

文章编号: 2095-2163(2021)12-0109-07

中图分类号: TP391

文献标志码: A

基于改进遗传算法的图像分割

胡承刚, 高建瓴, 喻明豪, 白羽飞, 陈楠

(贵州大学 大数据与信息工程学院, 贵阳 550025)

摘要: 针对大律法(Otsu)分割图像存在分割效率不高、抗噪性能不佳的缺点,提出一种改进的遗传算法,并对大律法阈值分割算法优化。改进的遗传算法主要有两个方面:首先,结合种群密度和算法进化代数的协同性来考虑改进算法的交叉概率,使算法自适应调节交叉概率。其次,计算种群中心区域密度,并结合算法进化代数对变异概率做出自适应调整,进化过程种群相对稳定时加入“天灾”设定,破坏种群结构,防止算法因为种群内部相对稳定而陷入局部最优解。经过实验,本文所提算法在收敛速度上比参考文献中改进的遗传算法和自适应遗传算法快,陷入局部最优解次数少。经过图像分割实验验证,本文算法分割效率高,抗噪性能更好。

关键词: 大律法阈值分割; 种群密度; 种群结构; 图像分割

Image segmentation based on improved genetic algorithms

HU Chenggang, GAO Jianling, YU Minghao, BAI Yufei, CHEN Nan

(School of Big Data and Information Engineering, Guizhou university, Guiyang 550025, China)

【Abstract】 Aiming at the shortcomings of the Otsu segmentation method that the segmentation efficiency is not high and the anti-noise performance is not good, an improved genetic algorithm is proposed and the Otsu threshold segmentation algorithm is optimized. The improved genetic algorithm mainly has two advantages. On the one hand, the crossover probability of the improved algorithm is considered by combining the population density and the synergy of the algorithm evolution algebra, so that the algorithm can adjust the crossover probability adaptively. On the other hand, the proposed method calculates the density of the population center area, and makes adaptive adjustments to the mutation probability in combination with the evolutionary algebra of the algorithm. When the population is relatively stable during the evolution process, the natural disaster setting is added to destroy the population structure and prevent the algorithm from falling into a local maximum due to the relative stability of the population. The experiments show that the algorithm proposed in this paper is faster than the standard genetic algorithm and the adaptive genetic algorithm in the convergence speed, and has fewer steps before getting into the local optimal solution. The image segmentation experiment verifies that this algorithm has high segmentation efficiency and better anti-noise performance.

【Key words】 Otsu threshold split; population density; population structure; image segmentation

0 引言

遗传算法最早由美国 Michigan 大学的教授 John Holland 模拟生物遗传和进化提出^[1]。二十世纪 80 年代,De Jong 基于遗传算法的思想在计算机上进行数值优化实验,归纳总结形成其基本框架。遗传算法通过将目标函数的自变量映射为生物遗传进化中的染色体,对染色体编码来实现向下进化,并找到使目标函数最优的染色体;同时,遗传过程中加入了交叉和变异操作,来优化下一代染色体。遗传算法中染色体的编码方式有浮点数编码和二进制编码两种方式。不同的染色体编码方式和采取的变异

算子、交叉算子的差异同样造成了遗传算法不同^[2]。由于遗传算法是一种全概率搜索优化算法,所以其对于目标函数的性质没有要求,比如:可微、连续等,所以在很多实际问题中都有应用,比如:数值函数寻优、多目标优化、机器学习、图像处理,最优路径规划、神经网络的训练等。

图像分割是区分图像中的对象和背景的过程。对于许多依赖于计算机视觉的应用,如医学成像、卫星图像中的物体定位、机器视觉、手指打印和人脸识别等许多应用来说,图像分割是一项必不可少的预处理任务。图像分割的准确性将对图像处理的后续阶段的有效性产生很大的影响。图像分割问题已被

作者简介: 胡承刚(1996-),男,硕士研究生,主要研究方向:机器学习、文本处理;高建瓴(1969-),女,硕士,副教授,硕士生导师,主要研究方向:数据挖掘、数据库应用;喻明豪(1997-),男,硕士研究生,主要研究方向:机器学习、图像处理;白羽飞(1996-),男,硕士研究生,主要研究方向:文本处理;陈楠(1997-),女,硕士研究生,主要研究方向:机器学习、图像处理。

