all gate pairs in Φ_{ijk}. In Figure 5d, $\tilde{p}_{2,3,4}$ is indicated with crosshatching, as is $\tilde{p}_{1,3,2}$ in Figure 5e.

For each gate pair (g^m_{ij}, g^n_{jk}) and its core polygon $\tilde{p}_{ijk}$, we find the optimal route from the midpoint of gate g^m_{ij}, denoted $mid(g^m_{ij})$, to the midpoint of gate g^n_{jk}, $mid(g^n_{jk})$, through $\tilde{p}_{ijk}$. We proceed by using the following path planning technique (Demyen 2007) to determine the widths and lengths of route segments, which we then use to construct the optimal route via network optimization.

5.4.1. Triangulating $\tilde{p}_{ijk}$ Following Demyen (2007), we first use Constrained Delaunay Triangulation (Kallman et al. 2003) to create a navigational mesh by decomposing $\tilde{p}_{ijk}$ into triangles. A Delaunay Triangulation (see de Berg et al. 2008) is a triangulation in which no vertex in V lies inside the circumcircle of any other triangle. As an example, Figure 6a shows a set of vertices, and

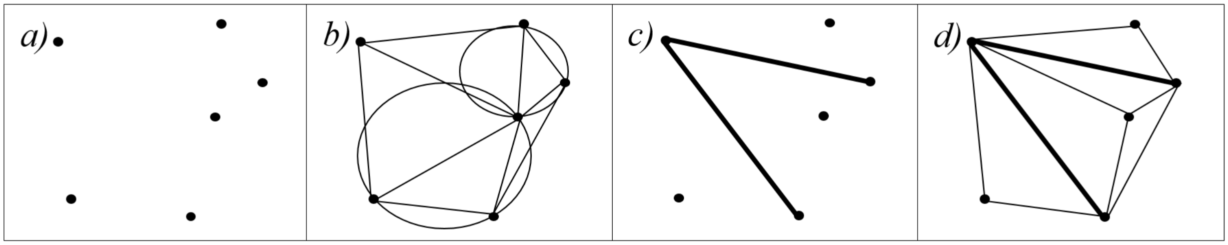

Figure 6 (a) A set of vertices V and (b) its Delaunay Triangulation. (c) A set of vertices V and edges E and (d) its Constrained Delaunay Triangulation.

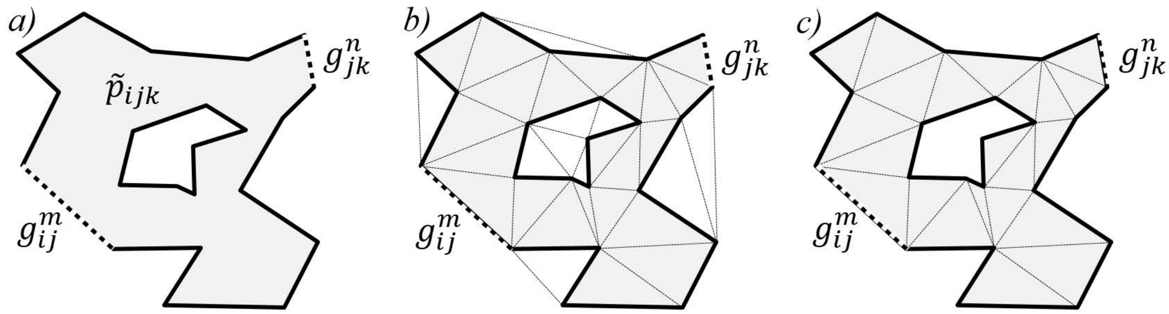

Figure 7 (a) Core polygon $\tilde{p}_{ijk}$, (b) CDT of $\tilde{p}_{ijk}$ and (c) $CDT'(\tilde{p}_{ijk})$.

Figure 6b shows its Delaunay Triangulation. A Constrained Delaunay Triangulation (CDT) is a Delaunay Triangulation that is formed with preexisting edges[1]. Given the vertices and edges in Figure 6c, the corresponding CDT is shown in Figure 6d. There are several algorithms for finding CDTs including Chew (1989) or Sloan (1993).

Given $\tilde{p}_{ijk}$ (Fig. 7a), consider the CDT of $\tilde{p}_{ijk}$ (Fig. 7b). Notice some edges of the CDT may not be contained in $\tilde{p}_{ijk}$; they either lay outside the outer boundary of $\tilde{p}_{ijk}$ or they are in a hole. Let $CDT'(\tilde{p}_{ijk})$ be the set of all edges of the CDT of $\tilde{p}_{ijk}$ that are internal to $\tilde{p}_{ijk}$ (Fig. 7c). We have now decomposed $\tilde{p}_{ijk}$ into triangles.

[1] Aside from a trivial case, the Constrained Delauney Triangulation produces a unique triangulation. A case where there exist multiple triangulations is when four vertices form a perfect square that does not contain any other vertex. In this example, there are two different triangulations that are valid CDTs. For more details, see Chapter 2 and Theorem 2.11 in Cheng et al. (2012).

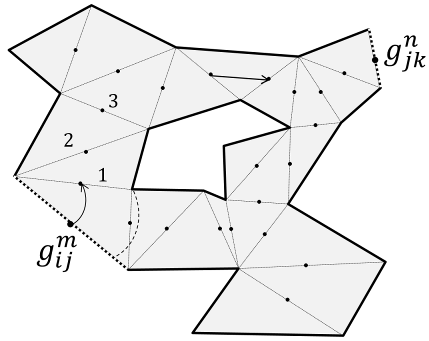

Figure 8 Triangle edge pair routes.

