<table>
<thead>
<tr>
<th>Title</th>
<th>A genetic algorithm for the minimum weight triangulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Qin, KH; Wang, WP; Gong, ML</td>
</tr>
<tr>
<td>Citation</td>
<td>IEEE International Conference on Evolutionary Computation Proceedings, Indianapolis, IN, 13-16 April 1997, p. 541-546</td>
</tr>
<tr>
<td>Issued Date</td>
<td>1997</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/45578">http://hdl.handle.net/10722/45578</a></td>
</tr>
<tr>
<td>Rights</td>
<td>This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.; ©1997 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
</tbody>
</table>
A Genetic Algorithm for the Minimum Weight Triangulation

Kaihuai Qin\textsuperscript{a,b}, Wenping Wang\textsuperscript{a} and Minglun Gong\textsuperscript{b}

ABSTRACT
In this paper, a new method for the minimum weight triangulation of points on a plane, called Genetic Minimum Weight Triangulation (GMWT for short), is presented based on the rationale of genetic algorithms. Polygon crossover and its algorithm for triangulations are proposed. New adaptive genetic operators, or adaptive crossover and mutation operators, are introduced. It is shown that the new method for the minimum weight triangulation can obtain more optimal results of triangulations than the greedy algorithm.

Key Words genetic algorithm, adaptive genetic operators, crossover, mutation, the minimum weight triangulation, greedy algorithm, Delaunay triangulation.

1 INTRODUCTION
Triangulation plays an important role in finite element methods, computer aided geometric design, numerical analysis and so on. Although triangulations of scattered points on the plane have been the subject of significant research in the past few decades, the minimum weight triangulation (MWT for short), in which the objective is to minimize the sum of the edge lengths of the triangulation of a set of points, is still one of the difficult problems. Interpolating values of two-argument functions is one of the applications of MWT\cite{1}.

The greedy triangulation and the Delaunay triangulation [1,2,3] are two well-known methods, but it is proved\cite{4,7} that neither greedy nor Delaunay triangulation approximates the optimum. In fact, the problem of MWT for a set of points may be NP-hard\cite{8,9} except the restricted classes of input points whose exact solutions can be found easily\cite{9,10,11}.

Fig. 1 Triangulations of points

In GMWT, a set of lower matrices $M_i$ ($i = 1,2,\ldots$) corresponding to a set of triangulations is referred to as a population. The matrix $M_i$ is called the $i$th string of the population.

2 GENETIC MINIMUM WEIGHT TRIANGULATION
2.1 ENCODING MECHANISM FOR TRIANGULATION
Fundamental to the GA is the encoding mechanism for representing the solutions of optimization problems. For a point set $\{P_i\}_{i=0}^n$, which contains $n$ points, at most $n(n-1)/2$ edges can be connected. Thus we can represent the status of a triangulation by a lower triangular matrix. That is, if the edge (or, a straight-line) between point $i$ and point $j$ is selected in the triangulation, $m_{ij}$ is 1, otherwise $m_{ij} = 0$. This encoding method can make the genetic operators defined later easy to handle.

For example, we can get the following lower triangular matrices $M_1$ and $M_2$ of the triangulations shown in Fig. 1 (a) and (b), respectively:

In GMWT, a set of lower matrices $M_i$ ($i = 1,2,\ldots$) corresponding to a set of triangulations is referred to as a population. The matrix $M_i$ is called the $i$th string of the population.
2.2 FITNESS FUNCTION

The selection or reproduction operator is dependent on how fit a given chromosome is. The fitter the string, the greater is its probability of going from one generation to the next. In order to guarantee that a fitter string has larger fitness value, we can define the fitness function as follows:

$$f(M) = \sum_{i} \sum_{j} l_{ij}(1 - m_{ij}),$$

where \(l_{ij}\) is the distance between \(P_i\) and \(P_j\).

2.3 SELECTION

A new selection strategy a bit similar to the elitist strategy\[13\] is used for selection here. That is, the fitness value \(f_i\) of the best string \(M_i(k)\) of generation \(k\) is compared with the fitness value \(f_j\) of the worst string \(M_j(k + 1)\) of generation \(k + 1\), if \(f_i > f_j\), then string \(M_i(k)\) is substituted for \(M_j(k + 1)\). By means of this strategy, the maximum fitness value of the population does not decrease as the process of evolution continues.

2.4 CROSSOVER OF TRIANGULATIONS

2.4.1 POLYGON CROSSOVER

After the selection is completed, a mating pool of strings will be obtained. Although there are a lot of techniques for crossover, such as single-point crossover, multi-point crossover, uniform crossover and so on, most of them can be used for the MWT. They do not guarantee that the crossover operators produce legal triangulations.