通讯作者: 高建瓴 Email:454965711@qq.com

收稿日期: 2021-10-22

研究数年,但由于图像的不同模态直方图等特征,图像分割问题仍是一个有待进一步研究的问题。基于深度学习的图像分割虽然有不错的效果,但是需要大量标注数据来训练神经网络;基于阈值分割的Otsu是经典的图像分割算法,在单阈值分割时算法效率很高,但在多阈值分割时算法的速度极低,这是由于算法会遍历图像的每个像素点,计算类内方差和类间方差,算法速度慢;基于遗传算法的阈值分割可以解决多阈值图像分割效率低的问题。

本文分析遗传算法及其改进算法的性能,提出一种混合改进的遗传算法,并将改进的算法先用于数值函数寻优,以测试比较改进算法的性能;对图像阈值分割进行分析,找到最佳的阈值来分割图像,OTSU算法遍历像素的操作在多阈值分割时性能下降严重,通过对阈值分割算法分析,使用遗传算法来寻找最佳阈值,以分割图像。

1 混合改进的遗传算法

1.1 自适应遗传算法模型

遗传算法在多个领域的研究表明其高效性,其性能取决于在迭代的最后是否能够在函数优化中收敛到局部或者全局最优解。标准遗传算法(SGA)采用固定控制参数方法,即固定交叉概率和变异概率,缺点明显,因为前后期适应度不同,所以需要的交叉、变异概率也不同。赵大兴等人提出基于适应度对交叉、变异概率作出改进,改进的选择算子会根据个体适应度之间的相似性把种群分为若干组,以组为单位进行轮盘赌选择,在选中的组中产生新的个体^[3];陈璐等人将遗传算法分解为两部分,通过第一次寻优找到优秀个体,将最佳个体再次放入第二个种群中,得到最后的最优解^[4];江涛等通过在适应度函数中加入路径平滑度来改善遗传算法^[5];陈友青等人提出一种改进算子的遗传算法^[6]。虽然这些研究在一定程度上改善了遗传算法的性能,但是使用固定交叉概率和变异概率的方式固化了参数,算法前期和后期染色体使用的交叉概率、变异概率一样,算法收敛速度问题依旧没有得到很好的解决。

针对遗传算法参数固定带来的问题,M. Srinvas和L. M. Patnaik提出了自适应遗传算法(Adaptive GA, AGA),其思想是在迭代过程中根据种群适应度来自适应的调整交叉概率和变异概率^[7]。公式(1)如下:

$$\begin{cases} p_c = k_1 \frac{f_{\max} - f'}{f_{\max} - \bar{f}}, f' \geq \bar{f} \\ p_c = k_3, f' \leq \bar{f} \\ p_m = k_2 \frac{f_{\max} - f}{f_{\max} - \bar{f}}, f \geq \bar{f} \\ p_m = k_4, f \leq \bar{f} \end{cases} \quad (1)$$

式中, p_c 、 p_m 分别表示交叉概率和变异概率; f_{\max} 为当代种群最大适应度值; \bar{f} 表示当代种群的平均适应度值; f 表示变异个体的适应度值; f' 表示交叉两个个体中较大的适应度的值; k_i ($i=1,2,3,4$)为常数,取值范围 $0 < k_i \leq 1$ 。

AGA解决了算法前期和后期适应度高的个体和适应度低个体的交叉概率、变异概率相同的缺点。但自适应遗传算法没有从全局出发来控制交叉概率和变异概率,所以在算法后期容易陷入局部收敛,难以找到全局最优解。针对这个问题,曲志坚等人提出在算法进化过程中使用优秀个体代替适应度差的个体^[2];刘芳等人提出基于种群多样性的IAGA来改善算法后期易陷入局部最优解的问题^[8];闫春等人使用反正弦函数自适应调节变异概率^[9]。

通过对遗传算法以及各种改进方式的对比研究,本文提出了混合改进遗传算法:根据种群密度和进化代数的协同性,改进交叉概率;通过对种群密度的统计,提出基于种群密度调整变异概率。