5.4.2. Calculating the Width and Length of Triangle Edge Pairs We define a *triangle edge pair* (a,b) as an ordered pair of edges from $CDT'(\tilde{p}_{ijk}) \cup \{g_{ij}^m, g_{jk}^n\}$ that share a vertex and $(mid(a), mid(b))$ does not intersect any other edge in $CDT'(\tilde{p}_{ijk}) \cup \{g_{ij}^m, g_{jk}^n\}$. The set of all triangle edge pairs associated with gate pair (g_{ij}^m, g_{jk}^n) is denoted as $T_{mn}(\tilde{p}_{ijk})$.

Given a valid gate pair (g_{ij}^m, g_{jk}^n) and its core polygon $\tilde{p}_{ijk}$, a route from $mid(g_{ij}^m)$ through $\tilde{p}_{ijk}$ to $mid(g_{jk}^n)$ can be represented by the sequence of triangle edge pairs it crosses: $(g_{ij}^m, 1), (1,2), \ldots, (s, g_{jk}^n)$ (see Figure 8 for example). The width and length of the route is determined by the width and length of each triangle it crosses, entering the triangle at $mid(a)$ and exiting the triangle at $mid(b)$.

We calculate the width of the route from a to b, denoted by ψ_{ab} using Demyen (2007). This algorithm determines the width using: 1) the angle created by the triangle edge pair, 2) the lengths of a and b, and 3) whether the third edge of the triangle is in the boundary of $\tilde{p}_{ijk}$. Based on these triangle edge pair attributes, the algorithm determines whether the maximal width route from a to b is curved (such as the route from g_{ij}^m to 1 in Figure 8), then calculates the narrowest width of the route.

For wildlife corridors, knowing the exact route length through a triangle edge pair is not necessary. The route represents the estimated preferred path, and there is no guarantee that wildlife will follow it exactly. When the route through the triangle edge pairs curves, calculating route length requires integration and may be computationally expensive. Since an exact length is not a priority, a quickly calculated proxy will suffice. We use the distance between the midpoints of a and b, $\delta_{ab} = \|mid(a) - mid(b)\|$, as a linear approximation of route length. If the information on precise wildlife movement is available, such as movement among watering holes on the African savannah, the model can easily be modified to account for such information by replacing our linear approximation.

5.4.3. Network Optimization for Optimal Route. Given a gate pair (g_{ij}^m, g_{jk}^n) and $\tilde{p}_{ijk}$, we have now identified each triangle edge pair $(a,b) \in T_{mn}(\tilde{p}_{ijk})$ with associated width ψ_{ab} and length δ_{ab}. A binary decision variable y_{ab} is defined for each triangle edge pair (a,b) to indicate if it is included in the optimal route through $\tilde{p}_{ijk}$. To find the widest route from $mid(g_{ij}^m)$ to $mid(g_{jk}^n)$ through $\tilde{p}_{ijk}$, with a maximum length threshold, L_{max}, we formulate a network optimization problem as (1) - (7).

$$\max Z \tag{1}$$

subject to:

$$Z - M \le (\psi_{ab} - M) y_{ab} \qquad \forall\, (a,b) \in T_{mn}(\tilde{p}_{ijk}) \tag{2}$$

$$\sum_{(a,b)\in T_{mn}(\tilde{p}_{ijk})} \delta_{ab} y_{ab} \le L_{max} \tag{3}$$

$$\sum_{(g_{ij}^m, b)\in T_{mn}(\tilde{p}_{ijk})} y_{mb} = 1 \tag{4}$$

$$\sum_{(a, g_{jk}^n)\in T_{mn}(\tilde{p}_{ijk})} y_{an} = 1 \tag{5}$$

$$\sum_{(a,b)\in T_{mn}(\tilde{p}_{ijk})} y_{ab} - \sum_{(b,c)\in T_{mn}(\tilde{p}_{ijk})} y_{bc} = 0 \qquad \forall\, b \in CDT'(\tilde{p}_{ijk}) \tag{6}$$

$$y_{ab} \in \{0,1\} \qquad \forall\, (a,b) \in T_{mn}(\tilde{p}_{ijk}) \tag{7}$$

The objective (1) is to maximize Z ($\ge$0), a number which cannot exceed the minimum edge pair width in the route, as enforced by constraint (2). Constraint (3) ensures the length of the optimal route does not exceed a limit, L_{max}. Constraint (4) requires that the route starts at g_{ij}^m, whereas Constraint (5) requires that the route ends at g_{jk}^n. Constraint (6) is the network flow preservation constraint. Together, these constraints ensure a connected route. Lastly, Constraint (7) requires y_{ab} to be binary. Note that this form is equivalent to a network flow problem, for which very efficient algorithms exist.

Alternatively, if the shortest route with a minimum width W_{min} is sought, the model takes the form of a shortest path problem:

$$\min \sum_{(a,b)\in T_{mn}(\tilde{p}_{ijk})} \delta_{ab} y_{ab} \tag{8}$$

subject to:

$$(\psi_{ab} - W_{min}) y_{ab} \geq 0 \qquad \forall\, (a,b) \in T_{mn}(\tilde{p}_{ijk}) \tag{9}$$

and Constraints (4) to (7)

The objective (8) is to minimize route length. Since we are finding corridors with nonzero widths, it is reasonable to assume that there is a minimum width requirement, W_{min}, which is enforced in Constraint (9). Alternatively, we can define y_{ab} only for $a,b : \psi_{ab} \geq W_{min}$. We use Constraints (4) through (6) to ensure a connected route. Even though the decision variables y_{ab} are constrained to be binary in Constraint (7), there are efficient algorithms for the shortest path problem.

