geographical analysis

Geographical Analysis (2016) **48,** 176–190

Spatial Filtering for Identifying a Shortest Path Around Obstacles

Insu Hong[1], Alan T. Murray[2], Levi J. Wolf[3]

[1]Department of Geology and Geography, West Virginia University, Morgantown, WV, USA, [2]Center for Spatial Analytics and Geocomputation, College of Computing and Informatics, Drexel University, Philadelphia, PA, USA, [3]GeoDa Center for Geospatial Analysis and Computation, School of Geographical Sciences and Urban Planning, Arizona State University, Tempe, AZ, USA

The shortest path between two locations is crucial for location modeling, spatial analysis, and wayfinding in complex environments. When no transportation system or network exists, continuous space movement adds substantial complexity to identifying a best path as there are increased travel options as well as barriers inhibiting potential movement. To derive the shortest path, various methods have been developed. Recent work has attempted to exploit spatial knowledge and geographic information system functionality, representing significant advantages over existing methods. However, a high density of obstacles increases computational complexity making real-time solution difficult in some situations. This article presents a spatial filtering method to enhance Euclidean shortest path derivation in complex environments. The new approach offers substantial computational improvement while still guaranteeing an optimal path is found. Application results demonstrate the effectiveness of the approach and its comparative superiority.

Introduction

The shortest path between two points in continuous space around obstacles is fundamentally important for route planning, spatial analysis, and location modeling. It can be used for measuring proximity and distance, for deriving service areas and for studying behavioral movement (Hong and Murray 2013a,b). The continuous space shortest path that avoids line or polygonal obstacles has been referred as the Euclidean shortest path (ESP) (Guibas and Hershberger 1989; Hershberger and Suri 1999; Mitchell 2000). It is a well-recognized problem that continues to be of interest due to its practical relevance. Several solution techniques have been developed to solve the ESP, including visibility graph (Lozano-Pérez and Wesley 1979), local visibility graph (Zhang et al. 2005), shortest path map (Mitchell 2000), and Voronoi diagram (Papadopoulou and Lee 1995) to name a few. Solution approaches for the ESP focus on generating a graph through which a path can be formed. As the ESP is a continuous space problem, there is no predefined network. However, research has demonstrated that a network may be

Correspondence: Insu Hong, Department of Geology and Geography, West Virginia University, Morgantown, WV 26506, USA
e-mail: insu.hong@mail.wvu.edu

Submitted: March 26, 2015. Revised version accepted: July 14, 2015.

doi: 10.1111/gean.12086

derived that contains the optimal shortest path (Wangdahl, Pollock, and Woodward 1974; Lozano-Pérez and Wesley 1979; Viegas and Hansen 1985; de Berg, Cheong, and Van Kreveld 2008), allowing for the transformation of a continuous space problem to one where travel occurs on a discrete network. While an important advance, this network based transformation for the ESP has limited capability for real-time shortest path identification due to fundamental limitations in efficient graph construction. An issue is that current approaches must consider all or most obstacle and boundary vertices in a given region, regardless of the location of the origin and destination. Although substantial attention has been devoted to enhance efficiency and filter out unnecessary obstacles, real-time assessment and route planning is generally not possible with these approaches.

To address computational limitations, Hong and Murray (Hong and Murray 2013a,b) proposed the convexpath algorithm to efficiently solve the ESP problem. The algorithm is based on derived spatial knowledge and geographic information system (GIS) functionality to efficiently generate a graph proven to contain the optimal ESP. This approach is generally efficient, requiring minimal computing time. An exception, however, is the case when obstacle density is high, as the resulting graph tends to consider obstacles that have no impact on the ESP. While still an improvement compared to alternatives like the visibility graph, near real-time response and application may be limited when the number of obstacles is relatively high.

A spatial filtering technique is developed in this article to achieve greater efficiency for real-time path solution across continuous space with high obstacle density. The derived graph size and computing requirements are dramatically reduced as a result of this filtering approach yet optimality conditions are maintained. This article is structured as follows. An overview of the problem and literature is given. The convexpath approach and important spatial operations are described. A new solution approach based on spatial filtering is then detailed. This is followed by empirical assessment to identify an ESP using a number of routing and planning settings. Finally, concluding comments are provided.

Background

The shortest path between two points reflects important information about behavior and movement but it is also a measure of proximity and distance. It is heavily relied in spatial analysis and is central to most metrics, test, and models (see Bailey and Gatrell 1995; Fotheringham, Brunsdon, and Charlton 2000; de Smith, Goodchild, and Longley 2007; O'Sullivan and Unwin 2010; Rogerson 2010). The most efficient or shortest path is, therefore, relied on as a representative pattern of movement when actual behavior is not known, but also may be used in a predictive or prescriptive manner. Paths of travel (and distance) must take into account geographic obstacles to movement lest they be biased, inaccurate, or wrong (Klamroth 2001; Bischoff and Klamroth 2007; Carling, Han, and Håkansson 2012). Research has attempted to appropriately account for obstacles to eliminate potential errors or biases in analysis (Katz and Cooper 1981; Larson and Sadiq 1983; Batta, Ghose, and Palekar 1989; Aneja and Parlar 1994; Klamroth 2001), but challenges remain.

The ESP through complex environments is an essential part of navigation. Lozano-Pérez and Wesley (1979) suggested a method to derive a collision-free shortest path using a visibility graph. Other approaches, such as pheromone signals (Szymanski et al. 2006), slime mold strategy (Schmickl and Crailsheim 2007), and the MAKLINK approach (Habib and Asama 1991), have also been developed to support robotic wayfinding. Several approaches have been detailed for trans-oceanic shipping, to estimate arrival times based on a shortest path avoiding various obstacles impeding a route. Approaches to address this include the visibility graph (Fagerholt, Heimdal, and Loktu 2000), Voronoi diagrams (Bhattacharya and Gavrilova 2007), buffers and minimum bounding rectangles (Tsou 2010), and tessellation of the area using a square grid (Chang, Jan, and I. Parberry 2003; Bekker and Schmid 2006; Zhang et al. 2011).

