



<b>Title</b>	<b>A genetic algorithm for the minimum weight triangulation</b>
<b>Author(s)</b>	<b>Qin, KH; Wang, WP; Gong, ML</b>
<b>Citation</b>	<b>IEEE International Conference on Evolutionary Computation Proceedings, Indianapolis, IN, 13-16 April 1997, p. 541-546</b>
<b>Issued Date</b>	<b>1997</b>
<b>URL</b>	<b><a href="http://hdl.handle.net/10722/45578">http://hdl.handle.net/10722/45578</a></b>
<b>Rights</b>	<b>Creative Commons: Attribution 3.0 Hong Kong License</b>

# A Genetic Algorithm for the Minimum Weight Triangulation

Kaihuai Qin<sup>a, b</sup>, Wenping Wang<sup>a</sup> and Minglun Gong<sup>b</sup>

## 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[1].

The greedy triangulation and the Delaunay triangulation [1,2,3] are two well-known methods, but it is proved[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[8,9] except the restricted classes of input points whose exact solutions can be found easily[9,10,11].

In this paper, an algorithm for Minimum Weight Triangulation (GMWT for short), which is based on the rationale of genetic algorithms, is presented. The Genetic Algorithm (GA, for short)[13-16] is rooted in the mechanisms of evolution and natural genetics. It is a general-purpose global optimization technique based on randomized search and incorporating some aspects of iterative algorithms, and is regarded as a useful method for solving complex optimization problems, especially combinatory optimization problems and problems whose derivatives can not be computed. The research on genetic algorithms and their applications are actually a very attractive field. In the paper, the principle and algorithms of selection, crossover, mutation and encoding mechanism for triangulations are proposed. The computed results of GMWT, which are compared with greedy triangulations, are given. It is shown that GMWT can obtain better triangulations than the greedy algorithm.

This paper is organized as follows: In Section 2 the

fundamental operators and algorithms for genetic minimum weight triangulations are presented, and the adaptive genetic operators and the structure of GMWT are introduced. In Section 3 the convergence of GMWT is discussed. The computed results are given in Section 4. In Section 5 the parameter selection for GMWT is discussed. Finally, several conclusions are obtained.

This paper assumes that the reader is familiar with genetic algorithms and the greedy algorithm for triangulation.

## 2 GENETIC MINIMUM WEIGHT TRIANGULATION

### 2.1 ENCODING MECHANISM FOR TRIANGULATION

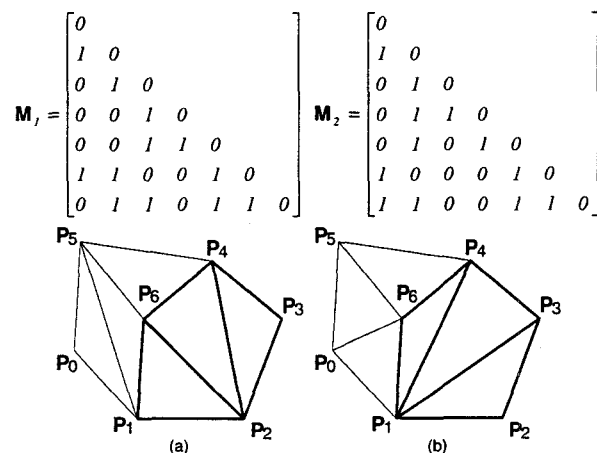


Fig. 1 Triangulations of points

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,  $m_{ij}$  that is the element at row  $i$  and column  $j$  of the matrix is equal to 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  $M_i$  ( $i = 1, 2, \dots$ ) 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.

<sup>a</sup> Dept. of Computer Science, University of Hong Kong, Hong Kong

<sup>b</sup> Dept. of Computer Sci. & Tech., Tsinghua University, Beijing 100084,



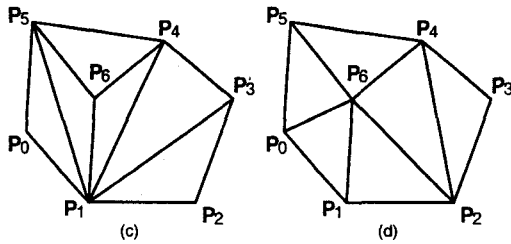


Fig. 2 The offspring: (a), (b) the new strings; (c), (d) triangulations

### 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  $P_i P_j$ . 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  $P_i P_j$  according to the result of  $M_1 XOR M_2$ .
- ii) Construct an "initial polygon" composed of edges  $P_i P_j$  and  $P_j P_i$ . Without loss of generality, assume that line  $P_i P_j$  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.

Fig. 3 Algorithm for polygon crossover

## 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  $M_1$  and the two triangles are linked by the edge  $P_i P_j$  (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  $P_i P_j$  must be replaced by line  $P_k P_s$ , 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.

