

# 用混合遗传算法求解两类多边形近似问题

王 斌<sup>1,2</sup>, 施朝健<sup>3,4</sup>

(1. 南京财经大学电子商务重点实验室, 江苏南京 210003; 2. 东南大学计算机科学与工程学院, 江苏南京 210096;  
3. 上海海事大学商船学院, 上海 200135; 4. 复旦大学计算机科学与工程系, 上海 200433)

**摘 要:** 多边形近似是一种重要的曲线描述方法. 研究用遗传算法求解平面数字曲线的多边形近似碰到的两个主要问题是不可行解难以处理和基本遗传算法局部搜索能力差. 针对这两个问题, 本文提出了一种组合拆分与合并技术的混合遗传算法(SMGA). 它将两种经典算法—拆分技术与合并技术引入到对染色体的修复过程. 采用这种方法, 一个不可行解不仅能得到快速的修复, 而且还能被推进到解空间中一个局部较优的位置. 它的另外一个优点是: 不同于已有的遗传算法, 只能解决一类多边形近似问题, SMGA 是一种能求解两类多边形近似问题的通用算法. 实验结果表明: 本文提出的算法比其他同类算法性能更优越.

**关键词:** 曲线描述; 多边形近似; 拆分与合并; 遗传算法; 染色体修复

**中图分类号:** TP391 **文献标识码:** A **文章编号:** 0372-2112 (2007) 08-1527-06

## Using Hybrid Genetic Algorithms to Solve Two Types of Polygonal Approximation Problems

WANG Bin<sup>1,2</sup>, SHI Chao-jian<sup>3,4</sup>

(1. Key Laboratory of Electronic Business, Nanjing University of Finance and Economics, Nanjing, Jiangsu 210003;

2. College of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu 210096;

3. Merchant Marine College, Shanghai Maritime University, Shanghai 200135;

4. Department of Computer Science and Engineering, Fudan University, Shanghai 200433)

**Abstract:** Polygonal approximation of digital curve is an important curve representation method. The main problems existing in genetic algorithm based methods are the difficulty of coping with the infeasible solutions and the poor local search ability. In this paper, we propose a hybrid genetic algorithm combined with split and merge technique (SMGA) for polygonal approximation. Its main idea is that two classical methods—split and merge techniques are used to repair the infeasible solutions. In this method, an infeasible solution can not only be repaired rapidly, but also be pushed to a local optimal location in the solution space. Another advantage of SMGA is that it can solve two types of polygonal approximation problems. The experimental results demonstrate that SMGA is robust and outperforms the other genetic algorithm based methods.

**Key words:** polygonal approximation; split and merge; genetic algorithms; chromosome-repairing

### 1 引言

在图像处理中一个目标的边界可以表示成一条闭合的数字曲线. 如何描述一条曲线是图像处理与模式识别领域中的一个热点问题. 多边形近似是一种有效的曲线描述方法, 其基本思想是用多边形来逼近闭合曲线. 由于多边形近似具有描述简单、结构紧致的特点且能任意精度的近似一条闭合曲线, 在目标识别、图像检索、图像压缩等领域有着广泛的应用.

在现有的文献中, 被大量讨论的两类多边形近似问题是: (1) 给定一条有  $N$  点的闭合的平面数字曲线, 在所有的近似该曲线的且有  $M$  个顶点的多边形中(顶点

在曲线上), 找一个具有最小近似误差的多边形. (2) 给定一条有  $N$  个点的闭合的平面数字曲线和容忍近似误差  $\epsilon$ . 在所有的近似该曲线且近似误差在容忍误差范围内的多边形(顶点在曲线上)中, 找一个具有最小顶点数的多边形. 这两类多边形问题都可归属于组合优化问题. 它们在优化目标和约束条件上互不相同. 用常规的方法求解这两类多边形近似问题的最优解需要很高的计算复杂度<sup>[1]</sup>. 现有的方法大都着眼于求局部最优解以减少计算上的开销. 这些方法中, 经典的算法有顺序跟踪法<sup>[2]</sup>、拆分法<sup>[3]</sup>、合并法<sup>[4]</sup>、拆分与合并法<sup>[5]</sup>以及关键点检测法<sup>[6]</sup>. 这些方法属于局部的优化方法, 其优点是计算速度快, 但由于对最初选定的起始点或初始产生的

解的依赖性,其最终得到的解有可能远离全局最优解<sup>[7]</sup>.近年来,一些基于全局优化的方法,如遗传算法<sup>[1,8,9,11,12]</sup>、禁忌搜索算法<sup>[7]</sup>、蚁群算法<sup>[13]</sup>以及微粒群算法<sup>[14]</sup>等用于求解第一类或第二类多边形近似问题并取得了较好的近似效果.但这些方法都存在着收敛速度慢,局部优化能力差的问题.而且这些方法是针对两类多边形近似问题中的某一类提出的,对另一类问题往往是不适用的.本文着眼于研究一种快速而有效的能求解两类多边形近似问题的通用的遗传算法.

