SOLVING GLOBAL, TWO-DIMENSIONAL ROUTING PROBLEMS USING SNELL'S LAW AND A* SEARCH

R. F. Ritchbourg, Neil C. Rowe, Michael J. Zyda, Robert B. McGhee

Department of Computer Science, Naval Postgraduate School, Monterey, California

Abstract

Long-range route planning is an important component in the intelligent control system of an autonomous agent. Most attempts to solve it with map data rely on applying simple search strategies to high-resolution, node-and-link representations of the map. These techniques have several disadvantages including large time and space requirements. We present an alternative which utilizes a more intelligent representation of the problem environment. Topographical features are represented as homogeneous-cost regions, greatly reducing storage requirements. Then, the A* search strategy is applied to a dynamically created graph, constructed according to Snell's law. Testing has shown significant speed improvements over competing techniques.

Introduction

Route planning ability is an important component of any intelligent control scheme for autonomous agents. It has two basic aspects, long-range and local-motion. Planning for local motions is well studied and the most easily solved problem. This type of planning is generally limited to reasoning about movements within the area local to the autonomous agent, local in the sense that the environment is within scanning range of the agent's sensor equipment. The problem typically is treated as binary in nature: every point in the environment can be classified as either impassable (part of an obstacle) or passable. To solve binary-terrain problems, the environment can be described as a set of polygons defining obstacle boundaries. Any point not inside an obstacle polygon is traversable. Then, a graph $G=(V,L)$ is created where $V$ is a set of points including all obstacle vertices as well as the start and goal location. The set $L$ contains a link between any two members of $V$ that can be connected by an unobstructed line segment. Given this representation, the problem is reduced to finding the shortest-distance path in a graph, a problem that can be solved by several standard techniques. There are several variations on this general approach.

The second class of route planning problems is intended to provide routes over long ranges. Here, some form of map is required since the range of movement is too great to be sensed by the on-board equipment of the agent. A more important difference is that binary assumption, local route planning is untenable. There are many situations in which optimality is critical, minimizing exposure to danger along a route as an example. Typically, many different cost regions exist over a large expanse of terrain. When optimality is important, terrain traversal cost differences must be recognized. Thus, it is not reasonable to assume that all passable areas have the same traversal costs. As an example, riding a bicycle through wet sand requires much more time and effort than riding on a roadway. Thus, the least time consuming path for a bicycle rider between two points on a sandy beach could easily be a longer-distance roadway route when the alternative is the shortest distance route across the sand.

Wavefront Propagation

The best-understood and most often used method to solve the long range routing problem can be characterized as a wavefront propagation technique. To employ it, the map must be converted to a lattice. The ratio of the number of points in the lattice to the physical distance represented by the map determines the resolution of the problem. Each lattice point must have a link to each of its immediately adjacent neighbors (normally the eight neighboring points, although the degree of connectivity can be any multiple of four). A cost is associated with each link. This cost must represent the cost for the agent to traverse the corresponding area in the physical environment. (Links to points within obstacle areas have infinite cost.) A solution route is generated by first "positioning" the agent at either the start or goal (or both if a bidirectional strategy is employed). Then, the algorithm simulates the passage of time while the agent is allowed to "move" in all directions. The effect is to generate a series of wavefronts, depicting possible locations for the agent at successive instances of time. When the wavefront reaches the goal, (or the wavefronts touch in the bidirectional form) a solution path is retrieved by referencing backpointers, solving wavefront gradients, or a similar technique.

The wavefront technique can solve the long-range routing problem. However, there are several drawbacks. First, omnidirectional search of a high-resolution lattice can be expensive computationally. The number of cells that must be examined is roughly proportional to the number of cells in a circle with a radius equal to the number of cells between the start and goal. There is also a digital bias inherent in the lattice representation that results in "stair-step" approximations to straight line segments. Because of this, the method may return a set of digitally equivalent approximations to the optimal path. An expensive procedure is required to deduce the true, non-digitally biased optimal solution from this set. Finally, most of the search effort is usually wasted in unproductive portions of the lattice.
Snell’s Law Based Route Planning