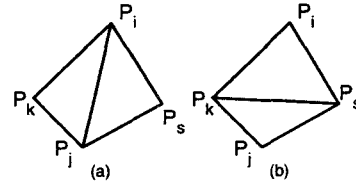


Fig. 4 Legal mutation

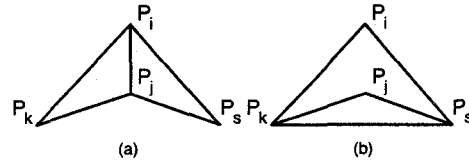


Fig. 5 Illegal mutation

As with the adaptive polygon crossover, an adaptive mutation operator, where the probability of mutation is dynamically determined depending on the fitness values, is used

$$P_{dm} = P_m + C_2 \frac{f_{max} - f}{f_{max} - f_{min}}$$

where  $P_{dm}$  is the dynamic probability of 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

```

    calculate fitness values and perform the select
    operation
    store the offspring
  endfor
  if the termination conditions have been met then stop
i) endwhile

```

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 systems 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 Levkopoulos[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  $(1, 1)$ ;  $\lfloor \sqrt{n} \rfloor$  pulling vertices, with coordinates  $(0, -i)$   $i = 1, 2, \dots, \lfloor \sqrt{n} \rfloor$ ;  $n - \lfloor \sqrt{n} \rfloor - 1$  crowd vertices, lying on the straight-line segment with endpoints  $(3, 3)$  and  $(4, 4)$ . The lowest pulling vertex is called  $P_\ell$  and the highest crowd vertex  $P_u$  (see Fig. 7 (a)).

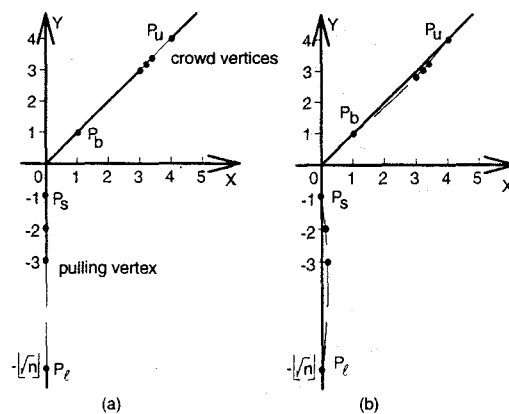


Fig. 7 Point set given by Levkopoulos

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  $P_\ell$ , such that the line passing through  $P_\ell$  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  $P_s$  is a tangent of 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  $P_\ell$  (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  $P_s$ , and all the pulling vertices with the highest crowd vertex  $P_u$  (see Fig. 8 (b)).

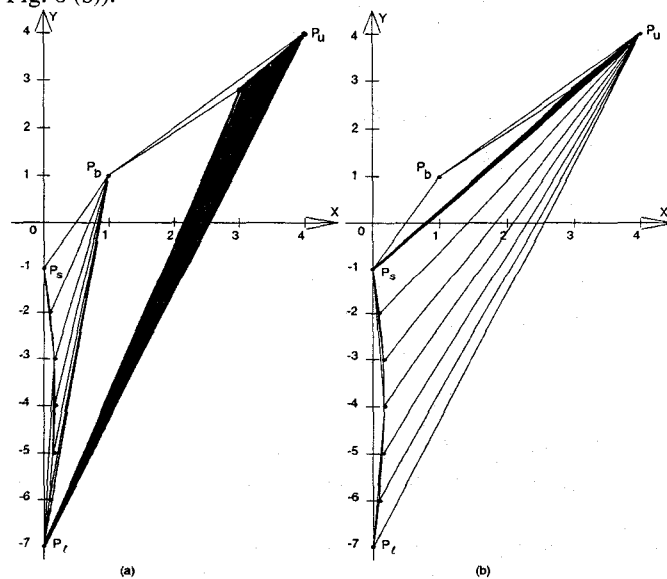


Fig. 8 The result of GT (a) and GMWT (b)

**Example 2:** Triangulations of twelve points.

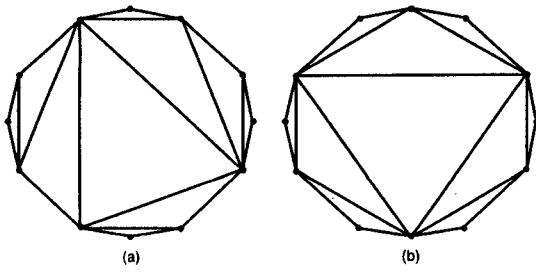


Fig. 9 The results of GT (a) and GMWT (b)

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.