Most solution methods for the ESP rely on the visibility graph. The visibility graph was proposed by Lozano-Pérez and Wesley (Lozano-Pérez and Wesley 1979) and subsequent research has suggested alternative graph generation techniques (Asano 1985; Welzl 1985; Asano et al. 1986; Rohnert 1986; Ghosh and Mount 1991; Pocchiola and Vegter 1996). The visibility graph attempts to connect vertices of obstacles, regional boundary vertices, and origin and destination vertices. If two vertices are mutually visible (unobstructed), the vertices are connected by an arc and they are included in the resulting graph. It has been proven that the visibility graph includes the shortest path that avoids obstacles between any two points in the graph (Viegas and Hansen 1985; de Berg et al. 2008). While several improvements have been suggested in the literature (Welzl 1985; Ghosh and Mount 1991; Pocchiola and Vegter 1996), the efficiency of the visibility graph approach is limited for practical and/or real-time usage. The reason for this is that it must evaluate all or most vertex pair combinations in the region (Hong and Murray 2013a). To make the visibility graph approach more efficient, filtering techniques have been proposed. The local visibility graph, as an example, utilizes proximity-based filters to reduce the number of obstacles evaluated (Kim et al. 2004; Zhang et al. 2005; Gao et al. 2011; Li, Gao, and Lu 2011). Unfortunately, the local visibility graph approaches remain computationally prohibitive and may omit the optimal ESP for an origin and destination pair (Hong and Murray 2013a).

Derivation of a graph has proven to be a necessary approach to solve the ESP in support of continuous space movement. While much progress has been made to more efficiently generate a graph in various ways, real-time navigation and wayfinding remain an elusive goal for ESP application, and continues to be an important area of research. Proposed in this article is an important step in this direction. An approach to substantially enhance efficiency in graph derivation is detailed that enables only necessary portions of a study region to be considered based on spatial knowledge and GIS functionality.

Convexpath

As noted above, a common approach for solving the ESP is the visibility graph. However, it is computationally intensive, limiting its usefulness for real-time navigation, and wayfinding as well as making it ineffective in big data environments. To address these limitations, Hong and Murray (2013a) proposed an algorithm, referred to as *convexpath*, to derive a graph containing the ESP. For simplicity, only polygonal obstacles are assumed, while line obstacles can be processed without any limitation. The convexpath algorithm is more efficient than the visibility graph because it intelligently exploits spatial knowledge and GIS functionality. The notion of a convex hull is utilized in graph construction, effectively eliminating unnecessary vertices and edges from being evaluated and included in the graph when they are not part of the ESP.

Consider the following definition (de Berg et al. 2008; Hong and Murray 2013a):

Definition (convex hull):
Given a set S in R^n, the *convex hull* of S, denoted CH(S), is the collection of all convex combinations between elements in S.

A convex hull, minimum bounding geometry has a property that for any two points v_1 and v_2 that lie on the boundary, $v_1, v_2 \subset \mathrm{CH}(S)$ and $v_1, v_2 \cap \mathrm{int}(CH(S)) = \emptyset$ where int(.) is the function identifying the interior of a set, -$v_1v_2 \in \mathrm{CH}(S)$, the line segment -$v_1v_2$ is completely contained in the hull and no portion of the line segment extends beyond or outside of the hull.

A property attributable to the convex hull as used to derive the ESP is the following theorem.

Theorem 1. *Given v_o and v_d, origin and destination locations, respectively, with straight line travel impeded by object k, the ESP lies on $CH(v_o, v_d, k)$.*

Proof: Hong and Murray (2013a) detail a proof of this theorem. However, the minimal perimeter property discussed above is sufficient to prove the result. Given that the convex hull has the minimal perimeter enclosing the vertices, origin, and destination, paths on its boundary dominate all other paths exterior to the hull.■

Also found in Hong and Murray (2013a) is an approach to derive the ESP for the case of multiple obstacles based on an iterative application of convex hulls to resolve any case where a hull boundary segment is impeded by obstacles in the region as well as the proof of optimality.

Hong and Murray (2013b) detailed an extension to account for a situation where an origin and/or destination vertex resides in the interior of any resulting convex hull. Given v_o, v_d, and k, then this is the case of $v_o \cap \mathrm{int}(CH(v_o, v_d, k)) \neq \emptyset$ and/or $v_d \cap \mathrm{int}(CH(v_o, v_d, k)) \neq \emptyset$. Vector overlay can resolve this complication. Consider the following definition:

Definition (line-polygon overlay):
For some polygon k and origin and destination points, v_o and v_d, the *line-polygon overlay* procedure splits a polygon into two or more disjoint, boundary-sharing faces with -v_ov_d such that $\cup_j f_j = k$.

Line-polygon overlay is an algorithm helpful in the derivation of the ESP. Let a straight line segment l with end points v_o and v_d, the origin and destination, respectively, be obstructed by polygon k. This means that $l \cap \mathrm{int}(k) \neq \emptyset$. In addition, let either v_o or v_d be *contained* within $\mathrm{CH}(k)$. It is assumed that v_o and/or v_d do not both lie on the resulting convex hull, so for some v_o, v_d, and k, then $v_o \cap \mathrm{int}(CH(v_o, v_d, k)) \neq \emptyset$ and/or $v_d \cap \mathrm{int}(CH(v_o, v_d, k)) \neq \emptyset$. The line-polygon overlay operation then divides the polygon k into multiple faces f_j using line segment l such that $\cup_j f_j = k$ with face edges formed by l (or segments of l, $\hat{l} \subseteq l$) and edges of k (or segments of edges defining k) such that the convexpath algorithm can generate feasible paths using the faces instead of k itself. The algorithm to generate the ESP detailed in Hong and Murray (2013b) then constructs $\mathrm{CH}(v_o, f_j)$ and $\mathrm{CH}(v_d, f_j)$ for each face f_j (or subpolygon), ensuring that the resulting graph includes both v_o and v_d. Any edge l' on the identified hulls that intersects the interior of the obstacle, $l' \cap \mathrm{int}(k) \neq \emptyset$, is not included in the resulting graph.