After solving the appropriate network optimization problem, we can use the optimal solution y_{ab}^* to calculate the width for the optimal route between the gate pairs, $w_{ijk}^{mn} = \min_{a,b:y_{ab}^*=1} \psi_{ab}$ and the length is $\ell_{ijk}^{mn} = \sum_{a,b:y_{ab}^*=1} \delta_{ab} y_{ab}^*$.

To summarize, we start by formulating a graph representation of landscape Ω in which each node is the midpoint of a gate g_{ij}^m and each edge connects a valid gate pair, (g_{ij}^m, g_{jk}^n). For each gate pair, we determine the width and length of the optimal route from g_{ij}^m to g_{jk}^n through $\tilde{p}_{ijk}$ by finding triangle edge pairs $T_{mn}(\tilde{p}_{ijk})$. A network optimization model is then used to find the optimal route across the network. The widths and lengths of the gate pairs are assigned as the widths and lengths of the corresponding edge weights in the landscape graph. In the final step of the OCCA, we use this graph to construct the optimal corridor.

5.5. Formulate and Solve the Optimal Corridor Construction Problem

The final step in the Optimal Corridor Construction Approach (Step 5 in Table 1) is to use the gate pair route widths w_{ijk}^{mn} and lengths ℓ_{ijk}^{mn} to create and solve yet another network optimization problem; this time to find the optimal corridor.

The Optimal Corridor Construction Problem is analogous to the network optimization models introduced earlier for finding optimal routes between gate pairs. Rather than selecting triangles to traverse a polygon, this time polygons are selected to traverse the landscape. The resulting set of polygons forms a corridor containing optimal routes.

Let $x_{ijk}^{mn} \in \{0,1\}$ be the binary decision variable that indicates whether the route from gate g_{ij}^{m} to gate g_{jk}^{n} through polygon $\tilde{p}_{ijk}$ is included in the corridor and let the set of all gate pairs on the landscape Ω be Φ_Ω. Let p_0, p_ω denote the polygons we wish to connect. Again, these polygons could represent lake-, or seashores or other landscape features. Recall that the landscape was originally partitioned into parcels Q, which we used to create polygons. For each parcel q, let Λ_q be the set of polygons that contain q. Also let D be an upper bound on all widths w_{ijk}^{mn}. Let c_j be the cost associated with including polygon p_j in the corridor, and let B be the total budget. If our objective is to create a maximal width corridor with an upper bound on corridor length L_{max} and a budget constraint, the network optimization problem is as follows:

$$\max W \tag{10}$$

subject to:

$$W - D \leq (w_{ijk}^{mn} - D)x_{ijk}^{mn} \qquad \forall\ (g_{ij}^{m}, g_{jk}^{n}) \in \Phi_\Omega \tag{11}$$

$$\sum_{(g_{ij}^{m}, g_{jk}^{n}) \in \Phi_\Omega} \ell_{ijk}^{mn} x_{ijk}^{mn} \leq L_{max} \tag{12}$$

$$\sum_{(g_{0j}^{m}, g_{jk}^{n}) \in \Phi_\Omega} x_{0jk}^{mn} = 1 \tag{13}$$

$$\sum_{(g_{ij}^m, g_{j\omega}^n) \in \Phi_\Omega} x_{ij\omega}^{mn} = 1 \tag{14}$$

$$\sum_{(g_{ij}^m, g_{jk}^n) \in \Phi_\Omega} x_{ijk}^{mn} - \sum_{(g_{hi}^r, g_{ij}^m) \in \Phi_\Omega} x_{hij}^{rm} = 0 \qquad \forall\, g_{ij}^m \in \bigcup_{i \neq 0, j \neq \omega} G_{ij} \tag{15}$$

$$\sum_{(g_{ij}^m, g_{jk}^n): p_i \in \Lambda_q} x_{ijk}^{mn} \leq 1 \qquad \forall\, q \in Q \tag{16}$$

$$\sum_{(g_{ij}^m, g_{jk}^n): p_j \in \Lambda_q} x_{ijk}^{mn} \leq 1 \qquad \forall\, q \in Q \tag{17}$$

$$\sum_{(g_{ij}^m, g_{jk}^n): p_k \in \Lambda_q} x_{ijk}^{mn} \leq 1 \qquad \forall\, q \in Q \tag{18}$$

$$\sum_{(g_{ij}^m, g_{jk}^n): p_j \in \Omega} c_j * x_{ijk}^{mn} \leq B \tag{19}$$

$$x_{ijk}^{mn} \in \{0, 1\} \qquad \forall\, (g_{ij}^m, g_{jk}^n) \in \Phi_\Omega \tag{20}$$

As in the model that finds maximum-width routes between gate pairs, the objective (10) here is to maximize corridor width, W ($\geq$0), which is defined in Constraint (11). Corridor length is controlled by Constraint (12), while corridor connectivity is enforced by Constraints (13) through (15). Constraints (16)-(18) ensure that each parcel is in at most one polygon selected for the corridor. Lastly, Constraint (19) makes sure that the costs do not exceed the budget (B). If the shortest route with a minimum width threshold W_{min} is sought, the model becomes:

$$\min \sum_{(g_{ij}^m, g_{jk}^n) \in \Phi_\Omega} \ell_{ijk}^{mn} x_{ijk}^{mn} \tag{21}$$

subject to:

$$(w_{ijk}^{mn} - W_{min}) x_{ijk}^{mn} \geq 0 \qquad \forall\, (g_{ij}^m, g_{jk}^n) \in \Phi_\Omega \tag{22}$$

and Constraints (13) to (20)

Analogous to finding the minimum-length route between gate pairs, this model selects minimum length corridors by using an objective function (21) that minimizes the total length of the route. Constraint (22) ensures that the corridor satisfies the minimum width requirement. The rest of the constraints ensure that the corridor is connected. To simplify inequality (22), we can define x_{ijk}^{mn} only for gate pairs satisfying the property $w_{ijk} \geq W_{min}$. Due to the budget constraint (19) and Constraints (16) - (18), this optimization problem is equivalent to a shortest path resource-constrained problem, which is known to be NP-hard.