跟求解其它组合优化问题一样,用遗传算法求解多边形近似问题碰到的一个障碍是:如何处理因遗传操作所产生的不可行解,对这个问题处理是否得当,将直接影响到算法的性能.在目前的求解多边形近似问题的遗传算法中,普遍采用两类方法来处理这个问题.一类方法是惩罚函数法,其主要思想是:对评估函数附加一个罚函数,使得不可行解具有较低的生存概率.文献[1]采用了这种方法求解了第二类多边形近似问题.这种方法的缺点是:定义一个合适的惩罚函数往往是很困难的,惩罚的强度也很难控制,如果惩罚过弱,则最后得到的可能是不可行解,而惩罚过强将限制算法所能到达的搜索区域,从而影响求解的质量.另一类方法是增加杂交算子的复杂性的方法.文献[8,9,11]采用了通过增加传统的两点杂交算子的复杂性的方法来维持产生的解的可行性.传统的两点杂交算子(two-cut-point crossover)的两个杂交点是随机选取的.为维持后代的可行性,必须在染色体上选取合适的杂交点,这需要反复测试染色体上的候选杂交点.这种方法虽然不用设计专门的惩罚函数,但存在的问题是:算法复杂度较高而且限制了传统的两点杂交算子的灵活性.另外这种杂交点的选取不是总能成功的,如对第二类多边形近似问题.文献[12]将采用正交数组算子(Orthogonal array crossover)的遗传算法应用于求解第一类多边形近似问题.正交数组算子是一种杂交算子,它总能得到可行解.但它的计算的开销太大,而且对第二类多边形近似问题不适用.

本文提出了一种组合拆分与合并技术的混合遗传算法(SMGA)来求解平面数字曲线的多边形近似.它采用染色体修复策略来处理不可行解,并将传统的拆分与合并技术引入到对染色体的修复过程.这种方法简单快速,既克服了第一类方法难以设计合适的惩罚函数的问题,又克服了第二类方法计算复杂度高,灵活性差的问题.而且这种方法还借用了传统的拆分与合并技术局部优化能力强的优点.此外,它是一种能求解两类多边形近似问题的通用算法.通过大量的实验和跟相关工作的对比,证明了本文提出的算法的优越性.

## 2 问题描述

**定义 1** 一个闭合的平面数字曲线可以表示为一个沿顺时针方向的点的有序集  $C = \{p_1, p_2, \dots, p_N\}$ . 其中,  $p_{i+1}$  是  $p_i$  的后继点,  $p_1$  是  $p_N$  的后继点,  $N$  是平面数字曲线上的点的个数.

**定义 2** 从平面数字曲线的  $p_i$  点出发,沿曲线按顺时针方向到达曲线上的另外一点  $p_j$  所经过的弧用  $\overline{p_i p_j} = \{p_i, p_{i+1}, \dots, p_j\}$  表示. 该段弧所对应的弦是一个连接点  $p_i$  和  $p_j$  的直线段,表示  $\overline{p_i p_j}$ .

**定义 3** 用弦  $\overline{p_i p_j}$  近似其对应的弧  $\overline{p_i p_j}$  的近似误差定义为:  $e(\overline{p_i p_j}, \overline{p_i p_j}) = \sum_{p_k \in \overline{p_i p_j}} d^2(p_k, \overline{p_i p_j})$ , 这里  $d(p_k, \overline{p_i p_j})$  表示点  $p_k$  到  $\overline{p_i p_j}$  的投影距离.

**定义 4** 平面数字曲线  $C = \{p_1, p_2, \dots, p_N\}$  的多边形近似是一个直线段的有序集  $V = \{\overline{p_{t_1} p_{t_2}}, \dots, \overline{p_{t_{M-1}} p_{t_M}}, \overline{p_{t_M} p_{t_{M+1}}}\}$ , 这里  $t_i \in \{1, 2, \dots, N\}$ ,  $t_1 < t_2 < \dots < t_m$ ,  $t_{M+1} = t_1$  并且  $M$  是多边形的顶点的个数.