In the paper, every string is represented by a lower triangular matrix in which the elements are either 1 or 0. It is very important that crossover operator produces only legal triangulations. Therefore, a new crossover operator, called polygon crossover, is defined as follows:

First, find out the elements that are equal to 1 from the resulting matrix \(M'\) obtained by exclusive-or of the selected two strings \(M_1\) and \(M_2\). In \(M'\), element \(m_{ij}\) (or, the element at \(i\)th row and \(j\)th column of matrix \(M'\)) equal to 1 implies that there is an edge between \(P_i\) and \(P_j\) in one triangulation (say, \(M_1\)), without loss of generality) but not in another one (say, \(M_2\)). Then, randomly select an edge associated with the element equal to 1 in \(M'\), and determine the minimum polygon \(P_{min}\) that contains the points \(P_i\) and \(P_j\) and resides in both triangulations.

Let \(P_{max}\) be the convex hull of all the points, and \(S\) be the set of the edges that reside in \(P_{min}\).

$$I = \{(i, j) \in N \times N | e_{ij} \in S\}$$

where \(e_{ij}\) is the edge between the points \(P_i\) and \(P_j\), \(A = \{m_{ij}^1 | (i, j) \in I\}\),

\(B = \{m_{ij}^2 | (i, j) \in I\}\), where \(m_{ij}^1, m_{ij}^2\) are the elements of \(M_1\) and \(M_2\), respectively. A and B are regarded as the polygon chromosomes of the two parents \(M_1\) and \(M_2\), respectively. When \(P_{min} \neq P_{max}\), the polygon chromosomes A and B from the two parents are exchanged with different probabilities depending on the fitness values to generate two new offspring. The probability \(P_{dc}\) of polygon crossover is dynamically changed as follows:

$$P_{dc} = P_c + C_1 \frac{f_{max} - f}{f_{max} - f_{min}}$$

where \(P_c\) is the initialized probability of crossover, and \(C_1\) is a constant in the range 0.1–0.5. \(f_{max}\) and \(f_{min}\) indicate the maximum and minimum fitness values of the population in the current generation, and \(f\) gives the average of the fitness values of the two strings over which polygon crossover is to be performed.

The advantage of dynamically adaptive polygon crossover is as follows: If the fitness value of a string is low, then \((f_{max} - f)/(f_{max} - f_{min})\) is high. Increasing \(P_{dc}\) results in the disruption of the weaker strings. On the other hand, highly fit strings result in a low \((f_{max} - f)/(f_{max} - f_{min})\) value, forcing \(P_{dc}\) to be low. Therefore, the strings with higher fitness value are not disrupted as often as the weaker strings.

For example, as shown in Fig. 1, suppose that the straight-line \(P_1 P_3\) is selected. Then the minimum polygon is composed of \(P_1 P_4 P_5 P_6\). According to the triangulations of the minimum polygon one can get

\(A = \{m_{21}^2, m_{22}^2, m_{23}^2, m_{24}^2, m_{25}^2, m_{26}^2\}\), \(B = \{m_{11}^1, m_{12}^1, m_{13}^1, m_{14}^1, m_{15}^1, m_{16}^1\}\), from \(M_1\) and \(M_2\), respectively. A and B are regarded as the chromosomes.

The sub-triangulations associated with A and B are shown in Fig. 1. The offspring created with an adaptive probability by polygon crossover are shown in Fig. 2.

\[
M_1 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

\[
M_2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}
\]
2.4.2 ALGORITHM FOR FINDING POLYGON CHROMOSOMES

It is necessary in polygon crossover to find the polygon chromosomes from triangulations. An effective algorithm for finding the polygon chromosomes from strings $M_1$ and $M_2$, which are associated with triangulations 1 and 2, respectively, is given in Fig. 3. Regard the polygon generated at each step as an edge set equal to the union of all its contained triangles. Then all the polygons generated form a nested sequence with increasing areas, and are contained in a common polygon of both triangulations containing $P_{ij}$. Since $P_{min}$ is the unique minimum common polygon, the procedure will generate $P_{min}$ correctly as the first common polygon and terminate.

