

一种改进的演化算法及其在求解复杂优化问题中的应用*

李康顺^{1,2†}, 余锡伦¹, 张文生³

(1. 江西理工大学 信息工程学院, 江西 赣州 341000; 2. 华南农业大学 信息学院, 广州 510642; 3. 中国科学院 自动化研究所, 北京 100190)

摘要: 针对传统演化算法在求解函数优化,特别是多峰函数优化问题中出现的早熟现象以及演化后期收敛速度慢等问题,提出了一种新的反序小生境演化算法。该算法采用小生境反序交叉算子,以进一步增强局部寻优的能力;引入一种并行演化算法机制,加强群体寻优能力;同时,根据定义域划分初始种群,增加初始种群的覆盖面积。通过仿真实验表明,与传统的小生境演化算法相比较,利用该算法求解复杂多峰函数优化问题能够明显提高问题的求解精度和收敛速度,而且能够得到所有的全局最优解,更好地避免了求解问题时的早熟现象,达到了较好的效果。

关键词: 演化算法; 多峰函数优化; 小生境演化算法; 反序交叉算子

中图分类号: TP301 **文献标志码:** A **文章编号:** 1001-3695(2012)04-1223-04

doi:10.3969/j.issn.1001-3695.2012.04.006

Improved evolutionary algorithm and its application to solving complex optimization problems

LI Kang-shun^{1,2†}, YU Xi-lun¹, ZHANG Wen-sheng³

(1. School of Information Engineering, Jiangxi University of Science & Technology, Ganzhou Jiangxi 341000, China; 2. School of Information, South China Agricultural University, Guangzhou 510642, China; 3. Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Aiming at the problems of premature and slow convergent speed by using the traditional evolutionary algorithm in solving the optimization problems, especially in solving the optimization problems of multimodal functions, this paper proposed a niche inver-over evolutionary algorithm. This new algorithm adopted a niche inver-over operator to further enhance the local optimization ability. The algorithm introduced a mechanism of parallel evolutionary algorithm to strengthen the ability of swarm optimization. Besides, it divided the initial population heuristically to some sub-populations according to the domain for increasing the coverage area of initial population. Simulated experiments show that it enhances both solution precision and convergent speed more obviously by using this new algorithm to solve optimization problems of multimodal functions compared with the traditional niche evolutionary algorithm, and it avoids premature phenomenon better.

Key words: evolutionary algorithm; multimodal function optimization; niche evolutionary algorithm; inver-over operator

0 引言

演化算法^[1]是一类借鉴生物界的演化规律(适者生存、优胜劣汰遗传机制)演化而来的随机化搜索方法,由美国的 Holland 教授最先提出。由于演化算法具有自组织、自适应、自学习等特性以及其内在的并行性,相对于传统的搜索和优化方法,它有着先天的优越性,也受到了越来越多的关注和重视。演化算法提供了一种求解复杂系统优化问题的通用框架,也就是说,它在求解问题时可以不依赖于该问题的具体领域,对问题的种类具有很强的鲁棒性,所以广泛应用于函数优化、组合优化、人工智能、机器学习、图像处理和模式识别等领域。

尽管演化算法在理论研究和实际应用中已经取得了巨大的成功,但是其局部搜索能力差及早熟等问题还是一直存在的。导致早熟的一个重要原因在于种群的多样性,种群的多样性越大,种群就越有可能产生出更优的模式;相反,如果种群的多样性很低,种群个体趋于相同,那么种群产生新模式的可能性就很低,进而加快收敛导致早熟。

在简单的演化算法中,个体的交配完全是随机的,虽然这种随机的杂交形式在寻优的初始阶段保持了种群的多样性,但在演化的后期,种群中的个体趋于相同,因而很难跳出局部最优解,虽然变异算子在一定程度上能增加种群的多样性,但是变异的概率一般都设置比较小的值,以防破坏一些较优的个

收稿日期: 2011-08-20; 修回日期: 2011-09-27 基金项目: 国家自然科学基金资助项目(70971043); 江西省自然科学基金资助项目(2008GZS0028)

作者简介: 李康顺(1962-),男(通信作者),江西兴国人,教授,博导,博士(后),主要研究方向为演化计算、图像视觉、演化硬件(likangshun@sina.com); 余锡伦(1987-),女,硕士研究生,主要研究方向为演化计算; 张文生(1966-),男,河南郑州人,研究员,博导,主要研究方向为机器学习理论与算法。