It has been shown that Snell’s law, commonly used in optics, also characterizes minimal cost paths in the long range routing problem. We illustrate the rule in Figure 1. Let \( B \) denote a linear boundary and let the cost accrued per unit of distance traveled on one side of \( B \) be \( C_1 \) and on the other side of \( B \) be \( C_2 \). Then the \( S-I-G \) path is the minimal cost path between \( S \) and \( G \) if and only if \( \sin(\theta_1)/C_1 = \sin(\theta_2)/C_2 \). (Note that \( \theta_1 \) is the angle between the \( S-I \) segment and a normal to \( B \) while \( \theta_2 \) is the angle between the \( I-G \) segment and a normal to \( B \). Also, a 2:1 cost ratio was used to create Figure 1 where \( C_2 \) is more expensive than \( C_1 \).) This rule is intuitively appealing. Snell’s law forces a straight line between \( S \) and \( G \) to be bent so that increased distance in the low cost region \( (C_1) \) is traded for decreased path distance in the high cost \( (C_2) \) area.

Application of Snell’s law relies on a homogeneous region cost map as opposed to the discrete node-and-link representation of a continuous environment used in the wavefront technique. Thus, there is no digital bias in the solution paths and the computational cost of the algorithm is not tied to map resolution or the distance between start and goal. However, there are difficulties in applying Snell’s law to the homogeneous-cost region representation of the environment. First, using Snell’s law to find optimal cost point to point paths requires iterative search, there is no known closed-form solution for this problem. Secondly, the law allows "blind regions" to exist, regions that cannot be reached by any path obeying only Snell’s law.

A Suitable Problem Representation

The Snell’s law-based solution method requires a map of the environment where homogeneous-cost regions are depicted. A homogeneous-cost region is an area bounded by an arbitrary polygon such that all points inside have identical traversal costs. That is, whenever an agent is inside a homogeneous-cost region, the cost accrued by that agent per unit of distance traveled in any direction is constant.

For simplicity of discussion, we assume a ternary classification scheme for constructing the homogeneous-cost regions. Each homogeneous region on the map is classified into one of three disjoint classes: impassable obstacle, traversable at high cost, or traversable at low cost. The low cost (optimal) regions need not be specifically represented. We assume that these areas constitute the "background" for the map. Thus, obstacle and high cost regions are overlaid onto the optimal cost background.

Search and Snell’s Law

There is no known closed-form solution for a Snell’s law problem of finding a least cost, point to point path. Instead, some iterative technique such as bisection or golden-section search must be employed. The requirement for search interacts with the fact that Snell’s law allows blind regions to exist. Figure 2 exemplifies this dilemma. Suppose that we desire the minimal cost path from \( S \) to \( G \) and that a high-cost region, represented by the shaded triangle, lies between them. Assume a 2:1 cost ratio. \( P_1 \) is a Snell’s law path from \( S \) through vertex \( V \) that is refracted by intersecting sides \( B_1 \) and \( B_2 \) of the high cost region. \( P_2 \) is a Snell’s law path that begins at \( S \) and travels infinitesimally close to vertex \( V \) so that the high-cost region is not intersected. Because of the relative positions of \( P_1 \) and \( P_2 \), point \( G \) is in a blind region. Any Snell’s law path that intersects side \( B_1 \) of the high-cost region to the left of \( V \) will also intersect side \( B_2 \) of the region and yield a path that lies entirely to the left of path \( P_1 \). Any Snell’s law path that passes by \( V \) to the right does not intersect any side of the high-cost region and thus lies entirely to the right of path \( P_2 \). Therefore, paths \( P_1 \) and \( P_2 \) form physical limits on the location of Snell’s law paths from \( S \) and infinitely close to vertex \( V \). The only points that can be reached by a Snell’s law path involving sides \( B_1 \) and \( B_2 \) of the high cost region lie in the area between \( P_2 \) on the left and \( P_1 \) on the right. The only points reachable by Snell’s law paths projected from \( S \) that do not intersect the high cost region lie to the right of path \( P_2 \) or left of \( P_1 \). Thus, no path from \( S \) to \( G \) exists where each heading change along the path is determined by applying Snell’s law to the intersection of the path and a linear boundary of the high-cost region. Any iterative search strategy based on this premise will fail.

Because of the existence of blind regions, we must ensure the success of an iterative search before the search process begins. We must ensure that the portion of the cost-map under consideration "contains" the point that is to be the object of the search. We ensure this by creating "wedges" within the search space. Wedges define the portions of the map that can be reached by Snell’s law paths from a specified point and involve a specific set of region boundaries. In Figure 2, paths \( P_2 \) and \( P_1 \) form a wedge that begins at point \( S \) and involves sides \( B_1 \) and \( B_2 \) of the high cost region.