6. Illustrative Example: Eldorado Dataset

6.1. Methods

To demonstrate the use of the OCCA, we consider a region of the Eldorado National Forest in California (Fig. 9a). The test dataset was obtained from the Forest Management Optimization Site, a landscape data repository housed by the University of New Bruncwick (FMOS 2014). The landscape is assumed to comprise parcels that are both suitable wildlife habitat and available for sale to be included in a wildlife corridor. The objective is to purchase a subset of the parcels in such a way so that core habitat patches (represented by dark polygons) are connected with a corridor of maximal width. The purchase price of each parcel is assumed to be proportional to their size and the available budget B is sufficient for buying only 15% of the total land area.

Due to computational limitations, it is impractical to consider every contiguous set of parcels as potential polygons. For this study, we limited the polygon set to all of the 1,282 individual parcels, plus 527 polygons that comprised multiple parcels not exceeding a combined area of 20 hectares. The data included 5 polygons with holes. This resulted in a total of 1,814 polygons and 112,000 gate pairs. We used the optimization problem (1) through (7) to determine gate pair widths, and the model (10) through (20) to construct the corridor. To calculate corridor length, we introduced an accounting variable L_{tot} and added the following constraint:

$$\sum_{(g_{ij}^{m},g_{jk}^{n})\in\Phi_\Omega} \ell_{ijk}^{mn} x_{ijk}^{mn} \leq L_{tot} \tag{23}$$

We first solved the Optimal Corridor Construction Problem without maximum length threshold (Fig. 9(b)). Then, a maximum corridor length restriction of 40 km was used leading to a solution depicted in Fig. 9(c)).

6.2. Results

All of the experiments were run on a Dell Precision T7500 machine with Intel Xeon CPU, E5630@2.53Ghz (2 processors) with 4 GB of RAM and 64-bit Windows. All of the optimization problems were solved using CPLEX (2011) version 12.4. No special network optimization