1.2 混合改进自适应遗传算法

1.2.1 改进交叉概率

评价算法性能指标可以用进化性来指导,更好的下一代比上一代更能适应环境变化。评价算法进化性可以使用种群的密度来衡量,本文算法中计算每一代群体密度计算方式(2)如下:

$$\rho_1 = f_{\text{avg}}/f_{\max} \quad (2)$$

分析每一代种群密度的变化可以掌握算法性能,受改进的自适应遗传算法(IAGA)^[10]的启发,在种群密度的基础上加上进化代数的因素,从种群迭代次数以及种群集中程度的考量,改进交叉概率,公式(3)如下:

$$p_c = \begin{cases} \frac{t^{0.5}}{\rho_1^{0.5} + t^{0.5}}, 0 < \rho_1 \leq 0.5 \\ \frac{\rho_1^{0.5}}{\rho_1^{0.5} + t^{0.5}}, 0.5 < \rho_1 < 1 \end{cases} \quad (3)$$

其中, ρ_1 表示当前种群的集中程度, t 是当前种群的进化代数。

1.2.2 基于种群中心区域密度改进变异概率

在算法进化的后期,种群内部已经趋于相对稳定的状态,若算法此时得到的解是局部最优,由于状态稳定,算法难以发生实质进化和走出局部最优解,如图 1 所示。自然界中的各种生物会经历“天灾”这样的自然灾害,基于这种思考,选择设置类似于自然界中的“天灾”设定,一定程度上将旧种群初始化为新的种群,从而走出此处局部最优解,并试图寻找下一个可能的局部最优解,从而使模型更有希望与能力达到全局最优解。算法遭遇“天灾”之前,变异概率随之下降,同时设置“天灾”发生后在正常的种群中心区域密度 ρ_2 下,密度越高时,变异概率也会随之上升。

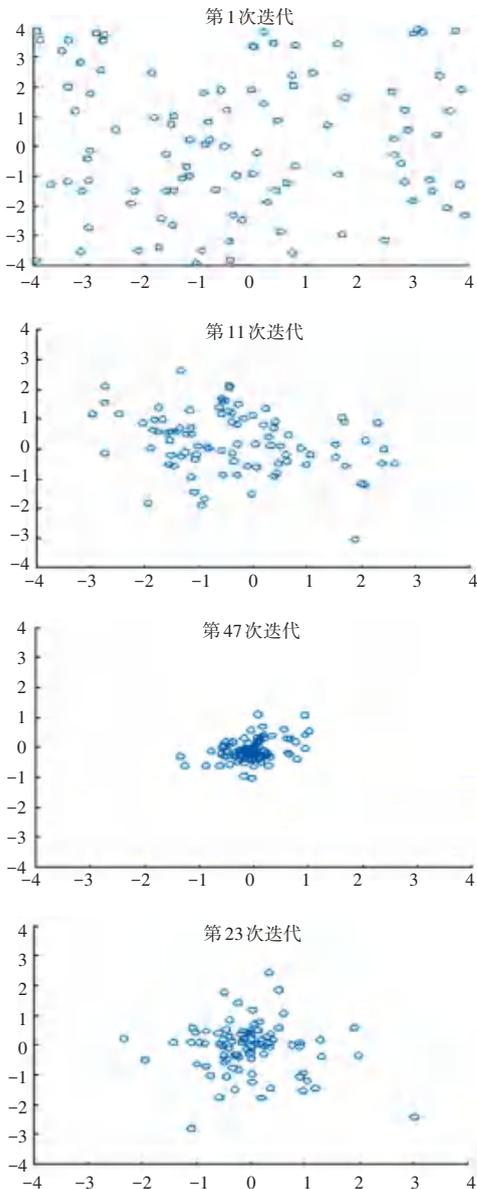


图 1 种群密度变化图

Fig. 1 Change of population density

为计算种群中心区域的密度 ρ_2 设定参数: M 表示群体所含的个体数目; f_{\max} 为种群最大适应度; f_{\min} 表示种群最小适应度; f_{avg} 表示种群平均适应度,则 t 代种群的范围 $M(H, t)$ 如式(4):

$$M(H, t) = f_{\max} - f_{\min} \quad (4)$$

某个体的适应度到群体适应度均值之间的距离 $m(h)$, 式(5):

$$m(h) = |f - f_{\text{avg}}| \quad (5)$$

群体中心区域的半径 σ 可以表示为式(6):

$$\sigma = \frac{M(H, t)}{M - 1} \quad (6)$$

某染色体适应度到当代群体适应度均值的距离小于 σ 的染色体数目 $N(\delta, f)$, 则群体中心的密度 ρ_2 为式(7):