**定义 5** 平面数字曲线  $C$  和它的一个多边形近似  $V = \{\overline{p_{t_1} p_{t_2}}, \dots, \overline{p_{t_{M-1}} p_{t_M}}, \overline{p_{t_M} p_{t_{M+1}}}\}$  间的近似误差定义为:

$$E(V, C) = \sum_{i=1}^M e(\overline{p_{t_i} p_{t_{i+1}}}, \overline{p_{t_i} p_{t_{i+1}}}).$$

那么,两类多边形近似问题可以定义为:

**第一类问题:** 给定一个平面数字曲线  $C = \{p_1, p_2, \dots, p_N\}$  和多边形的边数  $M$ , 这里  $3 \leq M \leq N$ . 设  $\Omega$  表示所有近似平面数字曲线  $C$  的多边形的集合,  $SP = \{V | V \in \Omega \wedge |V| = M\}$ , 这里  $|V|$  表示集合  $V$  的势. 找一个多边形  $P \in SP$  满足  $E(P, C) = \min_{V \in SP} E(V, C)$ .

**第二类问题:** 给定一个平面数字曲线  $C = \{p_1, p_2, \dots, p_N\}$  和容忍近似误差  $\varepsilon$ , 设  $\Omega$  表示所有近似平面数字曲线  $C$  的多边形的集合,  $SP = \{V | V \in \Omega \wedge E(V, C) \leq \varepsilon\}$ , 找一个多边形  $P \in SP$  满足  $|P| = \min_{V \in SP} |V|$ , 这里  $|V|$  表示集合  $V$  的势.

## 3 组合拆分与合并技术的混合遗传算法(SMGA)

### 3.1 染色体编码与适应值评估

因为一条数字曲线可以表示为一个点的有序集,所以采用传统的二进制编码是一个自然的选择.采用这种编码,染色体上的基因将与曲线上的点一一对应,当基因的值为 1 时,表示其对应的曲线上的点被选为多边形的顶点.染色体上基因值为 1 的基因的个数等于多边形的顶点数.例如,设一条数字曲线  $C = \{p_1, p_2, \dots, p_{10}\}$ , 给定一串二进制编码 '1010100010'. 则它表示的多边形是  $\{\overline{p_1 p_3}, \overline{p_3 p_5}, \overline{p_5 p_9}, \overline{p_9 p_1}\}$ .

设一条数字曲线  $C = \{p_1, p_2, \dots, p_N\}$ ,  $a = b_1 b_2 \dots b_N$

为近似曲线  $C$  的一个多边形的染色体编码. 根据两类多边形近似问题的定义, 其适应值  $f(\alpha)$  对于第一类问题定义为:  $f(\alpha) = E(\alpha, C)$ . 对于第二类问题定义为:

$$f(\alpha) = \sum_{i=1}^n b_i.$$

### 3.2 基因平移变异与两点杂交算子

变异算子和杂交算子是遗传算法的重要组成部分. 对于变异算子的设计, 针对问题的特点, 我们设计了基因平移算子. 其操作过程是: 首先随机的选取一个基因值为 1 的基因, 然后随机的将其左移或右移一个位置, 原来位置的基因值设置为 0. 基因平移变异算子的主要功能是将当前的多边形的一个顶点移动到它在曲线上的相邻点上.

我们采用传统的两点杂交算子. 其操作过程是: 随机选取两个杂交点的位置, 然后将两个父体在两个杂交点间的子串互换. 例如, 两个父串 '1010010101' 与 '1011001010', 随机选取的杂交点为 4 和 7, 则杂交后产生的后代为 '1011001101' 和 '1010010010' (如图 1 所示).

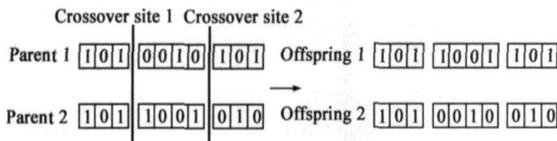


图 1 两点杂交算子

### 3.3 选择策略与最优个体保留策略

这里使用锦标赛 (tournament) 选择策略. 其操作过程是: 随机的在群体中选择 2 个个体, 其中适应值高的个体将被选择成为生成下一代的父体. 为了保护每一代的最优个体不被变异和杂交操作所破坏, 我们采用传统的最优个体保留策略, 即适应值最高的个体不进行配对交叉而直接复制到下一代.

