Evolutionary Algorithm for Optimal Visual Coverage Path in Raster Terrain

Jie Li
Dept. of Electronic Engineering and Information Science
University of Science and Technology of China
Hefei, China
lijie610@mail.ustc.edu.cn

Changwen Zheng, Xiaohui Hu
National Key Laboratory of Integrated Information
System Technology, ISCAS
Beijing, China
cwzheng@hotmail.com, xhh@ios.cn

Abstract—There is a wide range of applications in both military and civil fields for optimal path problem in raster terrain whose cost is visibility information. For the path search problem whose objective is maximal average horizon, the traditional algorithms are unsuitable due to the characteristic of the problem. This paper presents a method based on evolutionary algorithm, which may rapidly get the optimal solution by designing multiple effective evolutionary operators and self-adaptively adjusting the probability of each mutator. Experiments show that the method presented is superior to the simulated annealing algorithm.

Keywords—evolutionary algorithm; average horizon; visual coverage; raster terrain

I. INTRODUCTION

Visibility analysis is also called intervisibility analysis, which applies computing geometry theory to analyze the visibility between viewpoints or viewpoints’ sets. It includes line of sight analysis from a point to another one and viewshed analysis from a point to a plane. It is widely applied in many fields such as environment resource management, city planning and military action analysis, etc [1]. Visual coverage of path in raster terrain refers to the region seen by a path, which is composed of the viewshed of all points in the path. According to different objectives for optimization, optimal visual coverage path may be classified as path with maximal or minimal visual coverage area [2], [3], path with maximal or minimal average visual range [4], etc. This paper addresses the method for searching path with maximal average visual area, which means searching a path with the widest average view from the start point to the end. The problem may be extended to reconnaissance path search in military, sightseeing path search in travel or city planning, so it has significant military and civil value.

Toussaint [5] firstly presented guard path search problem in 2D polygon region based on visibility analysis in 1981. Later Jönsson [6] applied line of sight analysis to military path search in raster terrain, which judged if the vehicle could be probed by checking the line of sight from enemy’s sentry to the vehicle. Lee and Stucky [7] firstly applied viewshed analysis to minimal cost path planning in raster terrain. Caldwell et al. [8] developed a toolbox of visibility analysis, which might use viewshed analysis to search the most visible path and the least visible path.

In the traditional path search algorithms based on viewshed analysis, the total cost of each path is composed of cumulative visibility cost of all points in the path. However, the total viewshed of paths is the union set of each point’s viewshed, not simple addition since viewshed of adjacent points in the path is overlapped. So Lu Min et al. [4] presented the concept of average horizon, which is the ratio of viewshed of path to length of path to judge the optimality of visual coverage path. Moreover, they used a simulated annealing algorithm for the optimal path problem whose objective is maximal average horizon. The simulated annealing algorithm has the advantage of high quality, robust initial value, general purpose as well as easily being realized, etc. Nevertheless, in order to get the optimal solution, the algorithm requires high initial temperature, slow speed of lowering the temperature, low termination temperature and enough many times’ samples, so the optimization process of the simulated annealing algorithm is very time-consuming. Evolutionary algorithm, as a meta-heuristic method, has wide applications for its intrinsic parallelism and rapid convergence. This paper presents a self-adaptive evolutionary algorithm to solve the optimal path problem based on average horizon. The algorithm may get optimal solution with relatively less time by designing proper population structure and evolutionary operators.

II. MODELING AND ANALYSING

A. Path with Maximal Average Horizon

Given a path \( P \) on raster terrain which is composed of a set of adjacent points, \( P = (p_1, p_2, \ldots, p_m) \). If the viewshed of \( p_i \) is \( V(p_i) \), the viewshed of path \( P \) denoted by (1) is the union set of all points’ viewshed in \( P \).

\[
V(P) = \bigcup_{i=1}^{m} V(p_i).
\]  

Visual coverage area of path \( P \) denoted by (2) is product of the number of points in the viewshed of path and unit area of raster \( \Delta s \).

\[
A(P) = \text{Area} \left( \bigcup_{i=1}^{m} V(p_i) \right) = \Delta s V(P).
\]

Two objectives need be considered during the process of getting optimal path, which are maximal visual coverage area and minimal length of path. For convenience, it is feasible to get a path with maximal average horizon by combining visual coverage area with length of path. Average Horizon \( Ra(P) \) of path \( P \) is defined as ratio of visual
coverage area to length. It evaluates how wide the visual area seen by a path with unit length is.

\[ Ra(P) = \frac{A(P)}{Length(P)}. \]  

(3)

The model is generally set up to get minimal objective for optimization methods. So reciprocal of average horizon is used as objective function.