Theorem 2. *Given v_o and v_d obstructed by object k and that v_o and/or v_d do not lie on the convex hull, $v_o \cap int(CH(v_o, v_d, k)) \neq \emptyset$ and/or $v_d \cap int(CH(v_o, v_d, k)) \neq \emptyset$. The line-polygon overlay operation applied to polygon k and line segment l defined as -v_ov_d enables a graph to be derived containing the ESP.*

Proof: A full proof is provided in Hong and Murray (2013b). However, consider the case when $v_o \in \text{int}(\text{CH}(k))$ and/or $v_d \in \text{int}(\text{CH}(k))$. Without loss of generality, let the line l between v_o and v_d divide k into some finite countable set of faces f_j. By Viegas and Hansen (1985) we know that some collection of edges connecting vertices between vertices on f_j will contain the ESP. Then, by Hong and Murray (2013a) we have that the shortest path around f_j must lie on convexpath derived graph around them. However, the splitting line introduces *new* vertices into these obstacles at the intersections of l and f_j. Therefore, we can use the convexpath graph around f_j induced by the l and remove all links between vertices introduced in the splitting process. As this procedure satisfies the conditions presented in Hong and Murray (2013a), it is guaranteed to produce a graph sufficient for the ESP to be both optimal and feasible for the obstacle being divided. ■

Hong and Murray (2013b) prove that the convexpath graph for any set of obstacles K, where the convex hull for some obstacles contain v_o and/or v_d, includes the ESP. The reason why this technique is needed is because no convex hull between the obstacle and the interior vertex will contain an edge connecting the interior vertex to the vertices of the polygon. To handle this, the central realization of the proof is that k can be exchanged for $\cup f_j$, and the convexpath graph of $\cup f_j$ is then sufficient for ESP optimality.[1] This is because $\cup f_j$ derived using the dividing line l is exactly the original obstacle k. Feasibility and optimality conditions for this case are identical to the original problem because the convexpath graph is valid for all f_j. That is, feasibility and optimality conditions are preserved because the polygon-line overlay procedure keeps all conditions sufficient for an ESP solution intact.

Using the above theorems, the convexpath algorithm derives a graph by iteratively constructing convex hulls around impeding obstacles and the origin and destination vertices. The graph expands incrementally by replacing obstructed arcs of a hull with a subconvex hull. Hong and Murray (2013a,b) proved that the resulting graph guarantees inclusion of the ESP. Empirical results have demonstrated that the convexpath algorithm is up to 60 times faster than visibility graph and local visibility graph approaches.

Spatial Filtering

The convexpath algorithm is highly efficient for identifying a graph and finding the ESP compared to existing alternatives, particularly the visibility and local visibility graph approaches. However, performance tends to degrade as obstacle density increases because the graph expands in a way that ultimately includes obstacles that have no impact on ESP travel. Fig. 1 illustrates this point; the graph generated using convexpath systematically expands to include all obstacles.

In Fig. 1a, the graph that results from convex hulls around obstacles k_1 and k_2 includes an arc that intersects obstacle k_3. This requires more iterations of the convexpath algorithm to resolve infeasible arcs. Thus, the next iteration shown in Fig. 1b eliminates the intersecting arc with obstacle k_3, replacing it with corresponding subconvex hulls, but in doing so introduces one or more arcs in the graph that are obstructed by obstacle k_4. Fig. 1c shows resolution of infeasible arcs in the graph with obstacle k_4, but new arcs are introduced that intersect with obstacles k_5, k_6, and k_7. These are resolved in Fig. 1d, identifying the graph that avoids all obstacles and includes the optimal ESP. While this process is theoretically correct and fundamentally sound, resolving arcs around obstacles k_3, k_4, k_5, k_6, and k_7 has no actual impact on

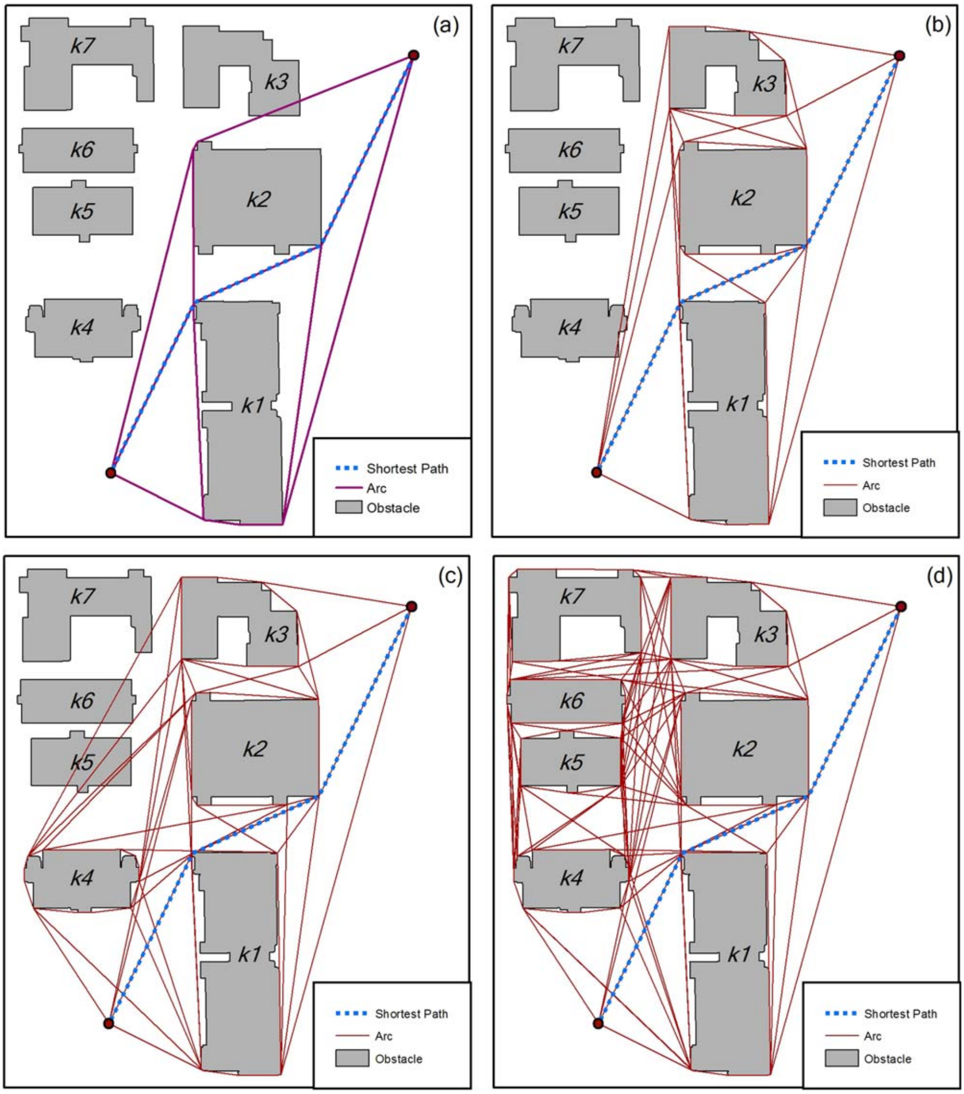

Figure 1. Iterative convexpath graph construction process.

the ESP for the given origin and destination in this case. This suggests potential for improving convexpath efficiency if obstacles can be eliminated from consideration under certain conditions without impacting the validity of the resulting graph and optimality of the ESP.