### 3.4 染色体修复策略

#### 3.4.1 拆分技术与合并技术

传统的拆分法是一种非常简单的求解闭合曲线多边形近似的方法, 其作法是: 首先产生闭合曲线的一个初始分割, 将闭合曲线分为两个弧段, 然后反复使用拆分操作, 将曲线拆分为越来越小的弧段, 直到顺序连结割点所得到的多边形满足给定的约束条件算法终止. 其中拆分操作的操作过程如下:

设曲线  $C$  当前已被分割成  $M$  个弧段:  $\overline{p_{i_1} p_{i_2}}, \dots, \overline{p_{i_{M-1}} p_{i_M}}, p_{i_M} p_{i_1}$ , 这里  $p_{i_j}$  是割点. 则对曲线  $C$  的一次拆分操作为: 对曲线上的每一个非割点的点  $p_i \in \overline{p_{i_j} p_{i_{j+1}}}$ , 计算其到对应的弦的投影距离  $D(p_i) = d(p_i, \overline{p_{i_j} p_{i_{j+1}}})$ . 在曲线  $C$  上找一个点  $p_u$  满足  $D(p_u) = \max_{p_i \in C} D(p_i)$ , 设其所在的弧段为  $\overline{p_{i_j} p_{i_{j+1}}}$ , 然后将弧段  $\overline{p_{i_j} p_{i_{j+1}}}$  在点  $p_u$  上拆分

成两个弧段  $\overline{p_{i_j} p_{i_u}}$  和  $\overline{p_{i_u} p_{i_{j+1}}}$ , 点  $p_u$  将被加入到割点集合中. 也就是说拆分操作给当前的多边形增加了一个顶点. 拆分操作的主要功能是利用启发式的方法在曲线上寻找新的可能作为多边形顶点的点.

传统的合并法的主要思想是: 初始认为曲线上所有的点都是多边形的顶点, 即所有的点都是割点. 然后反复执行合并操作, 合并两个相邻的弧段以消除一个割点, 直到最后连接所有的割点所构成的多边形满足给定的约束条件. 这里合并操作是拆分操作的逆操作. 其操作过程如下:

设曲线  $C$  当前已被分割成  $M$  个弧段:  $\overline{p_{i_1} p_{i_2}}, \dots, \overline{p_{i_{M-1}} p_{i_M}}, p_{i_M} p_{i_1}$ , 这里  $p_{i_j}$  是割点. 则对曲线  $C$  的一次合并操作为: 对每一个割点  $p_{i_j}$ , 计算其到连接它的相邻的两个割点的直线段的距离  $Q(p_{i_j}) = d(p_{i_j}, \overline{p_{i_{j-1}} p_{i_{j+1}}})$ . 在当前所有的割点中找一个割点  $p_{i_j}$  满足  $Q(p_{i_j}) = \min_{p_{i_j}} Q(p_{i_j})$ , 这里  $V$  是当前所有割点的集合. 然后将两个弧段  $\overline{p_{i_{j-1}} p_{i_j}}$  和  $\overline{p_{i_j} p_{i_{j+1}}}$  合并成一个弧段  $\overline{p_{i_{j-1}} p_{i_{j+1}}}$ . 割点  $p_{i_j}$  将被从割点集合中去掉. 也就是说合并操作将去掉当前多边形的一个顶点. 合并操作的主要功能是利用启发式的方法尽可能的去掉当前多边形上的冗余的顶点.

#### 3.4.2 基于拆分与合并技术的染色体修复算子

可行解经过杂交或变异操作有可能产生不可行解. 这里我们设计了一个染色体修复算子, 将拆分技术与合并技术应用到对染色体的修复过程. 由于两类多边形近似问题的约束条件不同, 所以对不可行解的处理略有不同.

对第一类多边形近似问题, 设给定的多边形的边数为  $M$ , 则一个不可行解  $\alpha$  必定是一个边数大于或小于  $M$  的近似多边形, 即  $L(\alpha) \neq M$ , 这里  $L(\alpha)$  表示近似多边形  $\alpha$  的边数. 则对第一类问题的不可行解  $\alpha$  的修复操作如下: 如果  $L(\alpha) > M$ , 则反复执行合并操作直到  $L(\alpha) = M$ , 如果  $E(\alpha) < M$ , 则反复执行拆分操作直到  $L(\alpha) = M$ .

对第二类多边形近似问题, 设给定的容忍近似误差为  $\varepsilon$ , 则一个不可行解  $\alpha$  必定是一个近似误差大于  $\varepsilon$  的近似多边形, 即  $E(\alpha) > \varepsilon$ . 则对第二类问题的不可行解  $\alpha$  的修复操作为: 如果  $E(\alpha) > \varepsilon$ , 则反复执行拆分操作直到  $E(\alpha) \leq \varepsilon$ .