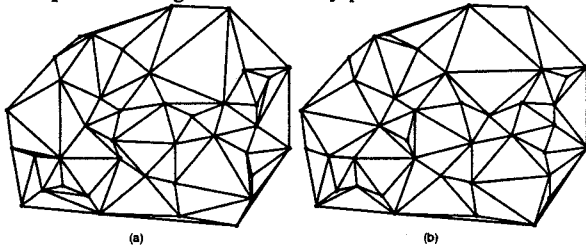


Fig.10 GT (a) and GMWT (b) of point set obtained randomly

The triangulation generated by GT is shown in Fig. 10 (a). Its total length is 4110.3. The best result got by GMWT is shown in Fig. 10 (b), whose total length is 4054.09. Although the improvement of the total length is small, there are many differences between the two triangulations.

The results of GT and GMWT are compared in the following table:

	Example 1		Example 2		Example 3	
	Total Length	%	Total Length	%	Total Length	%
GT	528.21	159.19	1814.19	103.99	4110.3	101.39
GMWT	331.81	100	1744.62	100	4054.09	100

## 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.

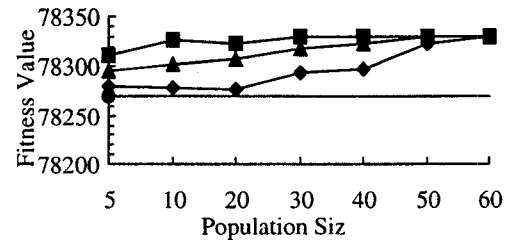


Fig. 11 Effect of population size on performance of GMWT

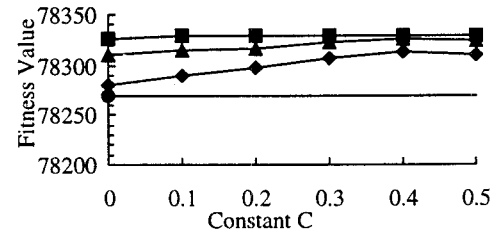


Fig.12 Effect of adaptive constant C on GMWT

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  $C_1 = C_2 \equiv C$  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

1. Lioyd, E.L. On triangulations of a set of points in the plane, *Proc. 18th Ann. IEEE Conf. on the Foundations of Computer Science*, (1977)
2. Preparata, F.P. and Shamos, M.I. *Computational Geometry, An Introduction*, Texts and Monographs in Computer Science, New York, (1985)
3. O'urke, oseph, *Computational Geometry in C*, Cambridge University Press, New York, (1994)
4. Manacher, G.K. and Zobrist, A.L. Neither the Greedy nor the Delaunay triangulation of a planar point set approximates the optimal triangulation. *Inform. Process. Lett.* Vol.9 (1979) pp.31-34
5. Levkopoulos, C. An  $\Omega(\sqrt{n})$  lower bound for the nonoptimality of the Greedy triangulation, *Inform. Process. Lett.* Vol.25 (1987) pp.247-251
6. Lingas, A. The Greedy and Delaunay triangulations are not bad in the average case, *Inform. Process. Lett.* Vol.22 (1986) pp.25-31
7. Kirkpatrick, D.G. A note on Delaunay and optimal triangulations, *Inform. Process. Lett.* Vol.10 (1980) pp.127-128
8. Garey, M.R. and Johnson, D.S. *Computer and Intractability - A Guide to the Theory of NP-Completeness* Freeman, New York, (1979)
9. Meijer, H. & Rappaport, D. Computing the minimum weight triangulation of a set of linearly ordered points, *Inform. Process. Lett.* Vol.27 (1992) pp.35-38
10. Gilbert, P.N. *New results in planar triangulations*, M.Sc. Thesis, Coordinated Science Laboratory, University of Illinois, Urbana, IL, (1979)
11. Klincsek, G.T. Minimal triangulations of polygonal domains, *Ann. Discrete Math.* Vol.9 (1980) pp.121-123
12. Holland, J.H. *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor (1975)
13. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, (1989)
14. Srinivas, M and Patnaik, L.M. Genetic algorithms: A survey, *IEEE Computer*, Vol. (1994) pp.17-26
15. Caponetto, R., Fortuna, L., Graziani, S. & Xibilia, M.G. Genetic algorithms and applications in system engineering: a survey, *Trans. Inst. MC* Vol.15 No.3 (1993) pp.143-155
16. Fogel, David B. An introduction to simulated evolutionary optimization, *IEEE Trans. on Neural Networks* Vol.5 No.1 (1994) pp.3-14
17. Rudolph, G. Convergence analysis of canonical genetic algorithms, *IEEE Trans. on Neural Networks* Vol.5 No.1 (1994) pp.96-101