Observation 1: A feasible path $\hat{P}$ between v_o and v_d around obstacles K satisfies the condition $\hat{P} \cap \text{int}(K) = \emptyset$.

Observation 2: The optimal ESP P^* is that path for which $|P^*| \leq |\hat{P}|$ for all other feasible paths $\hat{P}$ where $|.|$ is the operator defining the length of the path.

Thus, a path is the ESP if and only if Observations 1 and 2 are satisfied. Note that nothing in these sufficient conditions for solving the ESP suggests that *all* obstacles define/comprise the graph through which the optimal path will traverse. Rather, it is only necessary to implicitly or explicitly consider all paths and associated obstacles so long as these two conditions are satisfied at termination. The challenge therefore is developing an approach that can filter out obstacles when they have no impact on the ESP, reflecting their implicit consideration. Interestingly, this is sort of the idea behind local visibility graph approaches, such as proximity-based filtering methods to reduce obstacles in graph construction (Zhang et al. 2005; Gao et al. 2011; Li, Gao, and Lu 2011). While these approaches improve computational performance compared to visibility graph, they may exclude the ESP and are less efficient than convexpath (see Hong and Murray 2013a). To seek out and exploit efficiencies through spatial filtering, we detail an approach that enhances convexpath, maintaining its theoretical properties and optimality conditions. This new approach is referred to as convexpath-sf (convexpath-spatial filtering).

The convexpath algorithm generates a graph of edges and vertices, G, through which the optimal ESP can be found. G is obtained through an iterative process where an initial graph is found then expanded and/or modified strategically in subsequent iterations. At any iteration t, graph G_t is augmented, continuing the process until no arc intersects any obstacles (see Fig. 1). The initial graph, G_0, consists of the origin and destination connected by the straight-line segment arc. Subsequent iterations involve replacing arcs that intersect obstacles with their corresponding subconvex hulls. By construction, G_t is always a connected graph with one or more paths from v_o to v_d, but some arcs may be infeasible if they intersect the interior of an obstacle.

The insight gained from the final graph in Fig. 1d is that computational effort is being devoted to graph expansion/modification for arcs and obstacles that are actually irrelevant to the optimal ESP. For example, convex hulls were derived and arcs added/replaced to deal with obstacle k_7 that does not impact on the ESP (see Fig. 1a–c). It may, therefore, be possible to filter out or eliminate obstacles in such cases. To develop a valid and efficient spatial filter, consider how convex hulls around obstacles help to spatially define potential shortest paths.

Lemma 1. *Given the convexpath graph G for origin and destination points v_o and v_d, a set of identified obstacles Ψ, and the set of all obstacles K such that $\Psi \subseteq K$, there exists a shortest path $P^* \in G$ that dominates all other possible paths around obstacles in the set Ψ, that is, a path exists in G that satisfies Observations 1 and 2.*

Proof: The proof follows from Hong and Murray (2013a). If G contains the shortest feasible path between v_o and v_d, then the path discovered must only consider obstacles within $k \in \Psi$. Assume that there exists some shorter path P' including obstacles $k \notin \Psi$ that is shorter than P^*. Consider the graph G as composed of edges fully contained within CH (v_o, v_d, Ψ) and edges forming the boundary of CH(v_o, v_d, Ψ) itself. By Theorem 1, either some fully contained feasible path exists or a boundary path is the shortest feasible path. This implies no shorter path between v_o and v_d can exist outside G, so P' cannot exist and the assumption is contradicted. Thus, there must exist a path in G around obstacles in $k \in \Psi$ that satisfies Observations 1 and 2.■

With Lemma 1, it is not necessary to explicitly evaluate all obstacles as long as a feasible path has been identified. This establishes a spatial bound on the search space. Considering the boundary paths of CH(v_o, v_d, Ψ) and the paths contained within CH(v_o, v_d, Ψ) in the convexpath graph G, either an interior path is feasible and shortest or a boundary path is feasible and shortest. If the interior path length is large, then Ψ can be thought of as one large composite obstacle and the shortest path is on the boundary of CH(v_o, v_d, Ψ). This means that the search space can be narrowed considerably, as either an interior path or a boundary path of G will constitute a lower bound on all possible paths.

Let us now consider the construction of a convexpath graph over all iterations. Given a graph G_t at iteration t, let δ_t^* be the current shortest path in the convexpath graph connecting the origin to the destination that does not intersect any currently identified obstacle $k \in \Psi_t$. Thus, δ_t^* is a valid shortest travel route around the identified obstacles, and is a lower bound on the optimal ESP as well. This information is helpful for graph construction in subsequent iterations and offers potential as a spatial filter because an obstructed arc can be assessed in terms of its potential to contribute to the optimal ESP. Specifically, assume for iteration t that graph G_t has a valid path δ_t^* between v_o and v_d around the current set of obstacles Ψ_t, where $\Psi_\text{t} \subseteq K$. If G_t includes an edge l that intersects the interior of an obstacle k such that $k \notin \Psi_t$ then $l \cap \text{int}(k) \neq \emptyset$. Accordingly, based on the convexpath algorithm, this edge must be replaced by a corresponding subconvex hull, $\text{CH}(v_1, v_2, k)$ where l is defined as -$v_1 v_2$. However, if the edge l is not part of δ_t^*, then, by definition, any path from the origin to the destination that includes l is longer than $|\delta_t^*|$ and there is neither a reason to include edge l nor a need to derive its corresponding subconvex hull. Thus, a spatial filtering approach can be devised based on this insight. This holds for any iteration t.