\[ \min f(P) = \frac{1}{Ra(P)}. \]  

(4)

For unit area of raster \( \Delta A \) is a constant, objective function is denoted as (5).

\[ \min f(P) = \frac{Length(P)}{V(P)}. \]  

(5)

### B. Basic Property of Path with Maximal Average Horizon

**Theorem 1:** There exist arcs with negative weight, but not loops with negative weight.

**Proof:**

1) Assume a path \( P_0 = \{p_1, \ldots, p_k, \ldots, p_m\} \) composed of path segments \( P_1 = \{p_1, \ldots, p_k\} \) and \( P_2 = \{p_k, \ldots, p_m\} \). Let \( f_0, L_0, A_0 \) are cost, length and visual coverage area of \( P_0 \) and \( P_1 \) respectively. \( \Delta A \) is the change value of length and \( \Delta A \) is the change value of visual coverage area. \( \rho_A \) is the change ratio of visual area and \( \rho_L \) is the change ratio of length.

\[ \rho_A = \frac{\Delta A}{A} = \frac{(A_k - A_1)}{A_1}. \]  

(6)

\[ \rho_L = \frac{\Delta L}{L} = \frac{(L_k - L_1)}{L_1}. \]  

(7)

If \( \rho_A > \rho_L \), then

\[ \frac{(A_k - A_1)}{A} > \frac{(L_k - L_1)}{L}. \]  

(8)

\[ \Rightarrow \frac{A_k}{A} > \frac{L_k}{L}. \]  

(9)

\[ \Rightarrow \frac{L_k}{A} > \frac{L_0}{A_0}. \]  

(10)

\[ \Rightarrow f_0 < f_1. \]  

(11)

That is, there exist arcs with negative weight when change ratio of visual area is larger than that of length.

In Fig. 1, light gray region is the viewshed of path \( A \rightarrow B \rightarrow C \) while deep gray region is the viewshed of point \( D \). After path segment \( C \rightarrow D \) is added, the change ratio of length is 1/2 while the change ratio of viewshed is 5/9. The cost of \( A \rightarrow B \rightarrow C \) is 2/9 and cost of path \( A \rightarrow B \rightarrow C \rightarrow D \) is 3/14. The cost is decreased when length of path is increased.

2) The length and visual coverage area are both positive values. It is known from (3) and (5) that cost of any path is positive. Therefore, the cost of any loop in raster terrain is also positive. It is impossible to exist any loop with negative weight. Therefore, there exists optimal solution.

**Theorem 2:** Optimality principle does not hold true for the problem of path with maximal average horizon.

The traditional optimal path problems all satisfy optimality principle, which is whatever states and decisions in the past, for the states formed by previous decision, every decision of the rest must form optimal strategy. The following example illustrates that the problem of path with maximal average horizon does not satisfy this principle.

In Fig. 2, there are two paths from the start point \( A \) to the objective point \( E \), \( A \rightarrow B \rightarrow C \rightarrow E \) and \( A \rightarrow B \rightarrow D \rightarrow E \). The lengths of path segments \( B \rightarrow C \rightarrow E \) and \( B \rightarrow D \rightarrow E \) are equal. The region surrounded by dashed lines is visual area of path, and the shadow part \( O \) is the region not seen. Obviously, the optimal solution of sub problem from \( B \) to \( E \) is \( B \rightarrow C \rightarrow E \). But for global problem, the globally optimal path is \( A \rightarrow B \rightarrow D \rightarrow E \) since it is larger for the overlapped part of the path segments \( B \rightarrow C \rightarrow E \) and \( A \rightarrow B \).

Since optimality principle does not hold true for the problem of path with maximal average horizon, the traditional algorithms like A* and dynamic programming etc. are not suitable. Global optimization algorithms such as evolutionary algorithm may be used to solve it.

### III. EVOLUTIONARY ALGORITHM

Evolutionary algorithm is a global search algorithm inspired by evolutionism, which is independent of concrete problem. The main difference from traditional optimization algorithms is that it begins to search from initial population but not single individual, so it covers wide region and is advantageous to globally selecting optimum. There is significant effect for solving problems to creatively design and apply evolutionary operators and methods [9], [10]. This paper presents a method based on evolutionary algorithm to solve the problem of path with maximal average horizon. Multiple operators are designed and the probability of
choosing each mutator is self-adaptively adjusted during the evolutionary process so that the optimal solution space may be explored very rapidly. In following search process, multiple mutation modes are applied to the current optimal path to make it approach the optimum.