值得注意的是, 对于第二类多边形近似问题, 一个不可行的处理只需要执行拆分操作, 没有用到合并操作, 这是与第一类多边形近似问题在处理不可行解的不同的地方. 但合并操作是一个很重要的操作, 它利用启发式的方法, 尽可能的去掉多边形的冗余的顶点. 所以对第二类多边形近似问题, 对每一个可行解  $\alpha$ , 执行

一个优化操作. 其操作过程如下: 对操作“保存当前解  $\alpha$  到  $\beta$ , 然后对  $\alpha$  执行合并操作”反复执行直到  $E(\alpha) > \varepsilon$ , 置  $\beta$  到  $\alpha$ .

### 3.5 通用的算法框架

下面给出用混合遗传算法求解两类多边形近似问题的通用框架. 算法的参数为群体规模  $N_s$ 、杂交概率  $p_m$ 、变异概率  $p_c$  和迭代次数  $G_n$ . 输入为平面数字曲线  $C = \{p_1, p_2, \dots, p_N\}$ , 容忍误差  $\varepsilon$  或多边形的边数  $M$ . 输出为曲线  $C$  的多边形近似.

**Step1** 随机的产生  $N_s$  个个体构成一个初始群体, 其中每个个体的产生方式是: 随机产生 1 的个数大于 2 并且小于等于  $N$  的二进制串. 迭代计数器  $g$  置 0.

**Step2** 对当前群体中的每一个不可行解执行染色体修复操作. (对第二类多边形近似问题的每一个可行解还另外附加一个基于合并技术的优化操作).

**Step3** 计算每个个体的适应值, 并将当前群体适应值最大的个体直接复制到下一代.

**Step4** 反复执行选择操作在当前群体中选择父体, 并依据杂交概率和变异概率分别执行杂交操作和变异操作, 产生下一代群体的  $N_s - 1$  个体.

**Step5** 迭代计数器加 1, 如果  $g \leq G_n$ , 转 *Step2*.

**Step6** 输出当前群体中最优个体.

## 4 实验结果和讨论

为了评估本文提出的算法 SMGA 的性能. 第一组实验是用被广泛采用的四条 benchmark 曲线来测试算法的性能. 第二组实验是将算法实际应用于两个湖泊地图轮廓线的多边形近似. 为证明 SMGA 算法的优越, 我们将近几年来, 为求解第一类多边形近似问题提出的

遗传算法 Yin<sup>[8]</sup>、Sun<sup>[9]</sup>、Sarkar<sup>[10]</sup> 和 Ho<sup>[12]</sup>, 以及为求解第二类多边形近似问题提出的遗传算法 GA<sup>[11]</sup>, 一一编程实现以资对比. 由于相对于求解第一类多边形近似问题, 求解第二类多边形近似问题在处理不可行解的问题上更加困难, 所以, 针对第二类多边形近似问题, 提出的遗传算法较少. 所以, 我们引用了近几年提出的求解第二类多边形近似问题的禁忌搜索算法 TS<sup>[7]</sup> 和蚁群算法 ACO<sup>[13]</sup> 用 benchmark 曲线作测试的实验结果, 以资和 SMGA 算法作更多的对比. 实验平台是一台 CPU 为 AMD Athlon XP 3000+, 操作系统为 Windows XP 计算机, 编程工具为 Borland Delphi 7.

### 4.1 实验一: 用 benchmark 曲线测试算法的性能

图 2 给出了 4 条 benchmark 曲线, 文献[7]给出了这四条曲线的链码. 它们的轮廓线上的点的个数分别为: 45、60、102 和 120. 这里 SMGA 算法的参数设置为: 群体规模  $N_s = 30$ 、迭代次数  $G_n = 80$ 、杂交概率  $p_m = 0.7$  和变异概率  $p_c = 0.3$ . 其他用来比较的 5 种遗传算法的参数按原文献中的设置.