$$\rho_2 = \frac{N(\delta, f)}{M} \quad (7)$$

通过对种群中心区域密度的分析,对变异概率改进如式(8):

$$P_m = \frac{|\log_2(t_m - t + 0.5)|}{T/4} \rho_2 \quad (8)$$

式中, T 表示最大进化次数; t 表示当前进化次数; t_m 是经历“天灾”的某代种群。

文中设定当中心区域密度达到 0.5 时,种群经历“天灾”,之后变异概率会逐渐上升,如图 2 所示。

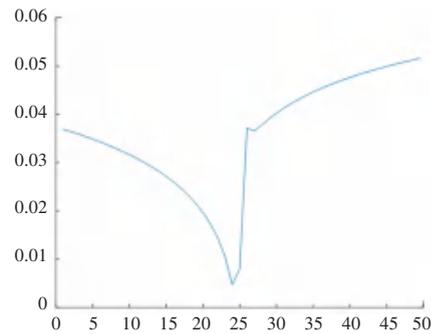


图 2 变异概率变化曲线

Fig. 2 Variation curve of mutation probability

2 仿真实验结果

本文仿真实验基于 Windows 10 (x64) 操作系统, Intel(R) Core(TM) i5-7500 CPU 3.40 GHz 主频, RAM 8G 内存。编译环境采用 MATLAB R2018a。12 个函数的形式, 寻优范围和函数最优值见表 1。使用 12 个测试函数来对本文改进后的算法进行仿真测试, 其中函数 $F_2, F_3, F_5, F_7 - F_{11}$ 拥有许多局部极小值, F_1, F_4, F_6, F_{12} 用于多维空间测试。

表1 测试函数
Tab. 1 Test functions

测试函数	范围	理论最优
$F1 = \sum_{i=1}^n x_i^2$	$[-5.12, 5.12]$	0
$F2 = 100(x_1^2 - x_2^2) + (1 - x_1)^2$	$[-2.048, 2.048]$	0
$F3 = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 - 0.001(x_1^2 + x_2^2)]^2}$	$[-100, 100]$	0
$F4 = \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \cdot \sum_{i=1}^5 i \cos[(i+1)x_2 + i]$	$[-10, 10]$	-186.731
$F5 = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1 \cdot x_2 + (-4 + 4x_2^2)x_2^2$	$x_1 [-3, 3], x_2 [-2, 2]$	-1.031 628
$F6 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0
$F7 = -\frac{1 + \cos(12 \sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	$[-5.12, 5.12]$	-1
$F8 = -0.0001(\sin(x_1)\sin(x_2) \cdot \exp(100 - \frac{\sqrt{x_1^2 - x_2^2}}{\pi}))^{0.1}$	$[-10, 10]$	-2.262 61
$F9 = -(x_2 + 47) \cdot \sin(\sqrt{ x_2 + \frac{x_2}{2} + 47 }) - x_1 \sin(\sqrt{ x_1 - (x_2 + 47) })$	$[-512, 512]$	-999.640 7
$F10 = -\exp(-0.5 \sum_{i=1}^n x_i^2)$	$[-1, 1]$	-1
$F11 = x_1^2 + 2x_2^2 - 0.3\cos(\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	$[-100, 100]$	0
$F12 = \sum_{i=1}^n x_i ^2$	$[-1, 1]$	0

实验对表1中的函数在搜索空间中进行优化测试,对比算法选择SGA以及IAGA,SGA参数设置: $p_c = 0.7, p_m = 0.01$; IAGA参数设置: $p_{c1} = 0.9, p_{c2} = 0.6, p_{m1} = 0.1, p_{m2} = 0.001$ 。使用12个函数对算法进行2维寻优性能测试分析,函数F1、F6、F10、F12进行30维测试。

2.1 改进算法性能测试

在2维情况下,将SGA,IAGA与本文算法在12个测试函数上进行试验,3种算法独立求解测试函数所得收敛曲线如图3所示,从F1~F5函数在算法迭代50次寻优的曲线,可以看出本文算法在函数F1~F5上都能近似到最优效果;同时图3还给出了F6~F12函数迭代200次的寻优效果,可以看出本文算法收敛速度明显快于SGA和IAGA。由图3可以看出,通过种群密度和迭代次数的全局考虑,以及