体。所以当用演化算法解决多峰问题时,往往得到的只是局部最优解或者是个别最优解,不能得到全部最优解。为了能够找出全部的最优解,文献[2]提出了小生境的概念,并取得了较好的效果,之后也有不少关于小生境演化算法的研究^[3-9]。虽然这些算法取得了一定的成功,但是在精确度和收敛速度上也还是有局限的,实现过程也较为复杂。为了进一步提高算法的性能,本文提出一种求解复杂多峰函数优化问题的反序小生境演化算法,实验结果表明,该方法有效地增加了种群中个体的多样性,不仅能很快地得到所有全局最优解,而且有较快的收敛速度。

1 传统的小生境演化算法

小生境是来自于生物学上的一个概念,是指在自然界中,物以类聚、人以群分,往往特征、形状相似的物种会聚集在一起,共同繁衍后代。通常所说的小生境是指物种赖以生存的资源环境。在有限的生存环境中,各种不同的生物必须要相互竞争各种有限的资源,这样他们才能够延续生存。基于这个思想,De Jong 于 1975 年提出了一种基于排挤机制的小生境演化算法^[2],其基本思想就是在算法中设置一个排挤因子 CF (一般取 2 或 3),从种群中随机挑选出 $1/CF$ 个个体组成排挤成员。将新产生的个体与排挤成员进行相似性比较,排挤掉那些与排挤成员相似的个体。其中个体间的相似性通常使用个体间的海明距离或欧式距离来度量。在排挤过程中,个体逐渐被分类,形成一个个小生境,从而更好地保持了群体的多样性,避免陷入局部最优,令算法特别适合求解复杂的多峰函数优化问题。

基于上述思想的小生境演化算法描述如下(以下算法默认求函数最大值):

a) 随机生成一个包含 M 个个体的初始种群,设定好各参数值,并计算各个体的适应值。

b) 根据适应值进行降序排序,并保存该种群前 N 个个体,其中 $N < M$ 。

c) 对种群进行选择、交叉、变异等遗传操作。

d) 小生境排挤运算。将步骤 c) 得到的 M 个个体和之前保存的 N 个个体合并,得到一个包含 $M + N$ 个个体的种群,对这 $M + N$ 个个体进行两两比较,求出个体间的海明距离或是欧式距离 d_{ij} 。当两者间的距离 $d_{ij} < L$ (L 为小生境距离) 时,比较两者的适应值,并对适应值较低的个体适应值处以罚函数 penalty,一般这个罚函数设置很小。为了极大降低其适应度,使得在距离 L 范围内只有一个较优的个体,增加种群间个体的差异性。

e) 对这 $M + N$ 个个体新的适应值进行降序排序,并保存前 N 个个体,替代之前保存的 N 个个体。

f) 终止条件判断。若不满足终止条件,则将步骤 e) 得到的前 M 个个体作为新一代种群,返回步骤 c) 重新演化;否则,输出最优个体。

2 改进的小生境演化算法

上述提到的是一种常用的小生境演化算法,本文在该算法的基础上作了如下改进。

a) 加入并行的思想。由于演化算法具有先天的并行性,

早前也有很多并行演化算法的研究^[10-12],并取得了很好的效果,所以将并行的思想加入到小生境演化算法将更能体现出并行思想的优越性。将种群分为若干个子种群,各子种群并行工作,各自独立完成自身的进化,并且在算法中引入迁移算子,采用精英迁移策略将子种群中最好的个体分发给其他的子种群,代替这些子种群中最差的个体,使得好的个体在各子群中传播,从而可以提高求解精度和求解速度。在改进算法中子种群中最好个体不是直接取代其他各子种群中最差的个体,而是先计算它与各子种群中最差个体间的欧式距离,只有当这个距离小于小生境距离 L 时才替代子种群中最差的个体,避免在进化后期某些较好的个体直接被取代而得不到所有的全局最优解。

b) 种群初始化不是纯随机进行的,而是根据定义域初始化分组后随机进行。种群的初始化也是一个不可忽视的环节,如果初始种群选择得好,产生好的后代的概率也比较高。往往都希望初始种群可以覆盖整个定义域区间,而随机初始化产生的种群并不一定能完全覆盖整个定义域区间。所以首先根据定义域来划分子空间,然后再进行随机初始化子种群,这样就加大了初始种群的覆盖面积,更能提高算法的性能。

c) 采用反序交叉算子。反序交叉算子^[13]是由 Guo Tao 等人提出的一种新演化算子,同时具有反序和交叉的特征,实验证明利用反序交叉算子在产生后代时能跳离局部最优值,并且使算法具有自适应性等优点。所谓反序就是将两点间的基因序列作反向处理产生新个体,提高算法的局部搜索能力。起初这种新算子的提出主要是针对求解 TSP 问题,并取得了很好的效果^[14,15]。在函数优化的后期,个体间差异很小,往往难以跳出局部最优值,通过引入该新算子能进一步解决这一问题。