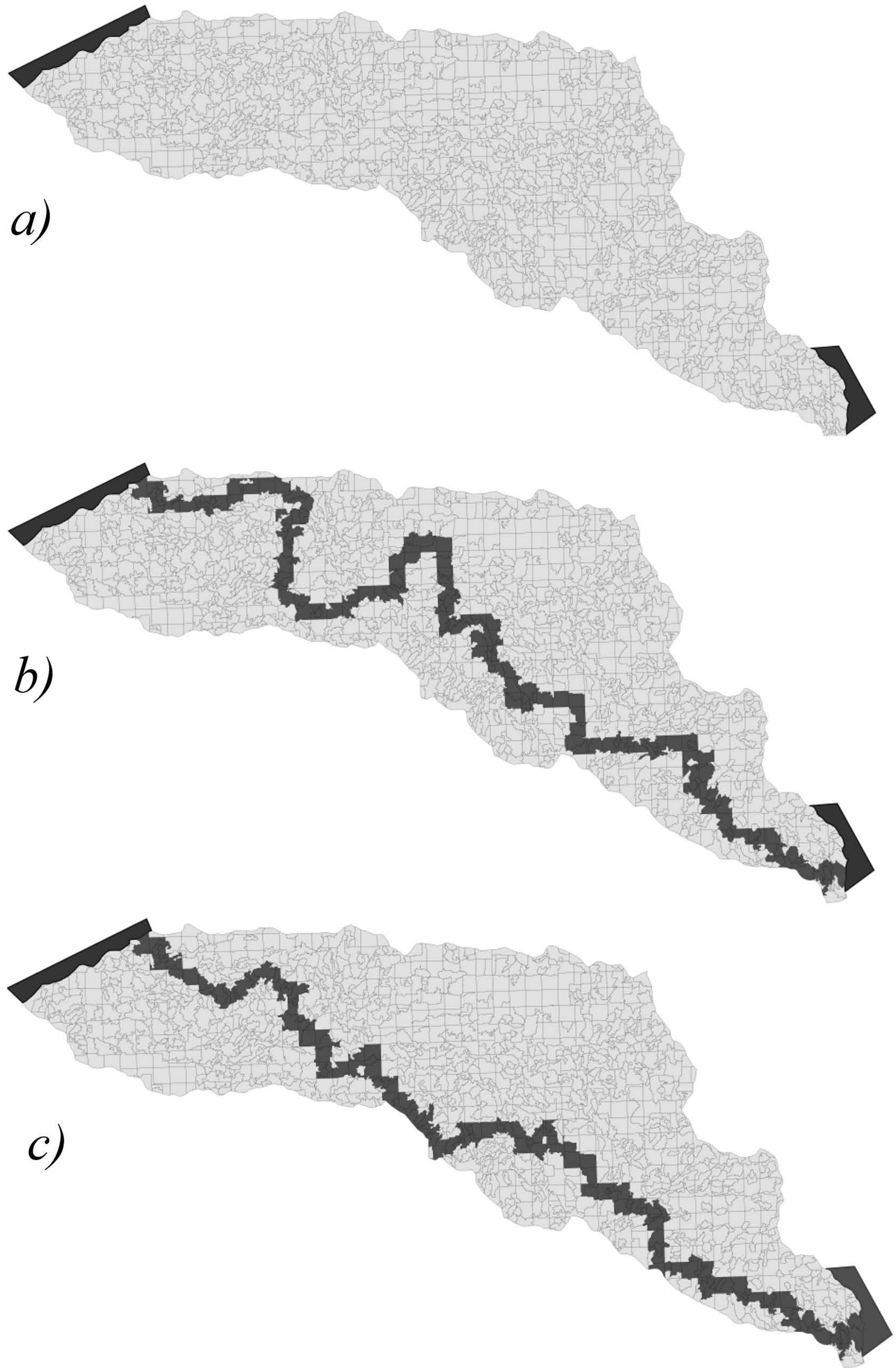

Figure 9 a) Eldorado landscape, b) maximal width corridor with no max length threshold and c) maximal width corridor with maximum length threshold of 40 km.

algorithms were used. Python 2.7 was used to generate gate pairs (Step 3) and the constraints for the route and corridor optimization problems (Steps 4 and 5). The code is available at https://www.faculty.washington.edu/toths/OCCA.zip.

The gate pair optimization problems (Step 4 of the OCCA) solved to optimality, taking less than 48 minutes for all 112,000 problems. With no maximum length requirement, the maximal width corridor optimization problem (Step 5) solved to optimality within 137 minutes and found the corridor shown in Figure 9(b), which is 309.1 meters wide and 48.0 km long. With the maximum length restriction, the corridor optimization problem solved to optimality in 12 minutes. The optimal corridor shown in Figure 9(c) is 299.1 m wide and 39.98 km long.

6.3. Discussion

Although there were over 100,000 gate pairs, all of the gate pair optimization problems solved to optimality very quickly due to their network model structure. The corridor optimization problem also proved to be computationally tractable. The length-restricted corridor problem could be solved within minutes, whereas the problem with no length restriction was solved in approximately three hours.

In our example, we used the OCCA to select a corridor of maximal width. The first corridor found maximized width, but was long and winding. When we included a restriction on corridor length, the resulting corridor was only slightly (3.2%) narrower, but was 8.02 km (16.7%) shorter than the corridor with no length restriction. Even though the length-restricted corridor is narrower, it may be a more appealing corridor in cases with low minimum width thresholds because of the reduction in length.

St John et al. (2016) embedded the OCCA into a forest harvest scheduling model to select reindeer corridors in Sweden. The case study consisted of 3,823 management units (1,996 forested and 1,827 non-forested), and yielded 4,461 polygons and 120,572 gate pairs. The gate pair optimization problems (Step 4 of the OCCA) solved to optimality, taking less than 124 minutes for all 120,572 problems. The full model for harvest scheduling with reindeer corridors was too large to find even a feasible solution within a reasonable amount of processing time. Therefore, a procedure was devised for finding an initial feasible solution to use as a MIP start. The MIP start allowed CPLEX to find a reasonable solution (with a 3.42% optimality gap) within 30 hours. For more details, see St John et al. (2016). The case study showed that (1) the OCCA can select corridors of maximum width or minimum length, and (2) it can be incorporated as a new functionality in large harvest scheduling models.

7. Computational Experiments with Synthetic Landscapes

Application of mixed integer programs to large datasets, such as those of realistic landscapes, often results in computational issues. In particular, the number of units (parcels), average (vertex) degree or valency (the average number of adjacent parcels), and variation in unit area (parcel size) have been shown to affect computational performance of spatially explicit forest harvest scheduling models cast as MIPs (McDill and Braze 2000, Constantino et al. 2008, Tóth et al. 2012, Passolt et al. 2013). In order to test our approach, we created hypothetical landscapes by varying the values of these three spatial attributes using a Vorronoi Tesselation-based landscape generator called rlandscape by Passolt et al. (2013).

7.1. Experimental Design

Table 3 provides three levels for each characteristic to be used in generating landscapes, based on seven realistic landscapes in FMOS 2014 (St John et al. 2016). Each value has a small margin around it for ease of landscape generation. The number of units in each case (200, 400 and 600 units) was smaller than that of an average realistic landscape (1,557 units). This specification ensured problem tractability, but is still within the range of the seven landscapes (71 to 5,881 units). Vertex degree ranged between 3.8 and 5.3, with an average of 4.3, while variation in unit size was between 62 and 139, with an average of 100. In this experiment, five landscapes for each combination of factor levels were generated, resulting in $3^3 * 5 = 135$ landscapes.

For each landscape, OCCA is run to determine the widest corridor connecting the bottom-left-most unit to the top-right-most unit. For each landscape, the number of gate pairs and the total run

Factor	Level 1	Level 2	Level 3
Number of Units	200 ± 1	400 ± 1	600 ± 1
Degree	3.9 ± 0.1	4.8 ± 0.1	5.7 ± 0.1
Area Variation	56.0 ± 1.0	98.0 ± 1.0	140.0 ± 1.0

Table 3 Landscape factors and levels for the computational experiment.

time (in CPU seconds) are recorded. The total run time includes the time to determine multiunit polygons and gate pairs, to formulate and solve gate pair optimization problems, and to formulate and solve the final corridor optimization problem.