加入类似“天灾”设置的情况下,算法收敛速度快,易跳出局部最优值,而其余两种算法则出现收敛速度慢或陷入局部最优解等问题。虽然本文算法在前期函数F3收敛速度略劣于前两种算法,但是在中后期收敛速度明显高于前两种算法,对比来看本文算法更具优势。

2.2 收敛精度与收敛率分析

实验使用了表1中测试函数F1~F6,进行了50次独立测试,测试结果见表2。可以看出本文算法在50次独立实验中收敛精度上优于其余两种算法,收敛率(在50多次独立实验中的收敛次数)也明显高于其余两种算法,函数F3在50次实验中收敛次数也高于前两种。30维搜索空间优化实验中改进算法对比SGA和IAGA在收敛速度和跳出局部最优解能力更有优势,收敛曲线如图4所示。

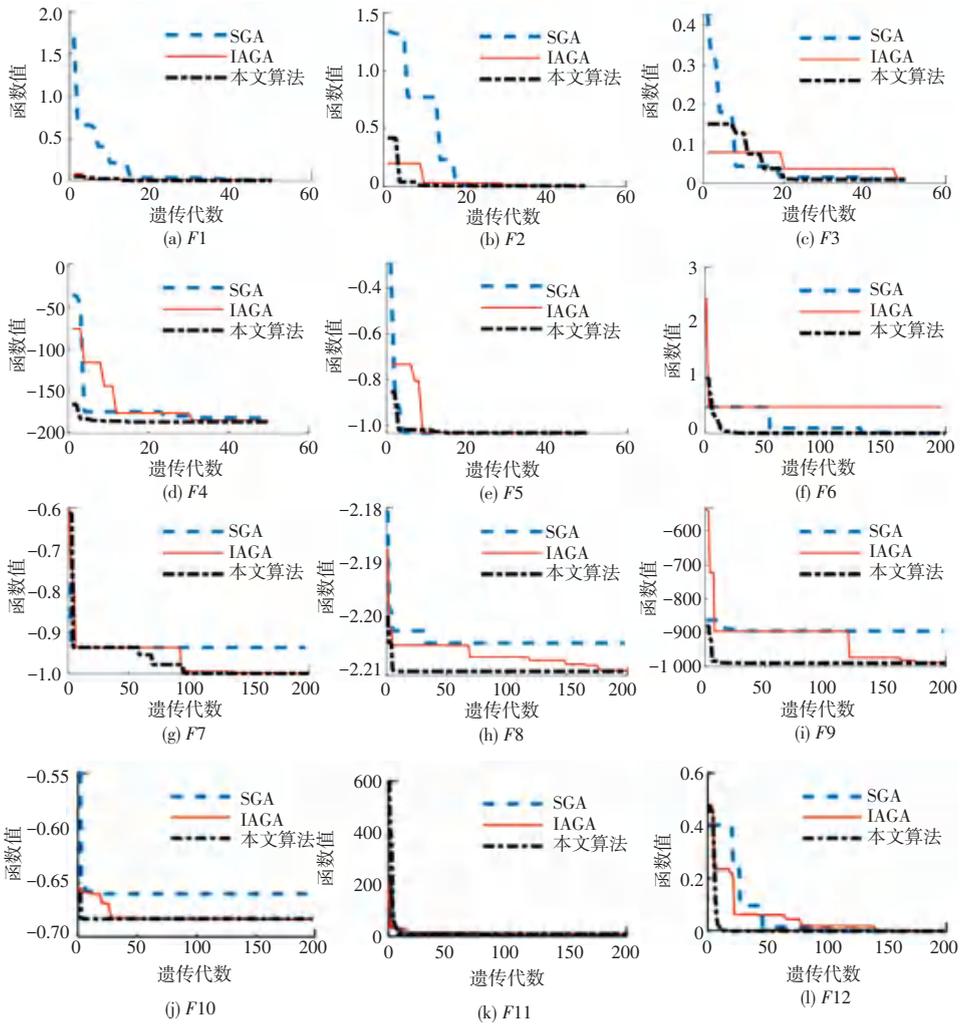


图 3 测试函数适应度变化曲线

Fig. 3 Fitness curve of test function

表 2 函数 F1~F6 在 50 次独立实验的 200 代平均值以及收敛率对比

Tab. 2 Comparison of average and convergence rate of function F1~F6 in 50 independent experiments with 200 generations