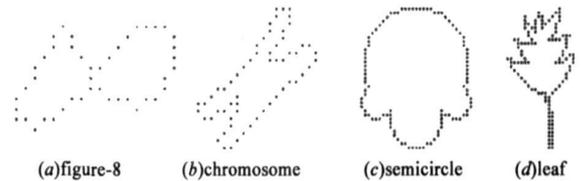


图 2 四条 benchmark 曲线

表 1 给出了算法 SMGA 和其它四种遗传算法 Yin<sup>[8]</sup>、Sun<sup>[9]</sup>、Sarkar<sup>[10]</sup> 和 Ho<sup>[12]</sup> 用图 2 中的 4 条曲线作测试实例, 求解第一类多边形近似问题的实验结果. 这里  $M$  是给定的边数, AVERAGE  $\varepsilon$  是 10 次算法独立运行的所得到的多边形的近似误差的平均, AVERAGE  $t$  则

表 1 用算法 SMGA 和其他四种遗传算法求解第一类多边形近似问题的实验结果的对比

Curves	$M$	AVERAGE $\varepsilon$					AVERAGE $t$ (s)				
		Yin <sup>[8]</sup>	Sun <sup>[9]</sup>	Sarkar <sup>[10]</sup>	Ho <sup>[12]</sup>	SMGA	Yin <sup>[8]</sup>	Sun <sup>[9]</sup>	Sarkar <sup>[10]</sup>	Ho <sup>[12]</sup>	SMGA
Semicircle $N = 102$	16	24.44	15.35	15.72	14.65	14.30	0.13	0.19	0.12	0.25	0.02
	22	11.94	9.26	8.28	8.34	8.02	0.12	0.17	0.12	0.23	0.02
	27	8.34	5.90	5.28	4.86	4.34	0.11	0.16	0.12	0.23	0.02
	30	6.67	4.48	4.41	3.45	3.05	0.11	0.15	0.12	0.22	0.02
Figure-8 $N = 45$	6	17.93	20.45	18.92	17.85	17.70	0.07	0.12	0.07	0.13	0.01
	10	5.10	4.22	4.00	3.75	3.72	0.06	0.09	0.07	0.12	0.01
	13	2.61	2.47	2.24	2.38	2.09	0.06	0.09	0.06	0.12	0.01
	16	1.55	1.84	1.57	1.58	1.47	0.06	0.08	0.06	0.11	0.02
chromosome $N = 60$	8	17.78	13.57	16.04	13.92	13.43	0.08	0.15	0.09	0.17	0.01
	14	5.85	4.84	4.87	4.75	4.39	0.08	0.12	0.08	0.16	0.02
	17	4.72	3.88	3.31	3.28	3.23	0.08	0.11	0.08	0.15	0.02
	18	3.75	3.22	3.13	3.02	3.01	0.08	0.11	0.08	0.15	0.02
leaf $N = 120$	17	78.89	56.34	32.84	32.89	25.78	0.14	0.21	0.15	0.30	0.02
	21	39.27	27.11	15.36	17.21	13.38	0.14	0.20	0.15	0.29	0.02
	26	24.04	13.05	8.83	8.43	8.08	0.14	0.19	0.15	0.28	0.02
	29	15.67	8.94	7.04	6.53	6.47	0.14	0.18	0.14	0.23	0.03

表 2 用算法 SMGA 和遗传算法 GA<sup>[1]</sup>、禁忌搜索 TS<sup>[7]</sup> 和蚁群算法 ACO<sup>[13]</sup> 求解第二类多边形形状近似问题的实验结果的对比(其中 TS 和 ACO 的实验结果直接引自文献[7, 13])

Curves	$\epsilon$	AVERAGE $M$				AVERAGE $t(s)$			
		GA <sup>[1]</sup>	TS <sup>[7]</sup>	ACO <sup>[13]</sup>	SMGA	GA <sup>[1]</sup>	TS <sup>[7]</sup>	ACO <sup>[13]</sup>	SMGA
Semicircle $N = 102$	60	13.3	11.0	10.0	10.0	0.13	~	~	0.02
	30	16.10	13.6	12.6	12.0	0.13	~	~	0.02
	20	18.8	16.2	16.4	14.0	0.13	~	~	0.02
	15	22.0	18.3	18.0	15.1	0.13	~	~	0.03
chromosome $N = 60$	30	8.5	6.7	6.0	6.0	0.08	~	~	0.02
	20	10.1	8.0	8.0	7.0	0.08	~	~	0.02
	10	12.0	11.0	10.0	10.0	0.08	~	~	0.02
	6	15.8	14.4	12.8	12.1	0.08	~	~	0.02
leaf $N = 120$	150	12.6	10.6	11.2	9.70	0.15	~	~	0.02
	100	15.0	13.7	13.0	12.0	0.15	~	~	0.02
	30	22.3	20.1	17.2	16.0	0.16	~	~	0.02
	15	27.5	23.1	22.2	20.2	0.15	~	~	0.03