Theorem 3. *Given G_t and G_{t+1} for origin v_o and destination v_d, the length of the current shortest path δ_t^* will not decrease as t increases. That is, $|\delta_{t+1}^*| \geq |\delta_t^*|$.*

Proof: For some iteration t, G_t contains edges around identified obstacles Ψ_t such that $\Psi_t \subseteq K$. δ_t^* is feasible for Ψ_t. Assume δ_t^* is still infeasible with respect to $k \in K$, which means that there exists some k for which $\delta_t^* \cap \text{int}(k) \neq \emptyset$. The next iteration, t+1, an infeasible edge $l \in \delta_t^*$ (where $l \cap \text{int}(k) \neq \emptyset$) is replaced with the boundary of $\text{CH}(l, k)$ and k is added to $\Psi_{t+1} = \Psi_t \cup k$. Next, the algorithm constructs G_{t+1} avoiding obstacles in Ψ_{t+1}. There must then be some new shortest path δ_{t+1}^* in G_{t+1} that dominates all other obstacle avoiding paths by Lemma 1 and Hong and Murray (2013a). For each infeasible edge in δ_t^*, the new feasible paths in G_{t+1} cannot be shorter than their corresponding infeasible edges in G_t. This implies that the shortest path in the next iteration is some path in G_{t+1} that is at least as long as the current shortest path in G_t. This implies that the length of the shortest path does not decrease as t increases because infeasible arcs are replaced by their corresponding subconvex hulls. ■

Due to this nondecreasing property, the shortest paths provide a spatial filter to identify obstacles for evaluation that is sufficient for optimality. The nondecreasing length of shortest paths in each iteration means that shortest paths keep being impeded by at least one obstacle until they become feasible. As the shortest paths are lower bounds for each iteration, it is computationally efficient to evaluate only obstacles that impede shortest paths compared to evaluating all obstacles that impede any arc of the graph. This *shortest path spatial filter* is utilized to select only the obstacles that impede the shortest path at each iteration. Therefore, the total number of considered obstacles can be significantly reduced.

With optimality guaranteed at each step of the process when the shortest path spatial filter is used, this provides an early termination criterion that guarantees both feasibility and

Algorithm: convexpath-sf

1) Initialization. Given the region and set of all obstacles $k \in K$, set of discovered obstacles $\Psi = \emptyset$ and graph $G = \emptyset$.

2) Generate $\overline{v_o v_d}$, and add the arc and end points to G.

3) Derive the shortest path δ^* from v_o to v_d in G. If arc $l \cap int(k) \neq \emptyset$ for any $l \in \delta^*$, $k \in K$, then add k to Ψ. If nothing is added, terminate process. Optimal ESP found.

4) For all $l \in G$ and $k \in \Psi$ such that $l \cap int(k) \neq \emptyset$, replace l in G with $CH(l, k)$. If any end point of l ($l = \overline{v_1 v_2}$) not on hull boundary, then split k into faces f_j using line-polygon overlay and generate hulls for each face as detailed previously.

5) Go to step 3.

Figure 2. Convexpath-sf algorithm.

optimality of the final ESP. The algorithm for deriving a graph containing the optimal ESP using spatial filtering is described in Fig. 2. While implied, a formal proof of convexpath-sf optimality is now given.

Theorem 4. *The convexpath-sf algorithm terminates at an optimum feasible shortest path.*

Proof: Let the shortest path at the last iteration be δ_t^* in graph G_t identified by convexpath-sf. As discussed above, δ_t^* is both feasible and the shortest path in graph G_t at iteration t. Then, assume that G_t does not contain the optimal ESP, meaning that some feasible path δ' exists such that $\delta' \notin G_t$, $|\delta'| \leq |\delta_t^*|$, and $\delta' \cap \mathrm{int}(K) = \emptyset$ by Observations 1 and 2. In this case, if $|\delta_t^*| = |\delta'|$, then δ_t^* is equivalent in length to one of many possible optimal paths. By Theorem 3, this must be a lower bound, so convexpath-sf must have terminated at one of many potential ESPs and the assumption that δ_t^* is not an ESP is untenable. If $|\delta'| < |\delta_t^*|$, then δ' dominates δ_t^*. By Lemma 1, δ' must dominate all other feasible paths that exist in G_t for Ψ_t. However, δ_t^* dominates all other feasible paths that exist in G_t for $\Psi_t \subseteq K$. Thus, δ' cannot exist because, for any set of obstacles, δ_t^* is the lower bound on all potential ESP lengths and must be shortest. ■

Put differently, the convexpath-sf algorithm finds a path that is both feasible and optimal by satisfying Observation 2 in each iteration and terminates as soon as Observation 1 is satisfied. However, it should be noted that δ_t^* does not satisfy Observation 2 for *all* obstacles in K, but instead satisfies it for all paths around Ψ_t, the obstacles *identified* up to that iteration. Each iteration's shortest path, δ_t^*, provides a lower bound on the ESP, meaning Observation 2 is satisfied for each Ψ_t. But, Observation 1 is not satisfied until the last iteration, when δ_t^* becomes feasible for all obstacles $k \in K$. Because Observation 2 is always satisfied for path candidate δ_t^* over obstacles Ψ_t and $|\delta_t^*|$ is nondecreasing according to Theorem 3, convexpath-sf satisfies both Observation 1 and Observation 2 for all obstacles $k \in K$ at termination.

Convexpath-sf can be contrasted with convexpath, illustrated in Fig. 1. As shown in Fig. 3, the graph constructed by convexpath-sf is significantly smaller than the unfiltered convexpath graph (Fig. 1d). While the convexpath algorithm produces a graph with 124 edges, the convexpath-sf algorithm derives a graph with 18 edges, substantially smaller in size. As will be detailed below, this enhanced efficiency translates directly into faster

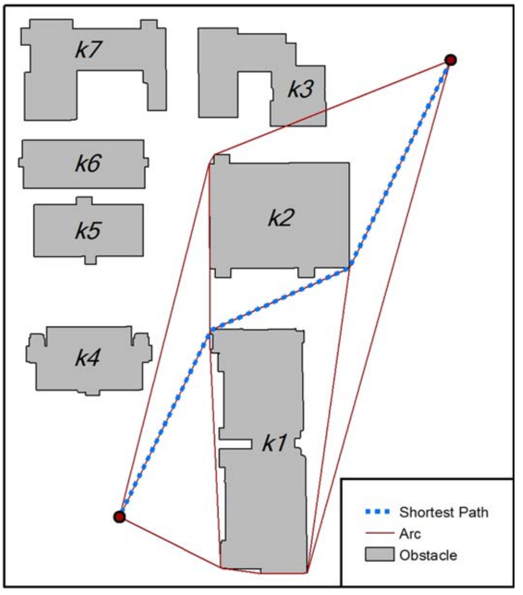

Figure 3. Resulting graph and ESP using the convexpath-sf algorithm.

computational processing. In this case, graph construction using convexpath-sf is over seven times faster. This substantial improvement is possible because the shortest path spatial filter explicitly considers only two obstacles, whereas convexpath evaluates all seven obstacles in ESP derivation.