测试函数	SGA		IAGA		本文算法	
	平均值	收敛率/%	平均值	收敛率/%	平均值	收敛率/%
F1	4.29E - 12	85	1.79E - 12	85	1.07E - 12	87
F2	0.046 274 18	89	0.003 462 759	88	0.000 430 662	90
F3	1.567 834 416	-	1.548 479 668	-	0.171 106 099	-
F4	- 186.728 895 9	80	- 186.542 529 7	79	- 186.729 906	91
F5	- 1.030 628 453	81	- 1.030 568 279	85	- 1.030 628 805	87
F6	7.61E - 06	84	6.82E - 06	86	3.51E - 06	91

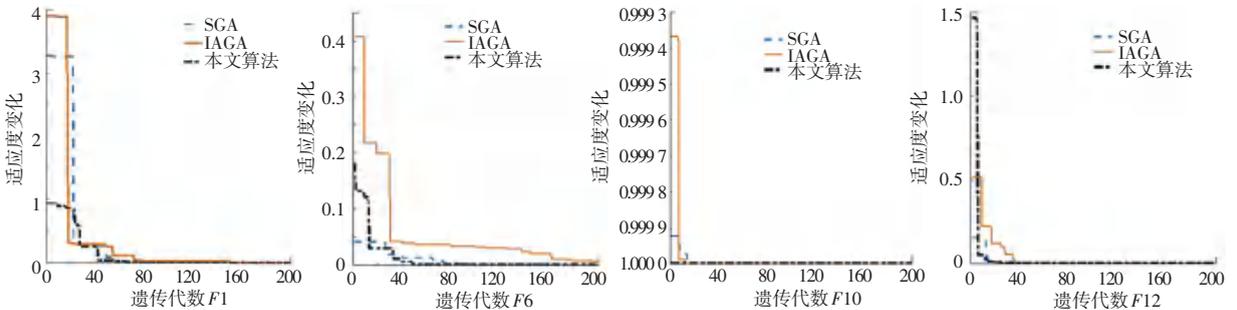


图 4 30 维搜索空间测试结果

Fig. 4 Test results of 30-dimensional search space

3 改进遗传算法图像分割

3.1 Otsu 图像分割

图像分割是计算机视觉中一个重要的研究内容,在很多领域中都是一项必不可少的图像预处理任务。基于阈值分割的大律法(Otsu)是经典的图像分割算法,但是在多阈值分割任务上性能很差,处理图像时间长,所以本文使用改进遗传算法用于阈值图像分割。图像阈值分割经典算法是 Otsu 算法,通过寻找灰度图像中的最佳阈值来分割图像,虽然在单阈值分割时 Otsu 算法表现不错,但是多阈值分割时一张图像需要处理的时间过长,算法效率太低。使用改进的遗传算法用于图像分割,流程图如图 5。

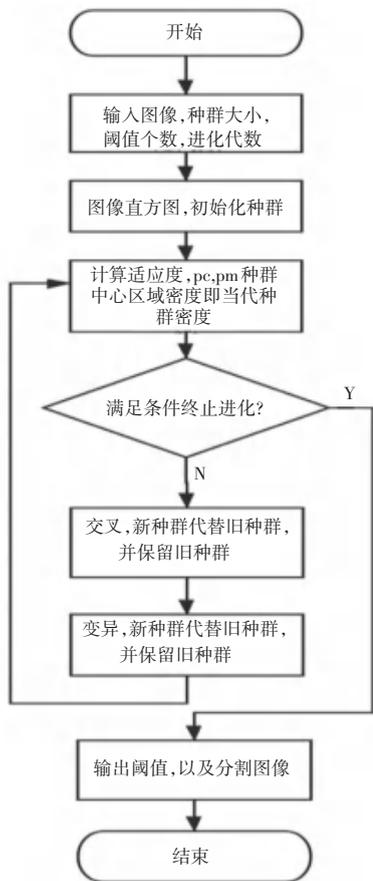


图 5 改进算法用于图像分割流程图

Fig. 5 Flow chart of improved algorithm for image segmentation

单个阈值把图像像素分为两类, Otsu 算法通过遍历每个像素点寻找能最大化类间方差的点,这个像素点就是最佳阈值。多阈值是在单阈值的基础上推广而来。若有 M 个阈值($t_1, t_2 \dots t_M$) 将图像分为 $M + 1$ 类 ($C_0, C_1 \dots C_M$), 分割标准的度量可以使用类间方差和类内方差共同决定。类内方差尽可能小,类间方差尽可能大,衡量标准则是类间方差和类内方差比值,式(9):