Creating the wedges serves another important purpose: iterative search techniques can use the information gained from proposed paths on one iteration of ray following to guide
the paths of successive attempts. Thus, the paths proposed on consecutive iterations must provide consistent information. To guarantee this, we must ensure that the paths all intersect exactly the same sequence of region boundaries. In Figure 3, path \( P_1 \) intersects boundaries \( B_1 \) and \( B_2 \) of the high cost region while path \( P_2 \) intersects \( B_1 \) and \( B_3 \). (Again, a 2:1 cost ratio was used to generate the figure.) In searching for a path to \( G \), \( P_1 \) indicates that the next attempt should intersect boundary \( B_1 \) between \( I_1 \) and \( V_1 \), while \( P_2 \) indicates that the interval between \( I_2 \) and \( V_2 \) is the most promising. Clearly, the information here is inconsistent and any iterative search technique will be confused.

To correct the situation in Figure 3, the upper portion of the wedge formed by \( P_1 \) and \( P_2 \) should be refined by creating new wedges associated with vertex \( V_2 \) of the high cost region. This implies a general principle. Once a wedge has been formed, the information available within that wedge is only guaranteed to be consistent up to the closest unsolved region vertex. A region vertex is unsolved if a Snell's law path to a vertex within a wedge has not yet been found. As an example, vertex \( V_2 \) in Figure 3 is an unsolved vertex.

Once a Snell’s law path to a vertex within a wedge has been found, three new wedges can be formed. Figures 4 and 5, again using a 2:1 ratio of costs, exemplify this process. In Figure 4, \( V_2 \) is the closest unsolved vertex and the Snell’s law path from \( S \) across boundary \( B_2 \) to \( V_2 \) has been found. Any Snell’s law path that intersects \( B_2 \) to the left of \( I \) will intersect \( B_1 \) to the left of \( V_2 \). Similarly, any Snell’s law path intersecting \( B_1 \) to the right of \( I \) will intersect \( B_2 \) to the right of \( V_2 \). Thus, to create new wedges in which the same sequence of region boundaries will be intersected, we split the \( S-I-V_2 \) path into two Snell’s law paths at \( V_2 \). Figure 5 depicts the result of splitting the path at \( V_2 \). Two new paths, \( P_2 \) (the left side split, intersecting \( B_1 \) and \( B_2 \)) and \( P_R \) (the right side split, intersecting \( B_3 \) and \( B_4 \)), have been created. They refine the original wedge formed by \( P_1 \) and \( P_2 \) into new wedges including one formed by \( P_1 \) and \( P_2 \) and another wedge defined by \( P_R \) and \( P_2 \). There is also a third wedge formed by paths \( P_1 \) and \( P_2 \). In Figure 5, this last wedge is empty (contains no points) since \( P_2 \) and \( P_R \) intersect each other immediately at vertex \( V_2 \). However, in Figure 2, \( P_2 \) and \( P_R \) define a similar wedge that is not empty. Here, the wedge is a blind region for paths from \( S \) and is not directly reachable by a simple Snell’s law path. To reach points inside the blind region, such as \( G \) in Figure 2, care must be taken in applying Snell’s law at vertex \( V_2 \). In the limit, we can view sides \( B_2 \) and \( B_3 \) of the region as a single, non-linear boundary in which \( B_2 \) is joined to \( B_3 \) by a circular curve infinitesimally close to the location of \( V_2 \). With this construction, Snell’s law paths can be made to enter the blind region by intersecting the boundary at points close to \( V_2 \). There is a simpler way to achieve the same result without merging multiple linear boundaries into a single non-linear one. Temporarily abandon adherence to Snell’s law at the point initiating the blind region (vertex \( V \) in Figure 2). The law is then reapplied in the further search within the wedge (beyond vertex \( V \) in Figure 2). This results in a path whose consecutive line segments obey Snell’s law unless their common endpoint is a vertex at the base of a blind region.

There is an analogy from the field of optics that applies in cases where paths include a blind region vertex as a turn point. The situation is similar to (single slit) diffraction optics.
(however, the path is constrained to remain within the boundaries of the wedge). Let a vertex at the base of a blind region be denoted as a diffraction vertex. Thus, vertex $V$ of Figure 2 is a diffraction vertex.

Thus, we have a general strategy to apply Snell's law in the search for optimal paths. Given an initial wedge, find the closest unsolved vertex that it contains. Find the Snell's law path to the unsolved vertex. Split the path into two paths at the vertex so that the original wedge is refined into three new wedges. Each refinement extends the number of boundaries that are guaranteed to be intersected by any path within the wedge.