7.2. Experimental Results

For each landscape factor, the distribution of the number of gate pairs is shown in Figure 10, and the distribution of the total run time is shown in Figure 11. Each of the factors appears to be positively correlated with total run time. The p-values for the full factorial ANOVA are shown in Table 4. Based on the ANOVA, the number of gate pairs is strongly related to computational time with a p-value less than $2.0 * 10^{-16}$.

Figure 12 illustrates the relationship between the number of gate pairs and total run time aggregated across all factors. Since the number of gate pairs appears to have a positive linear correlation with computation time, we fit a linear model to the data. Figure 12 includes the fitted line, $\hat{y} = -237.4 + 0.03789\hat{x}$, with adjusted $R^2 = 0.9677$, and Figure 13 shows the resulting residual and Q-Q plots.

7.3. Discussion

The ANOVA results indicate that number of units, vertex degree and variation of area are all positively correlated with total run time (4). The interaction of vertex degree and variation in area also impacts total run time, whereas the other pairwise interactions (number of units with vertex degree and number of units with variation in unit area) and the three-way interaction of all factors are not significant.

Factor(s)	p-value
num-units	0.0512
deg	0.0522
area-var	0.0136
num-units:deg	0.9731
num-units:area-var	0.6771
deg:area-var	0.0383
num-units:deg:area-var	0.9316

Table 4 ANOVA p-values for the computational experiment.

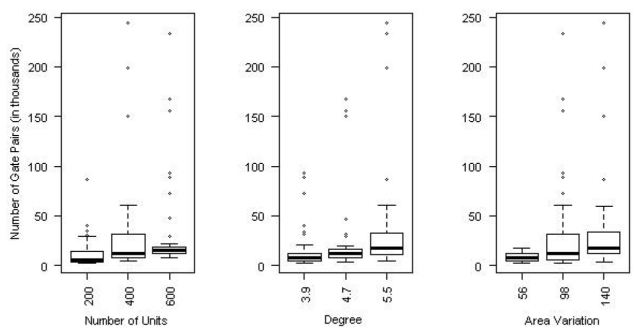

Figure 10 Distribution of number of gate pairs for each factor.

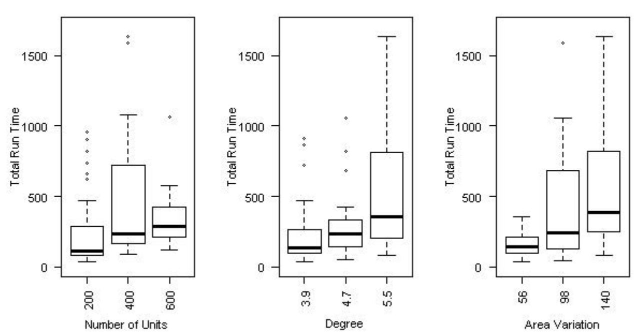

Figure 11 Distribution of total run time for each factor.

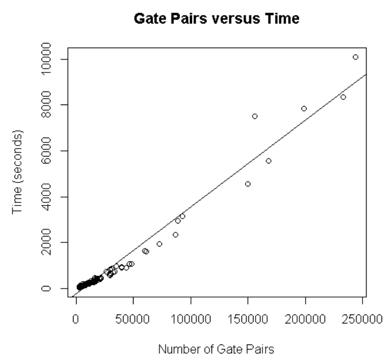

Figure 12 Number of gate pairs versus total run time.

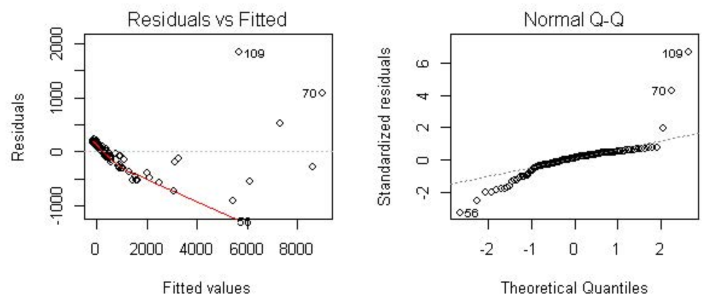

Figure 13 Residuals and Q-Q plots for the linear fit of number of gate pairs versus total run time.

The number of gate pairs in a problem tends to increase as each landscape factor increases (see Figure 10). The relationship between the number of gate pairs and total run time appears to be roughly linear (Figure 12), particularly for smaller problems (less than 100,000 gate pairs). Knowledge of this relationship can help users predict whether a problem may take a long time to solve, or even be intractable, based on the number of gate pairs.

In sum, the three landscape factors are statistically significant for computational performance. In3 addition, it appears that run time is linearly related to the number of gate pairs, which is a very promising computational result.

8. Conclusions

There are many extensions and variations of the Optimal Corridor Construction Approach to explore. In this paper, we introduced the OCCA for controlling width and length of wildlife corridors. However, the OCCA has the potential to control other geometric characteristics as well. For instance, if analysts want direct control over how straight the corridor must be, a metric for angle for each gate pairs can be calculated via the gate pair algorithm and included in the model. If a narrow or long path is desired, rather than a short or wide path, the objectives can be redefined accordingly.

For ease of presentation, we introduced the OCCA by constructing a corridor that connects two predetermined polygons. Rather than using the traditional graph interpretation of the landscape (see Figure 2), our new graph (Figure 3) can be implemented with other connectivity models as well, such as Jafari and Hearne (2013) or Conrad et al. (2012). In maximal covering problems, our approach can be used to create constraints and thus a solution where corridor width and length requirements must be met.