$$F = \frac{S_{bb}}{S_{wo}} \quad (9)$$

其中, S_{bb} 是类间方差, S_{wo} 是类内方差。

选取公式(9)为适应度函数,转换为函数寻优中的求最大值问题。灰度图像的像素级在 $[0, 255]$ 之间,阈值就是灰度值,将染色体编码为比特向量,每一个向量都有 $L \times M$ 位, L 是 $\log(\text{灰度级数})$, M 是阈值数,则每 L 位表示一个染色体,如图 6 所示。

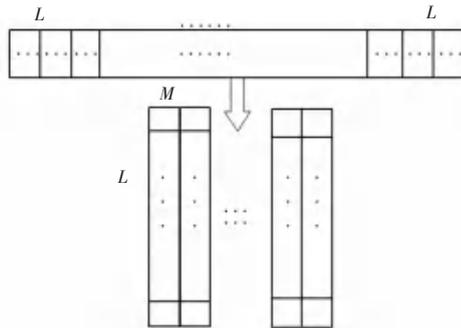


图 6 像素编码为染色体

Fig. 6 The pixels are encoded as chromosomes

3.2 图像分割实验

Otsu 算法在多阈值分割时出现分割速度慢、抗噪性能差的缺点,采用改进的遗传算法优化 Otsu 多阈值分割,能在一定程度上克服这些不足之处,实验结果见表 3、如图 7 所示。使用原 Otsu 算法、SGA、IAGA 以及本文改进的算法对图像进行分割,传统 Otsu 算法分割效率低;遗传算法的加入解决效率低的问题,但由图 7 的实验结果来看,SGA、IAGA 算法还有明显的噪声,在细小部分分割效果差。使用遗传算法分割图像在单阈值和多阈值时所用时间对比见表 3。

表 3 四种方法分割时间对比

Tab.3 Comparison among the four methods on time consumption

图像	OTSU		SGA		IAGA		本文算法	
	单阈值/s	多阈值/s	单阈值/s	多阈值/s	单阈值/s	多阈值/s	单阈值/s	多阈值/s
Man	1.971 3	7.885 2	0.180 0	0.654 0	0.196 0	0.529 0	0.177 0	0.389 0
Horse	2.106 61	8.426 5	0.311 0	0.431 0	0.188 0	0.391 0	0.178 0	0.372 0
Airplane	1.895 13	1.209 31	0.196 0	0.562 0	0.186 0	0.606 0	0.195 0	0.391 0