The Initial Wedges

Given a homogeneous region cost map, a start location, and a goal location, the shortest-distance start-to-goal path can be found by ignoring the high-cost region polygons and applying binary-case methods to the remaining obstacle regions. Clearly, such a path may not have optimal cost, but it must be a feasible solution. The cost of traversing this shortest distance path can be computed for the original ternary map; this is an upper bound on the cost of the optimal start-to-goal path. Given an upper bound, an ellipse containing the optimal path can be constructed. The maximum distance that can be traveled at optimal cost while not accruing a total cost greater than that of the shortest distance path can be computed. Call this maximum distance the bounding distance. The set of all points such that the distance from the start to that point added to the distance from that point to the goal is equal to the bounding distance defines an ellipse that has the start and goal locations as foci. The optimal path must lie entirely within the physical limits defined by the ellipse boundary. By definition, any path between start and goal that exits the ellipse must have cost greater than that of the shortest distance path. So only those homogeneous regions that are at least partially within the ellipse need be considered.

For computational simplicity, circumscribe the ellipse by a rectangle so that each side of the rectangle is tangent to the ellipse. Call this rectangle the bounding box. To create the initial wedges, project two Snell's law paths from the start, perpendicular to the direction to the goal, to the bounding box. These paths define two initial wedges, as depicted in Figure 6. (For illustrative purposes, Figure 6 is not generated according to actual data.) It is apparent that the lower wedge of Figure 6 is superfluous. This may not be true in all instances, since the optimal path may occasionally travel away from the goal. As an example, it may be beneficial to take the shortest way out of a high cost area, regardless of the direction to the goal.

Pruning Criteria

Given that we can create, refine, and search wedges, we can find all feasible (i.e., locally optimal) start-to-goal paths within the search space. Then, to find the optimal path, we can select the best feasible path. However, such a brute force search can be excessive computationally. Also, it is possible to prune some wedges (and thus their further refinements) from the search space.

First, consider wedges associated with blind regions. Any start-to-goal path within such a wedge must follow a fixed path from the start to the region vertex at the base of the blind region (the diffraction vertex). The cost of this path can be computed. A lower bound estimate on completing the path is obtained if we assume the remaining distance from the region vertex to the goal could be traveled at the optimal cost. Summing the estimate and the fixed path costs results in a lower cost bound for any start-to-goal path within the wedge. If it exceeds the current upper bound on the cost of the optimal solution, the wedge can be pruned. Also, if there is some other path from the start to the diffraction vertex, and the other path has a lower start to vertex fixed path cost, the new, higher-cost wedge can be pruned since it must be true that the optimal start-to-goal path is also the optimal path to all points on the path itself and otherwise, the optimal start to goal path can be "shortcut", resulting in a lower cost path.

Wedges not associated with blind regions can also be pruned. One purpose of building the wedges is that of constantly increasing the known boundaries that must be crossed by Snell's law paths within the wedge. There is a minimum cost path within any wedge that intersects all the known boundaries. The minimum cost path here is readily available without search.

This minimum cost path can be obtained by examining how the two Snell's law paths defining the wedge exit the last known boundary. If both of these paths are rotated in the same direction (either clockwise or counter-clockwise) as they cross the last boundary, then the minimum cost path through the wedge is one of them. Otherwise, a Snell's law path within the wedge that exits the last known boundary perpendicularly must exist; this path is the minimum cost path through the wedge and can be easily found since the function describing the cost of any path through the wedge is convex with distance along the last-found boundary $^6$.

Finding this minimum cost path through the wedge provides a lower-bound cost for a portion of a possible start-to-goal path. We can also compute a lower bound on the cost from the last known boundary to the goal by assuming that the minimum distance between them can be traveled at the
optimal cost. Summing the two costs provides a lower bound on the total cost of any start-to-goal path starting within the wedge. Again, if the lower bound cost exceeds the global upper bound on the cost of the optimal path, (initially taken from the ellipse analysis and later taken from the cost of subsequent start-to-goal paths) the wedge can be pruned.