A. Chromosome Structure

The real-valued chromosome representation with variable number of genes is used. A chromosome corresponds to a path. As illustrated in Fig. 3, each node (gene) of path is specified by the coordinates of raster. To avoid excessive cost due to too many nodes, critical points are used. A path is specified by a sequence of critical points. The cost between critical points is gotten by inserting points.

The number of individuals in the population, the number of initial critical points and the maximal and minimal numbers of critical points are set in advance. Initial population is generated randomly, that is the coordinates of nodes in each path are random except the first and the last node representing the start point and the end respectively.

After initializing, the cost of each individual is computed according to (5). The lower the cost is, the higher the probability that the genes enter into next generation for their higher fitness.

B. Evolutionary Operators

Our evolutionary algorithm uses eight operators: one crossover operator and seven mutation operators illustrated in Fig. 4.

1) Crossover

The parent individuals are divided randomly into two parts, then the first part of first parent individual is combined with the second part of the second parent individual, and the second part of first parent individual is combined with the first part of the second parent individual. There can be different number of critical points in the two parent individuals.

2) Multiple points perturb mutator

The coordinates of several points in the path are randomly changed. The perturb mutator perturbs the coordinates of node (x, y), then determines the height coordinate z according to (x, y).

3) Single points perturb mutator

It is similar to the multiple points perturb mutator, but only one node’s coordinates are changed. The range perturbed may be changed gradually during the evolutionary process to improve the ability of both global and local search.

4) Insertion mutator

A new node is randomly inserted between two adjacent nodes. When insertion operator is used, if the number of critical points has attained the maximum preset by the algorithm, single point perturb operator is used. By limiting the number of critical points, excessive computing cost is avoided while still keeping the agility brought by variable length chromosomes.

5) Deletion mutator

A middle node selected randomly is deleted. Insertion mutator is used if the number of critical points attains the minimal value.

6) Smooth mutator

Smooth mutator cuts the sharp angle as illustrated in Fig. 4(f). Two new nodes in the path segments adjacent to the middle node selected are inserted, and then the middle node is deleted. The probability of selecting a middle node is proportional to the turn angle of the node. In general, the larger the turn angle is, the more overlapped viewshed near the node is. For example, path A→B→C→E has larger turn angle in node B in Fig. (2), there is very large overlapped viewshed for path segments A→B and B→C. So average horizon of path may be effectively improved by smoothing the sharp angle.

7) Swap mutator

Randomly select two middle nodes in a path, swap the coordinates of two nodes to form a new path.

8) Reversion mutator

Randomly select two middle nodes in a path, swap the coordinates of two nodes, and reverse all nodes between two nodes to form a new path.

C. Self-adaptive Adjusting

It is very important for the entire evolutionary process to design proper evolutionary operator. The proper probability of using different mutators seriously affects the performance of evolutionary algorithm. To exert the function of every mutator, the probability of using every mutator may be self-adaptively adjusted according to their effects to paths during the evolutionary process.

At the beginning of the evolutionary process, every mutator is used with equal probability. The performance of mutators is evaluated for every fixed generations and the
probability is properly adjusted. The performance of every mutator is evaluated by the ratio of the number of paths improved by the mutator to the total number of paths improved by all the mutators.

\[ P_i = \frac{I_i}{\sum_{j=1}^{7} I_j}, \quad i = 1, \ldots, 7 \]  

(12)

D. Local Search

After having evolved for some generations, the current best solution is close to optimum. The region including optimum is located so that it is unnecessary to explore the new search space but exploit the current space to look for the optimum. The similarity of solutions in the population is very high so that it is hard for crossover operator to produce better solution, so a process like hill climbing begins. The population is initialized using the current optimal solution. The different mutators are applied to each individual, and the better individual is selected to initialize the population again. This process is repeated until the solution has not been changed for some continuous generations.

E. Algorithm description

1) Randomly generating \( N \) initial paths.
2) Evaluating each path.
3) Randomly selecting parent individuals to cross to generate new individuals.
4) Evaluating the new individuals.
5) Selecting \( N \) optimal paths from individuals of two generations to form the next population.
6) Applying the mutators to all individuals. The current optimal individual is compared with that before mutating. If the current optimal individual is superior to the old one, the old one is replaced.
7) If the period \( T \) is attained, evaluating the performance of each mutator, then modifying the probability used.
8) If the number of evolutionary generations \( G \) is attained, goto 9, otherwise goto 3.
9) Initializing the population by the current optimal individual to form a population containing \( N \) same individuals.
10) Applying the mutators to each individual.
11) Evaluating the individuals after mutating.
12) Selecting current optimal path. If the optimal individual has not been changed for \( G' \) continuous generations, the process terminates, otherwise goto 9.