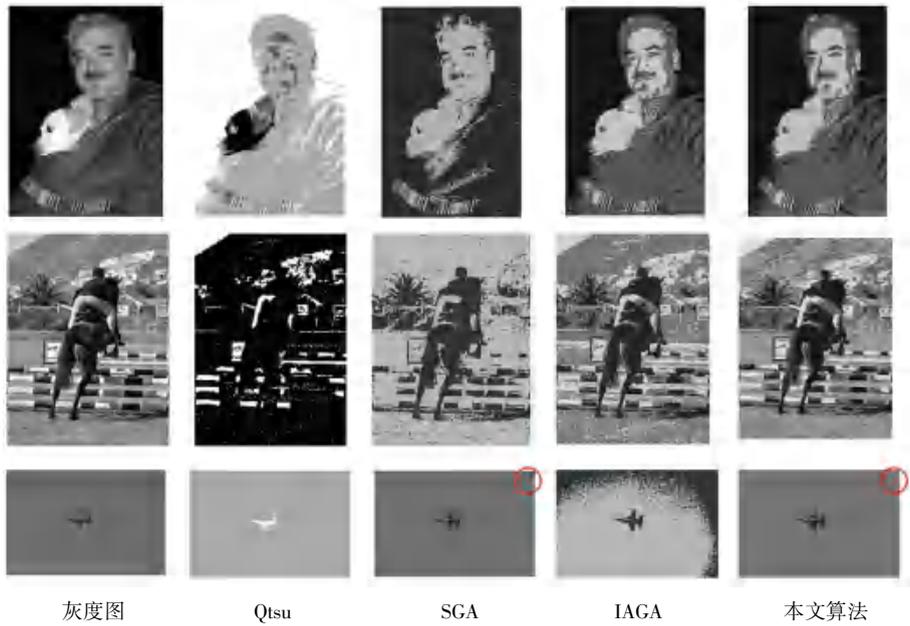


图 7 图像分割结果

Fig. 7 Image segmentation results

4 结束语

针对标准遗传算法在进化过程难以跳出局部最优解、收敛率较低和收敛速度慢等问题,本文通过以下两种方式改进遗传算法:首先,考虑种群密度和进化代数的协同性,改进交叉概率,以增强交叉概率自适应性;其次,在进化过程中又设定类似于“天灾”设置,结合进化代数改进变异概率,增强种群跳出局部最优解的能力。相较于简单遗传算法和自适应遗传算法,本文算法在收敛速度和收敛率上有明显优势,在跳出局部最优解的能力上也优于 SGA 和 IAGA。通过改进算法分割图像也达到不错的效果,相较于传统的 Otsu 算法,结合遗传算法优化之后效果更好,在与 IAGA、SGA 优化对比中,本文改进算法更具有鲁棒性,抗噪能力更强。

参考文献

[1] John H. Holland. *Adaptation in Natural Artificial Systems* [M].

MIT Press, 1975.

- [2] 曲志坚,张先伟. 基于自适应机制的遗传算法研究[J]. 计算机应用研究, 2015, 32(11), 3222-3225, 3229.
- [3] 赵大兴,等. 基于高适应度值遗传算法的 AGV 最优路径规划[J]. 计算机工程与设计, 2017, 38(6), 1635-1641.
- [4] 陈璐,王秀. 改进遗传算法求解走班制下的排课问题[J]. 计算机工程与应用, 2019, 55(6): 218-224.
- [5] 江涛,张志安,程志,等. 改进遗传算法与领航跟随法的机器人编队方法[J]. 计算机工程与应用, 2020, 56(3): 240-245.
- [6] 陈友青,徐蔡星,钟天亮,等. 一种改进算子的遗传算法[J]. 计算机工程与应用, 2008, 44(2): 44-49.
- [7] SRINIVAS M, PATNAIK L M. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms[J]. IEEE TRANSACTIONS
- [8] 刘芳,马玉磊,周慧娟. 基于种群多样性的自适应遗传算法优化仿真[J]. 2017, 34(4): 250-255.
- [9] 闫春,厉美璇. 基于改进的遗传算法在函数优化中的应用[J]. 计算机应用研究, 2019, 36(10), 2982-2985.
- [10] WANG L, TANG D. An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem[J]. *Expert Systems with Applications*, 2011.

(上接第 108 页)

- [8] 曹建斌. 基于深度学习的高速铁路混凝土轨枕裂纹检测[D]. 上海:上海大学, 2020: 16-21.
- [9] 王宇桐,许贵阳,周素霞. 基于垂向定位与灰度投影的地铁扣件定位方法[J]. 铁道建筑, 2019, 59(6): 112-116.
- [10] 刘博,王胜正,赵建森,等. 基于 Darknet 网络和 YOLOv3 算法的船舶跟踪识别[J]. 计算机应用, 2019, 39(6): 1663-1668.
- [11] 李汉冰,徐春阳,胡超超. 基于 YOLOV3 改进的实时车辆检测方法[J]. 激光与光电子学进展, 2020, 57(10): 332-338.
- [12] 任欢,王旭光. 注意力机制综述[EB/OL]. 计算机应用: 1-7

[2021-04-09]. <http://kns.cnki.net/kcms/detail/51.1307.TP.20210122.1747.022.html>.

- [13] 王艺皓,丁洪伟,李波,等. 复杂场景下基于改进 YOLOv3 的口罩佩戴检测算法[J]. 计算机工程, 2020, 46(11): 12-22.
- [14] ZHOU Siqi, BI Yufeng, WEI Xu, et al. Automated detection and classification of spilled loads on freeways based on improved YOLO network[J]. *Machine Vision and Applications*, 2021, 32(2): 32-44.