Searching Within Wedges

The upper and lower cost bounds presented above also constitute a way to rate wedges. The wedge having the lowest lower bound for a complete start to goal path passing through it should be the first wedge to be refined. Clearly, we have a method well suited to a strategy using an ordered agenda and A* search. However, A* will not be used to explore a static graph of nodes and links. Instead, we use Snell's law and the search technique to dynamically create a tree with branching factor of at most three where each node is a region vertex. The root of the tree is the start location. Leaves of the tree are created when a wedge contains no points (and thus can not be refined further), when a wedge is pruned, or when a start to goal path is located. The three branches at tree nodes correspond to each of the three sub-wedges that can be created based on the Snell's law path to the node (i.e. region vertex).

We have now presented the basic concepts for an algorithm to conduct a Snell's law search over a homogeneous region cost map for the optimal path between two known points. The first step is to find a feasible solution and its cost; this becomes the cost upper bound. From the initial solution, create the bounding box and the two initial wedges. Rate the wedges and place them in order of increasing lower bound cost estimates on an ordered agenda. Exclude from the problem all vertices and portions of region boundaries outside the bounding box. Until the agenda is empty or the cost of the most favorable wedge on the agenda exceeds the current upper bound, repeat the following steps.

1) Remove the best-cost (first) wedge from the agenda.
2) Locate the closest vertex in the interior of the wedge to which no optimal path has yet been found. (If there is no such point, consider the next wedge on the agenda.)
3) Compute the Snell's law path, within the wedge, to this point.
4) If the point was the goal update the global upper-bound cost if the new path has lower cost.
5) Otherwise, create three new wedges. Rate each wedge by a lower bound on the cost of a start-to-goal path through it. Based on this estimate, either prune the wedge or insert it into the ordered agenda.

Implementation

There are many details involved in implementing this algorithm that we have not discussed. These include dealing with total internal reflections (situations in which Snell's law requires that the sine of the refraction angle (in the low cost region) is greater than 1), dealing with obstacles, finding the closest unsolved point, and similar issues. These issues have been solved in a prototype implementation of the Snell's law based algorithm.

A first version of the algorithm has been implemented in C-Prolog (an interpretive language). The algorithm solves problems given a ternary-representation of the homogeneous region cost map. The ternary representation was chosen since it is the simplest scheme that supports the development of a theory to solve n-ary map classifications. (It is also appropriate for some important autonomous agents.) Tests have shown that the algorithm performs well in a wide range of cases. Results indicate, however, that if the cost map includes many different homogeneous regions within a small area, a wavefront propagation technique is likely to be less time consuming.

Despite the worst-case superiority of wavefront techniques, our Snell's law based approach has several advantages. First, the method is suitable for parallel execution; the search process within a given wedge is almost entirely independent from the search within any other wedge. The only communication required is through the agenda (including consideration of the global upper bound). Thus, the algorithm is well suited to computer system architectures that support blackboard strategies. The method can also provide feasible solutions quickly as well as optimal solutions if more time is available. (As the cost ratio between low and high cost areas increases, a binary terrain solution that treats all cost regions as obstacles closely approximates the cost of the optimal solution.) The method avoids the problems associated with digitally biased paths and, as a result, returns path descriptions that contain the fewest turn points necessary to accurately describe the optimal path.

Example Solutions

Figures 7, 8, 9, and 10 depict solutions generated by the C-Prolog implementation of the Snell's law based algorithm. Each figure features a single high-cost region (the shaded polygon) overlaid on an optimal cost background. In Figure 7, the high cost region includes both the start and goal locations. The ratio of the two costs is 3:1. Note that the optimal path initially moves away from the goal to quickly exit the high-cost region. The path then follows along the region border until a good shortcut leading to the goal is found. Figures 8, 9, and 10 all show solutions paths between the same two points given the same region geometry but involving different cost ratios. In Figure 8, the ratio is 2:1 and a portion of the high cost area is included in the optimal path. In Figure 9, the ratio is 1:1; note that the optimal solution still contains a portion of the high cost region, although less of it. In Figure 10, the ratio is 8:1; at this ratio, the high cost region acts as an obstacle, and the shortest distance path around the region is the optimal path.

Future Work

We have not yet established an order class for the algorithm. Testing indicates that the worst case complexity may be exponential. However, we feel that a worst case would have to be a contrived example. The algorithm performs well in "average" cases that have been tested to date. A C-Prolog version of the wavefront algorithm has also been implemented and comparative performance testing of the two methods solving identical problems has begun.

An interesting extension is the development of a theory allowing the use of a mix of algorithms based on the problem at hand. That is, the system could use a wavefront algorithm when there are many distinct regions in a small area and use the Snell's law algorithm otherwise.
References