是计算时间的平均。从表 1 我们可以看出, 算法 SMGA 的平均解要优于其它四种算法, 而且计算速度最快。表 2 给出了算法 SMGA、禁忌搜索算法 TS<sup>[7]</sup>、蚁群算法 ACO<sup>[13]</sup> 以及遗传算法 GA<sup>[1]</sup> 用图 2 中的曲线 chromosome、semicircle 和 leaf 作测试实例, 求解第二类多边形近似问题的实验结果, 这里是给定的容忍误差。表中 TS<sup>[7]</sup> 和蚁群算法 ACO<sup>[13]</sup> 的实验数据直接引自于原文献。从表 2 可以看出, 算法 SMGA 在给定的相同的容忍误差下, 近似曲线所用的边数比其它三种算法 TS<sup>[7]</sup>、ACO<sup>[13]</sup> 以及 GA<sup>[1]</sup> 要少。计算时间也比同类算法 GA<sup>[1]</sup> 少得多。

#### 4.2 应用于湖泊地图轮廓线的多边形近似

目前大多数求解多边形近似问题的方法仅用图 2 中的四条 benchmark 曲线测试其性能。其缺点是: 测试数据的规模太小, 一般仅 100 个点左右, 而且是人为设计的。为了进一步证明 SMGA 算法在大规模测试实例上的有效性, 我们将 SMGA 算法实际应用于湖泊地图轮廓线的多边形近似。图 3 中的 (a) 和 (c) 分别是澳大利亚的两个著名的湖泊 Gairdner 和 Disappointment, 它们分别位于澳大利亚的南部和西部。(b) 和 (d) 是从地图中抽取的轮廓线。轮廓线上的点的个数分别为 1007 和 1330。因为数据的规模较大, 我们将 SMGA 算法的参数调整为: 群体规模、迭代次数。其它五种遗传算法 Yin<sup>[8]</sup>、Sun<sup>[9]</sup>、Sarkar<sup>[10]</sup>、Ho<sup>[12]</sup> 以及 GA<sup>[1]</sup> 的参数皆设为: 群体规模为 100, 迭代次数为 200。

对于两个湖泊轮廓线, 给定边数为 60, 进行第一类多边形近似。让五种算法独立运行 10 次, 图 4 给出了五种算法在 10 次独立运行中所得到的最好解 ( $\epsilon$  是近似误差,  $t$  是计算时间)。从图 4 可以看出, 算法 SMGA 给出的解的近似效果最好, 在给定相同边数的情况下, 相对于其它算法, SMGA 得到的近似多边形对湖泊轮廓线的

描述更加精确, 能描述湖泊轮廓线更多的细节, 而且所用的时间最节省。

图 5 则是用算法 SMGA 和算法 GA<sup>[1]</sup> 对两个湖泊轮廓线进行第二类多边形近似的实验结果。这里对湖泊 Gairdner 给定的容忍误差为 500, 对湖泊 Disappointment 给定的容忍误差为 2000。从该图看出, 给定相同的容忍误差, 算法 GA<sup>[1]</sup> 得到的多边形近似湖泊轮廓线需要更多的边数, 而且计算时间超过了 SMGA 算法的 10 倍。

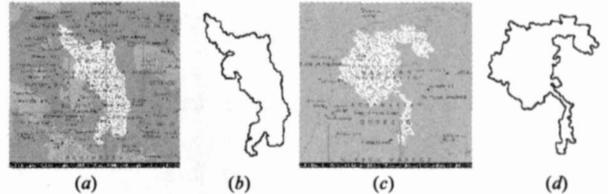


图 3 两个湖泊 Gairdner 和 Disappointment 的地图和从中抽取的湖泊的轮廓线 (分别为 1007 个点和 1330 个点)

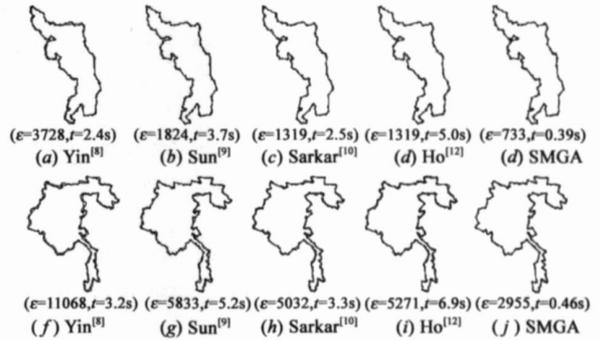


图 4 对湖泊 Gairdner 和 Disappointment 的轮廓线给定边数皆为 60, SMGA 和其他四种遗传算法经 10 次独立运行所得到的最好解 ( $\epsilon$  是近似误差,  $t$  是计算所用的时间)