IV. Experimental Results

The algorithm proposed in this paper is developed using vc.net. All the experiments are done on a 3.00GHz Pentium PC with 512M RAM. The size of the terrain data used is 256 \( \times \) 256 and the grid resolution is 30 meters. The viewshed of each point in the terrain is computed using XDRAW method [11] assuming the altitude of observers is 1.6 meter and the radius of the region of interest is 1000 meters. Only 1 bit is necessary to denote whether a grid point is visible for another one, so about 32M memory space is needed.

The size of population is selected according to a large amount of experiments. If the size of population is too large, it will be very time-consuming to search the optimum. But if the size of population is too small, the states of chromosomes in the population is so less that the probability of generating good individuals is too small for the algorithm to sufficiently search for optimum in the solution space, that is premature convergence. Proper size of population assures both diverse chromosomes and reduced computing time. In our algorithm, the number of individuals in population is 30. The start point of path is (35, 35), the end is (221, 221). The number of the initial critical points is 10 (including the start point and the end). The minimal number of critical points is five while the maximum is 30. The probability of crossover is one. The initial probability of selecting each of seven mutators is all \( 1/7 \). For multiple points perturb, the probability of whether mutating each critical point of a path is 0.5. The period of evaluating the performance of every mutator is 50 generations. After evaluating, the probability of selecting mutators is adjusted. The number of evolutionary generations is 200. After 200 generations, the current optimal path is used to initialize 30 individuals to form a new population. Here, no crossover operator is used. The mutators do not consist of swap mutator and reverse mutator since both of two mutators will no longer increase the fitness of paths. For single point mutator the range of mutation is reduced. Mutators are selected in equal probability. If the optimum generated in each generation has not been changed for 200 continuous generations, the evolutionary process terminates.

A. Self-adaptive Adjusting

Fig. 5 illustrates the ratio of the number of paths improved by each mutator in every 50 generations.

Table I compares the cost of optimal paths gotten by two modes of selecting the mutators, one is equal probability and the other is self-adaptively adjusting the probability of selecting mutators. It is the statistical result of continuously running for 15 times respectively. The number of evolutionary generations is 200. It shows that the mode of self-adaptively adjusting is superior to that of equal probability.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Min</th>
<th>Max</th>
<th>Range</th>
<th>St dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal probability</td>
<td>157548.7</td>
<td>165777.2</td>
<td>15902.1</td>
<td>2409.1</td>
</tr>
<tr>
<td>self-adaptive</td>
<td>155459.7</td>
<td>159570.1</td>
<td>157586.3</td>
<td>1314.8</td>
</tr>
</tbody>
</table>

Table II.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Horizon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EA</td>
<td>1933.9</td>
<td>2000.2</td>
<td>1957.3</td>
</tr>
<tr>
<td>SA</td>
<td>1895.6</td>
<td>1996.5</td>
<td>1935.9</td>
</tr>
<tr>
<td>Run Time(s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EA</td>
<td>700</td>
<td>1644</td>
<td>1042.5</td>
</tr>
<tr>
<td>SA</td>
<td>4894</td>
<td>5358</td>
<td>5058.2</td>
</tr>
</tbody>
</table>
B. Compared with Simulated Annealing Algorithm

Tab. II compares the statistical result of continuous running 15 times for our algorithm (EA) and simulated annealing algorithm (SA). For the simulated annealing algorithm, the number of critical points in paths is 10, the number of generations is 300000 and the initial temperature is 1000000.

Fig. 6 is the gray map of optimal paths gotten by EA and SA respectively where the triangle refers to the start point of the path and the square refers to the end of the path.

V. CONCLUSIONS

This paper analyzes path search problem whose objective is maximal average horizon in detail. It may be put into a category of existing arcs with negative weight but no loops with negative weight. Due to the characteristic of viewshed amalgamation, general path search methods for this type of problem are unsuitable. And that it is infeasible to store viewshed information of each point in the path since very large amount of memory space is needed. In addition, it is very time-consuming to compute the viewshed. So critical points are used to represent the path. Only viewshed of critical points is stored while paths between the critical points are gotten by inserting some points, which are adjacent to the points on the straight line between critical points. Thus each time only the viewshed of changed path segments need be recomputed. On the other hand, chromosome structure with variable number of genes is used and the number of genes is constrained so that time and space are saved while assuring agility and multiplicity of chromosomes. Further, by self-adaptively adjusting the probability of using each mutator, optimal solution may be found rapidly. The algorithm presented is superior to the simulated annealing algorithm in both quality of solution and efficiency at the same time the stability is maintained.

ACKNOWLEDGMENT

The authors wish to thank the National Natural Science Foundation of China for contract (60873183) under which the present work is possible.

REFERENCES