Algorithm 1

i) Select at random a straightline $PP_j$ according to the result of $M_1$ XOR $M_2$.

ii) Construct an "initial polygon" composed of edges $PP_j$ and $P_iP_i$. Without loss of generality, assume that line $P_iP_j$ resides in triangulation 1.

iii) Push the edges of the initial polygon into stack $S$.

iv) While $S$ is not null do

    popping an edge $e_0$ out from stack $S$;

    if $e_0$ resides in both triangulations 1 and

    triangulations 2

    then $e_0$ is one of the edges of the minimum polygon

    and added in the set of edges of the minimum

    polygon.

    else the other two edges of the triangle on the right

    side of edges $e_0$ in triangulations 1 are pushed

    onto the top of stack $S$.

endif

v) endwhile

vi) Determine the polygon chromosomes $A$ and $B$ of the

    strings $M_1$ and $M_2$, respectively, according to the

    minimum polygon.

2.5 MUTATION

The mutation operator involves the perturbation of two adjacent triangles, one of which is randomly chosen and the other is chosen to be adjacent to the first. However, not all mutations are legal. For instance, suppose that $m_{ij}$, which must be equal to 1, is selected in $M_j$ and the two triangles are linked by the edge $PP_j$ (as shown in Fig. 4). If the value of $m_{ij}$ is changed from 1 to 0, then the value of $m_{ks}$ (or $m_{sk}$ if $s>k$) must be changed from 0 to 1 at the same time. This implies that line $PP_{jk}$ must be replaced by line $P_kP_s$, but this operation can be done if and only if the legal triangulation can be kept (as shown in Fig. 4 (a)). If the new triangles created by mutation will be illegal, as shown in Fig. 5, then the mutation can not be done.

Algorithm 2

i) Initialize the population of size PopuSize

ii) while generation counter < the given number of

    generations do

    for each $i <$ PopuSize do

    select two strings randomly and determine their

    polygon chromosomes

    perform the polygon crossover and mutation

    endfor

endwhile

Fig. 3 Algorithm for polygon crossover

Fig. 4 Legal mutation

Fig. 5 Illegal mutation

2.6 GENETIC MINIMUM WEIGHT TRIANGULATION

2.6.1 STRUCTURE OF GENETIC MINIMUM WEIGHT TRIANGULATION

Combining the above segment of coding, fitness function evaluation, and the operators of selection, polygon crossover and mutation, a basic structure of the genetic algorithm for the minimum weight triangulation is given as follows:

Algorithm 2

i) Initialize the population of size PopuSize

ii) while generation counter < the given number of generations do

    for each $i <$ PopuSize do

    select two strings randomly and determine their

    polygon chromosomes

    perform the polygon crossover and mutation

    endfor

endwhile

Fig. 3 Algorithm for polygon crossover
calculate fitness values and perform the select operation
store the offspring
endfor
if the termination conditions have been met then stop

Fig. 6 Structure of GMWT

The algorithm of GMWT shown in Fig. 6 contains two loops. The inner loop deals with the evolution from one generation to the next one. The outer loop specifies the maximum number of generations through which the system is permitted to evolve.

2.6.2 TERMINATION CONDITIONS

The evaluation process can terminate in the following ways:

1) If all the strings in the population of a generation have nearly equal fitness values, then convergence has been reached.
2) If the system has evolved through the specified maximal number of generations, it can be terminated.

Other termination conditions, such as the upper limit to be reached by the average, best or worst value in the population, can be used, but they are not adopted in the paper.

3 CONVERGENCE OF THE ALGORITHM OF GMWT

Convergence to the global optimum is one of the advantages of genetic algorithms. However, if there are a few extraordinary strings, which have high fitness values, at the initial generation, using the selection criterion alone may lead to premature convergence because of the dominance of the highly fit strings. The linear fitness scaling\[13\] and a large population size can be used to avoid the premature convergence. In addition, the adaptive mutation and selection probability may help the genetic algorithm converge to the global optimum\[17\]. Also, rank-based fitness is a further improvement.

4 COMPUTING RESULTS

To evaluate the results of GMWT, examples compared with the greedy algorithm are given as follows.

Example 1. Triangulations of a set of points introduced by Levcopoulos[5]. It contains the following points: A so-called sink vertex $P_s$ with coordinates $(0,-1)$ and a blocking vertex $P_b$ with coordinates $(1,1)$; $\sqrt{n}$ pulling vertices, with coordinates $(0,-i)$ $i = 1,2,\ldots, \sqrt{n}$; $n - \sqrt{n} - 1$ crowd vertices, lying on the straight-line segment with endpoints $(3,3)$ and $(4,4)$. The lowest pulling vertex is called $P_t$ and the highest crowd vertex $P_u$ (see Fig. 7 (a)).