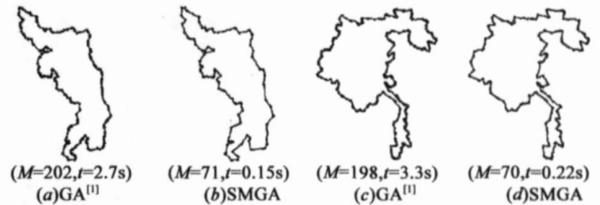


图 5 对湖泊 Gairdner 和 Disappointment 的轮廓线分别给定容忍误差为 500 和 2000, 算法 SMGA 和 GA<sup>[1]</sup> 10 次独立运行所得到的最好解 ( $M$  是边,  $t$  是计算所用的时间)

#### 5 结论

本文提出了一种组合拆分与合并技术的混合遗传算法 (SMGA) 来求解平面数字曲线的两类多边形近似, 其核心思想是将拆分与合并技术用到对非法染色体的修复中。采用这种方法, 克服了在用遗传算法求解多边形近似问题时, 不可行解难以处理的问题和传统遗传算法的局部优化能力差、收敛速度慢的问题。实验结果表明: 本文提出的算法 SMGA 无论是在求解的质量, 还是在计算速度上都要明显优于其它同类算法, 而且对两类多边形近似问题都能进行求解。进一步的研究工

作将是:一方面拓展该算法的实际应用领域,如应用于基于植物叶形的识别和检索等,另一方面将该算法由二维曲线拓展到三维曲线。

#### 参考文献:

- [1] P Y Yin. Genetic algorithms for polygonal approximation of digital curves[J]. Int J Pattern Recognition Artif Intell, 1999, 13(7): 1061- 1082.
- [2] Sklansky J, Gonzalez V. Fast polygonal approximation of digitized curves[J]. Pattern Recognition [J]. 1980, 12(2): 327- 331.
- [3] Douglas D H, Peucker T K. Algorithm for the reduction of the number of points required to represent a line or its caricature [J]. The Canadian Cartographer, 1973, 10(2): 112- 122.
- [4] Leu J G, Chen L. Polygonal approximation of 2D shapes through boundary merging [J]. Pattern Recognition Letters, 1988, 7(4): 231- 238.
- [5] B K Ray, K S Ray. A new split-and-merge technique for polygonal approximation of chain coded curves [J]. Pattern Recognition Letter, 1995, 16(2): 161- 169.
- [6] Teh H C, Chin R T. On detection of dominant points on digital curves[J]. IEEE Trans Pattern Anal Mach Intell, 1989, 11(8): 859- 872.
- [7] P Y Yin. A tabu search approach to the polygonal approximation of digital curves[J]. Int J Pattern Recognition Artif Intell, 2000, 14(2): 243- 255.
- [8] P Y Yin. A new method for polygonal approximation using genetic algorithm [J]. Pattern Recognition letter, 1998, 19(11): 1017- 1026.
- [9] Huang S-C, Sun Y-N. Polygonal approximation using genetic algorithms[J]. Pattern Recognition, 1999, 32(8): 1409- 1420.
- [10] Bistwajit Sarker, Lokendra Kumar Singh, Debranjana Sarkar. A genetic algorithm-based approach for detection of significant vertices for polygonal approximation of digital curves[J]. International Journal of Image and Graphics, 2004, 4(2): 223- 239.
- [11] Sun Y-N, Huang S-C. Genetic algorithms for error-bounded polygonal approximation [J]. Int J Pattern Recognition and Artificial Intelligence, 2000, 14(3): 297- 314.
- [12] Ho S-Y, Chen Y-C. An efficient evolutionary algorithm for accurate polygonal approximation [J]. Pattern Recognition, 2001, 34(12): 2305- 2317.
- [13] P Y Yin. An ant colony search algorithms for optimal polygonal approximation of plane curves [J]. Pattern Recognition, 2003, 23(8): 1783- 1797.
- [14] P Y Yin. A discrete particle swarm algorithm for optimal polygonal approximation of digital curves [J]. Journal of visual communication and image representation, 2004, 15(2): 241 - 260.

#### 作者简介:



王 斌 男, 博士, 1969 年生, 博士后研究人员. 研究方向为智能计算、模式识别、图像处理. E-mail: wangbin\_cs@fudan.edu.cn

施朝健 男, 1957 年生, 教授, 博士研究生, 研究方向为图像处理、交通信息工程及控制.