As suggested, the efficiency of the convexpath approach is significantly improved by the spatial filtering method developed in this article. By reducing the number of obstacles evaluated, the resulting graph and required computing time are notably smaller than convexpath. Furthermore, the shortest path spatial filter does not require significant additional computing time, as it uses existing spatial information in intermediate graphs.

Application Results

To evaluate the efficiency of the convexpath-sf algorithm to support wayfinding, three different planning contexts are evaluated (Fig. 4). To assess impacts of increased obstacles on performance, each planning context has a different number of obstacles. For the low obstacle density context, travel across Tampa, Florida considering six obstacles are utilized (Fig. 4a). The study area was also used in Hong and Murray (2013a). The medium-density case represents travel through a building interior, and was examined in Hong and Murray (2013b) (Fig. 4b). This has 50 obstacles, including walls and tables. For the high density obstacle context, a portion of the Arizona State University campus is considered (Fig. 4c). There are 178 buildings that impede travel. To avoid any potential collision with an obstacle, a small buffer is applied to each obstacle, one foot for the medium and three feet for high density contexts. A total of 30 origin-destination pairs are considered for each travel context, with all cases encountering at least one impeding obstacle. The convexpath and convexpath-sf algorithms were implemented using Python. Derivation of the ESP in the various planning contexts was done on an Intel i5 personal computer (2.80 GHz) with 12 GB of RAM. In all cases, the optimal ESP is found.

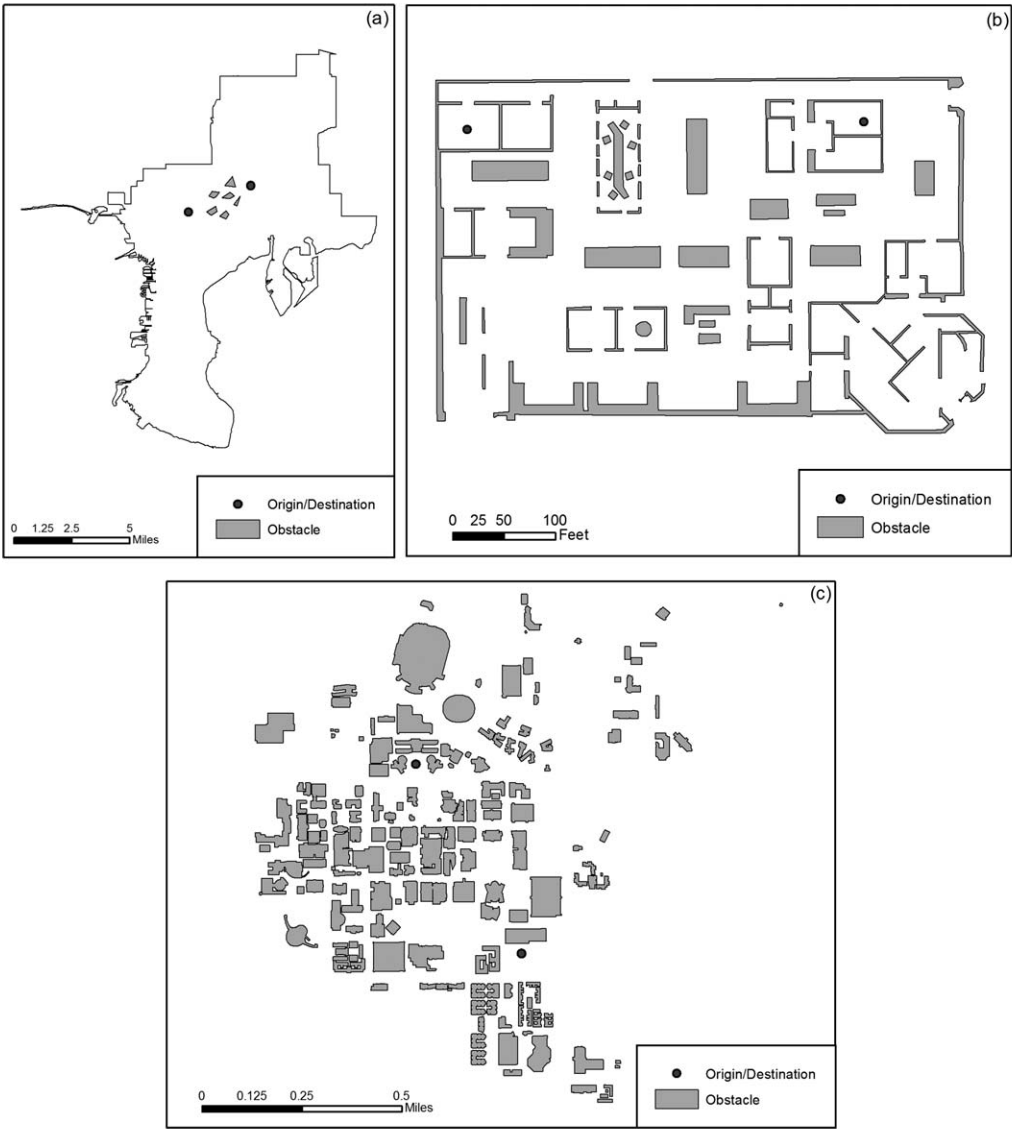

Figure 4. Travel path planning contexts, each showing one origin-destination pair: (a) Low obstacle density; (b) Medium obstacle density; and (c) High obstacle density.

Comparative results for the two algorithms are summarized in Table 1, and average relative performance difference in percentage term is shown in Fig. 5. The average number of obstacles, average number of arcs generated and average computing time is reported in Table 1 for the 30 different origin-destination paths to be found in the different obstacle density contexts. Table 1 indicates that convexpath evaluates 2.23 obstacles, on average, in the low-density case in deriving the graph needed to find the ESP. This increases to 38.03 and 74.67 in the medium and high cases for convexpath. In contrast, convexpath-sf explicitly evaluates only 1.47, 4.80, and 9.63 obstacles, on average, for the three different contexts. The significance of this, of course, is that a smaller graph results requiring less computing time. In the high density case, as an example, 5104.13 arcs on average are included in derived graphs requiring some

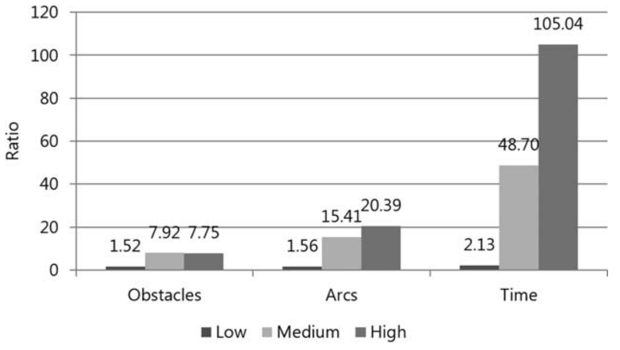

Figure 5. Average relative performance comparison of 30 origin-destination pairs in the different obstacle density contexts using the convexpath and convexpath-sf algorithms.