The OCCA can also be used in applications beyond wildlife corridors. In landscape management problems such as creating firebreaks for wildfires (Davis 1965) and emergency evacuation routes (Cova and Johnson 2003), connectivity of selected areas is required and controlling the geometric aspects of the connected area is often important.

As a last caveat, computational tractability can be a concern when OCCA is applied to very large landscapes, as mentioned in Section 5.2, and demonstrated in Section 7.2. To improve tractability, the number of eligible polygons used for corridor construction may be restricted at a cost to optimality. For example, in Section 6, if we increased the area restriction from 20 ha to 30 ha, the number of gate pairs would explode from 112,000 to over 5.1 million. Careful polygon selection is clearly critical to maintaining some degree of computational tractability. Further study is necessary on how to choose eligible polygons in order to reduce losses in optimality.

References

Alig, R., A. Platinga, S. Ahn and J. Kline. 2003. Land use changes involving forestry in the United States: 1952 to 1997, with projections to 2050. USDA Forest Service General Technical Report. PNW-GTR-587. USDA Forest Service, Pacific Northwest Research Station, Portland, OR. 92p.

Álvarez-Miranda, E., I. Ljubić and P. Mutzel. 2013. The Rooted Maximum Node-Weight Connected Subgraph Problem. In: Gomes C., Sellmann M. (eds) Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. CPAIOR 2013. Lecture Notes in Computer Science, vol 7874. pp. 300-315. Springer, Berlin, Heidelberg

Anderson, S. and K. Mann, Jr. H. Shugart. 1977. The effect of transmission-line corridors on bird populations. *Am. Midl. Nat.* 97(1) 216-221.

Beier, P. 1993. Determining minimum habitat areas and habitat corridors for cougars. *Conserv. Biol.* 7(1) 94-108.

Beier, P. and S. Loe. 1992. In my experience: a checklist for evaluating impacts to wildlife movement corridors. *Wildlife Soc. B.* 20(4) 434-440.

Beier, P. and R. Noss. 1998. Do habitat corridors provide connectivity? *Conserv. Biol.* 12(6) 1241-1252.

Bennett, A., K. Henein and G. Merriam. 1994. Corridor use and the elements of corridor quality: chipmunks and fencerows in a farmland mosaic. *Biol. Conserv.* 68 155-165.

Billionnet, A. 2013. Mathematical optimization ideas for biodiversity conservation. *Eur. J. Oper. Res.* 231 514-534.

Brás R., J. Cerdeira, D. Alagador and M. Araújo. 2013. Linking habitats for multiple species. *Environ. Modell. Softw.* 40 336-339.

Brodie, J., A. Giordano, B. Dickson, M. Hebblewhite, H. Bernard, J. Mohd-Azlan, J. Anderson and L. Ambu. 2015. Evaluating multispecies landscape connectivity in a threatened tropical mammal community. *Conserv. Biol.* 29 122-132.

Carvajal, R., M. Constantino, J.P. Vielma and A. Weintraub. 2013. Imposing connectivity constraints in forest planning models. *Oper. Res.* 61(4) 824-836.

Cerdeira, J., K. Gaston and L. Pinto. 2005. Connectivity in prioirity area selection for conservation. *Environ. Model. Assess.* 10(3) 183-192.

Cheng, S., T. Dey and J. Shewchuk. 2012. *Delaunay Mesh Generation.* CRC Press. Boca Raton, Florida, December 2012.

Chew, L. 1989. Constrained Delaunay Triangulations. *Algorithmica* 4 97-108.

CPLEX, IBM ILOG 2011. http://www.ibm.com/software/integration/optimization/cplex/.

Conrad, J., C. Gomes, W. van Hoeve, A. Sabharwal and J. Suter. 2012. Wildlife corridors as a connected subgraph problem. *J. Environ. Econ. Manag.* 63 1-18.

Constantino, M., I. Martins and J.G. Borges. 2008. A new mixed-integer programming model for harvest scheduling subject to maximum area restrictions. *Oper. Res.* 56 542-551.

Cortés, J., T. Siméon, V.R. de Angulo, D. Guieysse, M. Remaud-Simon and V. Tran. 2005. A path planning approach for computing large-amplitude motions of flexible molecules. *Bioinformatics* 21 116-125.

Coster, S.S., J.S. Veysey Powell J and K.J. Babbitt. 2014. Characterizing the width of amphibians movements during postbreeding migration. *Conserv. Biol.* 28(3) 756-762.

Cova, T. and J. Johnson. 2003. A network flow model for lane-based evacuation routing. *Transport. Res. A-Pol.* 37(7) 579-604.

Davis, L. 1965. The economics of wildfire protection with emphasis on fuel break systems. State of California Department of Conservation Division of Forestry.

Demyen, D. 2007. Efficient triangulation-based pathfinding. Thesis, Department of Computing Science, University of Alberta.

de Berg, M., O. Cheong, M. van Kreveld and M. Overmars. 2008. *Computational Geometry: Algorithms and Applications, 3rd ed.* (Springer Berlin Heidelberg).

Dilkina, B. and C.P. Gomes. 2010. Solving connected subgraph problems in wildlife conservation. Lodi A., M. Milano and P. Toth, eds. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* . Lecture Notes in Computer Science, Vol. 6140 (Springer Berlin), 102-116.

European Green Belt. 2015. http://www.europeangreenbelt.org. Retrieved June 1.

FMOS- Forest Management Optimization Site. 2014. Forest Management Optimization. Accessed February 12, 2014, http://ifmlab.for.unb.ca/fmos/.