Fig. 7 Point set given by Levcopoulos

This point set is degenerate. In order to avoid degeneracy, we adjust the $x$ coordinates of pulling vertices to make them lie on a circular arc that goes through $P_s$ and $P_b$, such that the line passing through $P_t$ and $P_b$ is a tangent to the circular arc. Analogously, we adjust the $y$ coordinates of the crowd vertices to make them lie on a circular arc that goes through $P_u$ and $P_b$, such that the line passing through $P_u$ and $P_t$ is a tangent to that circular arc (see Fig. 7 (b)).

It has been proved that in greedy triangulation all the pulling vertices are connected with the blocking vertex $P_b$ and all the crowd vertices are connected with the lowest pulling vertex $P_t$ (see Fig. 8 (a)). Hence, the total length obtained by the GT is $\Omega(n \times \sqrt{n})$. The triangulation computed by the GMWT in our runs is much better than that by the greedy algorithm. Using GMWT, we can get the triangulation in which all the crowd vertices are connected with the highest pulling vertex $P_u$, and all the pulling vertices with the highest crowd vertex $P_u$ (see Fig. 8 (b)).

Example 2: Triangulations of twelve points.

544
The triangulation generated by greedy algorithm is shown in Fig. 9(a). Its total length is 1814.19, but genetic triangulation algorithm can converge to the triangulation whose total edge length is 1744.62 as shown in Fig. 9 (b).

**Example 3:** Triangulation of forty points selected randomly.

The triangulation generated by Greedy Triangulation by the round point.

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Length</td>
<td>%</td>
<td>Total Length</td>
</tr>
<tr>
<td>GT</td>
<td>528.21</td>
<td>159.19</td>
</tr>
<tr>
<td>GMWT</td>
<td>331.81</td>
<td>100</td>
</tr>
</tbody>
</table>

5 SELECTING PARAMETERS FOR GMWT

Five main parameters must be chosen in the GMWT algorithm. They are population size, adaptive constants $C_1$ and $C_2$, probability of polygon crossover, and probability of mutation. Each of the parameters affects the performance of GMWT.

5.1 POPULATION SIZE

In Figures 11 and 12, the best, worst and average fitness values of the population which are obtained after computing twenty times for Example 3 are marked by square, rhombus and triangle points, respectively, and the fitness value of Greedy Triangulation by the round point.

The choice of population size is very important. If the selected population size is too small, then the algorithm will converge quickly without sufficient processing of the schemata, and it may result in premature convergence. On the other hand, a large population size will lead to a highly time-consuming search.

Fig. 11 shows the performance of GMWT for different population sizes, for Example 3. When population size < 30, although the algorithm converges fast, it may occur premature convergence. Thus, a population size that avoids the problem of premature convergence has to be selected. For instance, a proper population size can be selected between 30 and 60 in Fig. 11.

5.2 ADAPTIVE CONSTANTS

Adaptive genetic operators can be used for enhancing performance of GMWT. In general, let $C_1 = C_2 = C$ in GMWT approach. Fig. 12 shows the performance of GMWT for different adaptive constants, for Example 3. A proper adaptive constant $C$ can not only speed up the convergence of GMWT, but also contribute to avoidance of premature convergence. In practice, the adaptive constant $C$ is generally selected between 0.1 and 0.5. As shown in Fig. 12, the best selection of the adaptive constant $C$ may be 0.4 for Example 3.

5.3 PROBABILITIES OF POLYGON CROSSOVER AND MUTATION

The probabilities $P_c$ and $P_m$ of polygon crossover and mutation have to be selected carefully. Neither very low nor very high probabilities of polygon crossover and mutation are good for the process of evolution of GMWT. In practice, adaptive constants can dynamically balance the influence of the probabilities of polygon crossover and mutation on the evolutive process of GMWT. When $C = 0.4$, the probability of polygon crossover can be selected in the range 0.5 to 0.6, and the probability of mutation in the range 0.001 to 0.1.
6 CONCLUSIONS

The problem of MWT for a set of points may be NP-hard, so that it is classified as a combinatorial optimization problem. The optimal solution has to be found by a search carried out on an exponential search space. This paper attempts to solve the problem of the minimum weight triangulation of points on a plane using genetic algorithm approach. The applicability of GMWT to the problem of MWT has been investigated by comparing GMWT with the greedy algorithm, which is often used in CAD tools to solve problems of computational geometry. It is shown that GMWT algorithm presented in this paper can get better triangulation of points on a plane than the greedy algorithm. The further extensions of this research work are as follows:

1. Extending this research to a set of 3D points to generate better tetrahedral meshes.
2. Using genetic algorithms to solve the problems of mesh generation of 2D and 3D points for finite element methods by trying out better coding mechanisms, and by devising better genetic operators such as selection, crossover and mutation.

REFERENCES


546