208.88 s using convexpath, whereas only 250.37 arcs on average are included in the graph requiring only 1.99 s using the spatial filtering algorithm (convexpath-sf). Comparative differences are shown in percentage terms in Fig. 5 indicating that convexpath-sf explicitly assess only 13.12% of the obstacles, generating 5.14% of the arcs needing 1% of the computing time. The performance trend is quite obvious in that as the number of obstacles increases, larger graphs and more computing time are required of the convexpath approach. The spatial filtering approach, however, is able to mitigate graph size growth and associated computational effort. This reduction is greatest in the higher density context, as suggested in Fig. 5. Convexpath-sf requires 53.42% of the computing time in the low-density context, 10.25% in the medium-density context, and only 1% in the high density context.

Discussion and Conclusions

The application results show that the convexpath-sf algorithm enhances capabilities for deriving the ESP in real time, especially in cases where a high density of obstacles is encountered. Table 1 and Fig. 5 clearly show that the effectiveness of the filtering technique increases with increasing obstacle density. Most certainly, the convexpath-sf algorithm can derive an ESP in real time, even in high obstacle density cases. For the high density obstacle context, convexpath-sf derives the ESP (graph and solution) in less than 2 s for 20 of the 30 cases. For the medium obstacle density context, 21 of the origin-destination pairs took less than 0.6 s using the convexpath-sf algorithm.

Another noteworthy point is that convexpath-sf is less influenced by oddly shaped obstacles. For example, the medium density obstacle set shown in Fig. 4b has several long and

Table 1 Average Performance Comparison of 30 Origin-Destination Pairs Using the Convexpath and Convexpath-sf Algorithms

	Convexpath			Convexpath-sf		
Density	Obstacles	Arcs	Time (s)	Obstacles	Arcs	Time (s)
Low	2.23	11.93	0.02	1.47	7.63	0.01
Medium	38.03	2366.27	27.60	4.80	153.53	0.57
High	74.67	5104.13	208.88	9.63	250.37	1.99

curved obstacles. This can lead to additional computing due to the inclusion of more obstacles that must be explicitly evaluated. However, convexpath-sf is insensitive to obstacle shape.

It is worth that there are differences between spatial filtering techniques developed in this research and local visibility graph approaches. The convexpath-sf approach essentially utilizes two spatial filters: convex hull (minimum bounding geometry) and shortest path (lower bound on the ESP). Although the fundamental strategy is similar, filtering out irrelevant obstacles/vertices, other approaches are based on simple proximity spatial filters (Zhang et al. 2005; Gao et al. 2011) and have limited general capabilities (Hong and Murray 2013a). The approach of Kim et al. (2004) can be seen as somewhat similar to convexpath and convexpath-sf since they utilize convex hull, however, their approach is only able to evaluate circular obstacles and can only eliminate obstacles in the context of a local visibility approach for graph construction. In contrast, convexpath-sf utilizes two spatial filters that enable efficient graph construction.

Another unique aspect of the convexpath-sf algorithm is its use of an identified shortest path in the resulting graph, regardless of feasibility, as a bound for the ESP. One might wonder why this approach focuses on deriving infeasible paths, which is unlike many other approaches. For example, operations research approaches commonly focus on bounding the problem by resolving the gap between a current feasible solution and the best possible infeasible solution. Branch and bound is such an approach, restricting the size of the problem search space by implicit consideration of unlikely branches. Through well-ordered evaluation, entire branches of a problem search tree can sometimes be pruned. At optimality, the upper and lower bound must converge. In contrast, the convexpath-sf algorithm uses a different tactic, focusing on the most promising path rather than worrying about finding an initial feasible solution.

The ESP is essential for spatial analysis and wayfinding. Numerous solution techniques have been developed for the ESP, but are not capable of supporting real-time path derivation. Although the convexpath approach is a major advance, a high number of obstacles can limit its performance. To address this, a spatial filtering method was developed to improve computational processing. By utilizing shortest paths as a bounding filter, the convexpath-sf algorithm derives the ESP very efficiently. Empirical results clearly support capabilities for real-time derivation of the ESP even when there is a high density of obstacles.

Note

1 Postprocessing is necessary to remove arcs that intersect the interior of obstacle *k*.

References

Aneja, Y., and M. Parlar. (1994). "Technical Note—Algorithms for Weber Facility Location in the Presence of Forbidden Regions and/or Barriers to Travel." *Transportation Science* 28, 70–6.

Asano, T. (1985). "An Efficient Algorithm for Finding the Visibility Polygon for a Polygonal Region with Holes." *IEICE Transactions (1976-1990)*, 68, 557–9.

Asano, T., L. Guibas, J. Hershberger, and H. Imai. (1986). "Visibility of disjoint polygons." *Algorithmica* 1, 49–63.

Bailey, T. C., and A. C. Gatrell. (1995). *Interactive Spatial Data Analysis*. Longman Scientific & Technical Essex, New York; Harlow Essex, England: Prentice Hall.

Batta, R., A. Ghose, and U. S. Palekar. (1989). "Locating Facilities on the Manhattan Metric with Arbitrarily Shaped Barriers and Convex Forbidden Regions." *Transportation Science* 23, 26–36.

Bekker, J., and J. Schmid. (2006). "Planning the Safe Transit of a Ship Through a Mapped Minefield." *ORiON: The Journal of ORSSA* 22, 1–18.

Bhattacharya, P., and M. L. Gavrilova. (2007). "Voronoi Diagram in Optimal Path Planning." In *4th International Symposium on Voronoi Diagrams in Science and Engineering, ISVD '07*, 38–47. Pontypridd, Wales.

Bischoff, M., and K. Klamroth. (2007). "An Efficient Solution Method for Weber Problems with Barriers Based on Genetic Algorithms." *European Journal of Operational Research* 177, 22–41.

Carling, K., M. Han, and J. Håkansson. (2012). "Does Euclidean Distance Work Well When the P-Median Model Is Applied in Rural Areas?" *Annals of Operations Research* 201, 83–97.