Goycoolea, M., A.T. Murray, F. Barahona, R. Epstein and A. Weintraub. 2005. Harvest scheduling subject to maximum area restrictions: exploring exact approaches. *Oper. Res.* 53(3) 490-500.

Harrison, R. 1992. Toward a theory of inter-refuge corridor design. *Conserv. Biol.* 6(2) 293-295.

Jafari, N. and J. Hearne. 2013. A new method to solve the fully connected Reserve Network Design Problem. *Eur. J. Oper. Res.* 231 202-209.

Jian-Jun, Z., Y. Ru-Qing, Z. Wei-Jun, W. Xin-Hua and Q. Jun. 2007. Research on semi-automatic bomb fetching for an EOD robot. *Int. J. Adv. Robot. Syst.* 4(2) 247-252.

Kallmann, M., H. Bieri and D. Thalmann. 2003. Fully dynamic Constrained Delaunay Triangulations. In *Geometric Modeling for Scientific Visualization*, 241-257. Springer Berlin Heidelberg.

Kiraly, A., J. Helferty, E. Hoffman, G. McLennan and W. Higgins. 2004. Three-dimensional path planning for virtual bronchoscopy. *IEEE T. Med. Imaging*, 23(11) 1365-1379.

Könnyű, N. and S.F. Tóth. 2013. A cutting plane method for solving harvest scheduling models with area restrictions. *Eur. J. Oper. Res.* 228(1) 236-248.

LaValle, S. 2006. *Planning Algorithms.* Cambridge University Press, Cambridge, Massachusetts.

Le Bras, R., B. Dilkina, Y. Xue, C.P. Gomes, K.S. McKelvey, M.K. Schwartz and C.A. Montgomery. 2013. Robust Network Design for Multispecies Conservation. *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013*, 1305-1312.

McDill, M.E. and J. Braze. 2000. Comparing adjacency constraint formulations for randomly generated forest planning problems with four age-class distributions. *For. Sci.* 46 423-436.

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

Önal H. and R. Briers. 2005. Designing a conservation reserve network with minimal fragmentation: a linear integer programming approach. *Environ. Model. Assess.* 10 193-202.

Önal, H. and R. Briers. 2006. Optimal selection of a connected reserve network. *Oper. Res.* 54(2) 379-388.

Önal, H. and Y. Wang. 2008. A graph theory approach for designing conservation reserve networks with minimal fragmentation. *Networks* 51 142-152.

Panthera.org. 2015. Panthera Jaguar Corridor Initiative. Accessed June 1, 2015, http://www.panthera.org/programs/jaguar/jaguar-corridor-initiative

Passolt, G., M. Fix and S.F. Tóth. 2013. A Voronoi tessellation based approach to generate hypothetical forest landscapes. *Can. J. Forest Res.* 43(1) 78-89.

Pinter, M. 2001. Toward more realistic pathfinding. *Game Developer Magazine* 8(4).

Proctor, M., C. Servgeen, W. Kasworm and T. Radandt. 2008. Grizzly bear linkage enhancement plan for the Highway 3 corridor in the south Purcell Mountains of British Columbia. *Birchdale Ecol, Ltd, Kaslo, BC, Canada.*

Roles, J.A. and H. ElAarag. 2013. A smoothest path algorithm and its visualization tool. *Southeastcon, 2013 Proceedings of IEEE* pp. 1-6. IEEE.

Sessions, J. 1992. Solving for habitat connections as a Steiner network problem. *For. Sci.* 38(1) 203-207.

Shirabe, T. 2005. A model of contiguity for spatial unit allocation. *Geogr. Anal.* 37(1) 2-16.

Sloan, S. 1993. A fast algorithm for generating constrained Delaunay triangulations. *Comput. Struct.* 47(3) 441-450.

Soule, M. and M. Gilpin. 1991. The theory of wildlife corridor capability. *Nature Conservation 2: The Role of Corridors*, eds. D.A. Saunders and R.J. Hobbs (Surrey Beatty & Sons, New South Wales, 1991), pp3-8.

St John, R., K. Öhman, S.F. Tóth, P. Sandstróm, A. Korosuo and L. Eriksson. 2016. Combining spatiotemporal corridor design for reindeer migration with harvest scheduling in Northern Sweden. *Scand. J. For. Res.* 31(7) 655-663.

Tóth, S.F., R. Haight, S. Snyder, S. George, J. Miller, M. Gregory and A. Skibbe. 2009. Reserve selection with minimum contiguous area restrictions: an application to open space protection planning in suburban Chicago. *Biol. Conserv.* 142 1617-1627.

Tóth, S.F., M.E. McDill, N. Könnyű, S. George. 2012. A strengthening procedure for the path formulation of the area-based adjacency problem in harvest scheduling models. *Math. Comput. For. Nat.-Resour. Sci.* 4 1-23.

Tóth, S.F., M.E. McDill, N. Könnyű and S. George. 2013. Testing the use of lazy constraints in solving area-based adjacency formulations of harvest scheduling models. *Forest Sci.* 59(2) 157-176.

Williams, J. 1998. Delineating protected wildlife corridors with multi-objective programming. *Environ. Model. Assess.* 3(1-2) 77-86.

Williams, J. 2002. A zero-one programming model for contiguous land acquisition. *Geogr. Anal.* 34(4) 330-349.

Williams, J., C. ReVelle and S. Levin. 2005. Spatial attributes and reserve design models: a review. *Environ. Model. Assess.* 10(3) 163-181.

World Land Trust. 2015. Wildlife Corridors in India. Accessed June 1, 2015, http://www.worldlandtrust.org/projects/india/wildlife-corridors