Chang, K.-Y., G. E. Jan, and I. Parberry. (2003). "A Method for Searching Optimal Routes with Collision Avoidance on Raster Charts." *The Journal of Navigation* 56, 371–84.

de Berg, M., O. Cheong, and M. Van Kreveld. (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag: Berlin Heidelberg.

de Smith, M. J., M. F. Goodchild, and P. Longley. (2007). *Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools*. Troubador Pub: Leicester, UK.

Fagerholt, K., S. Heimdal, and A. Loktu. (2000). Shortest Path in the Presence of Obstacles: An Application to Ocean Shipping." *Journal of the Operational Research Society* 51, 683–8.

Fotheringham, A. S., C. Brunsdon, and M. Charlton. (2000). *Quantitative Geography: Perspectives on Spatial Data Analysis*. Sage: London.

Gao, Y., J. Yang, G. Chen, B. Zheng, and C. Chen. (2011). "On Efficient Obstructed Reverse Nearest Neighbor Query Processing." In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 191-200. ACM: Chicago, Illinois, USA.

Ghosh, S. K., and D. M. Mount. (1991). "An Output Sensitive Algorithm for Computing Visibility Graphs." *SIAM Journal on Computing* 20, 888–910.

Guibas, L. J., and J. Hershberger. (1989). "Optimal Shortest Path Queries in a Simple Polygon." *Journal of Computer and System Sciences* 39, 126–52.

Habib, M. K., and H. Asama. (1991). "Efficient Method to Generate Collision Free Paths for an Autonomous Mobile Robot Based on New Free Space Structuring Approach." In *Proceedings IROS'91. IEEE/RSJ International Workshop on Intelligent Robots and Systems' 91. Intelligence for Mechanical Systems*, 563–567. IEEE: Osaka, Japan.

Hershberger, J., and S. Suri. (1999). "An Optimal Algorithm for Euclidean Shortest Paths in the Plane." *SIAM Journal on Computing* 28, 2215–56.

Hong, I., and A. T. Murray. (2013a). "Efficient Measurement of Continuous Space Shortest Distance Around Barriers." *International Journal of Geographical Information Science* 27, 2302–18.

Hong, I., and A. T. Murray. (2013b). "Efficient Wayfinding in Complex Environments: Derivation of a Continuous Space Shortest Path." In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science*, 61. ACM: Orlando, Florida, USA.

Katz, I. N., and L. Cooper. (1981). "Facility Location in the Presence of Forbidden Regions, I: Formulation and The Case of Euclidean Distance with One Forbidden Circle." *European Journal of Operational Research* 6, 166–73.

Kim, D. S., K. Yu, Y. Cho, D. Kim, and C. Yap. (2004). "Shortest Paths for Disc Obstacles." *Computational Science and Its Applications–ICCSA* 2004, 62–70, Assisi, Italy.

Klamroth, K. (2001). "A Reduction Result for Location Problems with Polyhedral Barriers." *European Journal of Operational Research* 130, 486–97.

Kwan, M.-P., and J. Lee. (2005). "Emergency Response After 9/11: The Potential of Real-Time 3D GIS for Quick Emergency Response in Micro-Spatial Environments." *Computers, Environment and Urban Systems* 29, 93–113.

Larson, R. C., and G. Sadiq. (1983). "Facility Locations with the Manhattan Metric in the Presence of Barriers to Travel." *Operations Research* 31, 652–69.

Li, Z., Y. Gao, and Y. Lu. (2011). "Continuous Obstructed Range Queries in Spatio-Temporal Databases." In *2011 International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, 267–70. IEEE: Guiyang, China.

Lozano-Pérez, T., and M. A. Wesley. (1979). "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles." *Communications of the ACM* 22, 560–70.

Mitchell, J. S. B. (2000). “Geometric Shortest Paths and Network Optimization.” In *Handbook of Computational Geometry*, 633–702, edited by J. U. J. R. Sack. North-Holland Publishing Co.: Amsterdam, Netherlands.

O’Sullivan, D., and D. Unwin. (2010). *Geographic Information Analysis*. Wiley: Hoboken, NJ.

Papadopoulou, E., and D. Lee. (1995). “Efficient Computation of the Geodesic Voronoi Diagram of Points in a Simple Polygon.” *Algorithms—ESA’95*, 238–51, Springer: Corfu, Greece.

Pocchiola, M., and G. Vegter. (1996). “Minimal Tangent Visibility Graphs.” *Computational Geometry* 6, 303–14.

Rogerson, P. A. (2010). *Statistical Methods for Geography: A Student’s Guide*. Sage Publications Limited: Los Angeles, CA, USA.

Rohnert, H. (1986). “Shortest Paths in the Plane with Convex Polygonal Obstacles.” *Information Processing Letters* 23, 71–76.

Schmickl, T., and K. Crailsheim. (2007). “A Navigation Algorithm for Swarm Robotics Inspired by Slime Mold Aggregation.” In *Swarm Robotics*, 1–13, edited by E. Şahin, W. Spears, and A. T. Winfield. Springer: Rome, Italy.

Szymanski, M., T. Breitling, J. Seyfried, and H. Wörn. (2006). “Distributed Shortest-Path Finding by a Micro-Robot Swarm.” In *Ant Colony Optimization and Swarm Intelligence*, 404–11, edited by M. Dorigo, L. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle. Springer: Brussels, Belgium.

Tsou, M.-C. (2010). “Integration of a Geographic Information System and Evolutionary Computation for Automatic Routing in Coastal Navigation.” *Journal of Navigation* 63, 323.

Viegas, J., and P. Hansen. (1985). “Finding Shortest Paths in the Plane in the Presence of Barriers to Travel (for any l_p-Norm).” *European Journal of Operational Research* 20, 373–81.

Wangdahl, G. E., S. Pollock, and J. B. Woodward. (1974). “Minimum-Trajectory Pipe Routing.” *Journal of Ship Research* 18, 46–49.

Welzl, E. (1985). “Constructing the Visibility Graph for N-Line Segments in O (n2) Time.” *Information Processing Letters* 20, 167–71.

Zhang, J., D. Papadias, K. Mouratidis, and Z. Manli. (2005). “Query Processing in Spatial Databases Containing Obstacles.” *International Journal of Geographical Information Science* 19, 1091–111.

Zhang, L., L. Zhang, R. Peng, G. Li, and W. Zou. (2011). “Determination of the Shortest Time Route Based on the Composite Influence of Multidynamic Elements.” *Marine Geodesy* 34, 108–18.