在反序交叉算子中,反序交叉概率 p 一般取较小的值,如果随机数 $\text{rand}() < p$,则进行反序操作,反之则进行交叉操作。反序交叉所产生的新个体必须与原父体相比较,只有当新个体优于父体时,新个体才可替代父体,否则保留原父体。

d) 采用自适应变异算子。变异算子也是演化算法的一个重要的组成部分,通过变异算子来产生一个不同于父代性质的个体,可以有效地增加算法的局部搜索能力,同时又能增加个体的多样性,有效避免早熟现象。变异概率 p_m 的选取尤为重要, p_m 取值太大则容易丢失优秀个体,反之则很难跳出局部最优。本文采用自适应变异算子, p_m 会随个体适应值的变化而变化,适应值较大的个体具有较小的变异概率,而适应值较小的个体具有较大的变异概率。

变异概率的计算公式如下:

$$p_m = \begin{cases} k_1 \frac{(f_{\max} - f)}{f_{\max} - f_{\text{avg}}} & f \geq f_{\text{avg}} \\ k_2 & f < f_{\text{avg}} \end{cases}$$

其中: f 为要变异个体的适应值; f_{\max} 为种群中最大的适应值; f_{avg} 为种群中的平均适应值; k_1 、 k_2 为系数,代表 p_m 变化的快慢程度,本文中 $k_1 = 0.2$, $k_2 = 0.2$ 。

算法描述如下(默认求函数最大值):

```
{
    将定义域区间分为 S 组,构成 S 个子种群;
    for( 每个子种群 ) {
        初始化各子种群,子种群规模为 M;
    } //end for
    Gen = 1; //初始演化代数
    while( Gen < = MaxGen ) //MaxGen 为最大演化代数
```

```

if(Gen % 预定代数间隔 t = 0) /* 每隔预定的代数间隔 t 就
进行精英迁移 */
for(每个子种群)
保存各子种群中适应值最大的个体及其适应值; //end for
将各子种群中适应值最大的个体集合到一起,形成一个缓冲池;
对这个缓冲池中的个体进行反序交叉操作,并记录此时适应值
最大的个体;
计算这个适应值最大的个体与各个子种群中适应值最小个体之
间的欧式距离  $d_i, i=1,2,\dots,S$ 
for(每个距离  $d_i$ )
if ( $d_i < L$ )
用该适应值最大的个体代替子种群 i 中适应值最小的个体;
//end if
//end for
//end if
for(每个子种群)
适应度降序排序;
记录前 N 个个体作为排挤成员;
反序交叉操作;
适应变异操作;
将子种群中新产生的 M 个个体和先前记录的 N 个个体合并;
排挤运算;
根据新适应值对此含有 M + N 个个体的种群降序排序;
将前 M 个个体作为新子种群,并记录前 N 个个体; //end for
Gen = Gen + 1
//end while
//end

```

3 实验仿真与结果分析

为了验证算法的性能,本文选取了三个具有代表性的 Schaffer 函数、Humpback 函数和 Shubert 函数进行实验。将本文所提出的小生境改进算法与传统演化算法 SEA 和传统小生境演化算法 NEA 进行性能对比研究。

1) Schaffer 函数

$$f_1(x,y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1 + 0.001(x^2 + y^2)]^2} \quad -100 \leq x, y \leq 100$$

函数 f_1 为 Schaffer 函数,分别用本文所提出的小生境改进算法、传统演化算法和传统小生境演化算法求出该函数的最大值及最大值所对应的点。该函数有无数个局部极大值点,围成一个圈,但它只有一个全局最大值点 $f_1(0,0) = 1$,这个最大值点被无数个局部极大值点所包围。在使用传统的算法时得到的往往是局部极大值,且很难跳出这个局部最优点。Schaffer 函数实验参数的选取如表 1 所示。

表 1 Schaffer 函数实验参数

| 参数 | 参数值 |
|---------------|--|
| 子种群数 S | 4 |
| 子种群 1 所在的定义域 | $-100 \leq x \leq 0, 0 \leq y \leq 100$ |
| 子种群 2 所在的定义域 | $0 \leq x \leq 100, 0 \leq y \leq 100$ |
| 子种群 3 所在的定义域 | $-100 \leq x \leq 0, -100 \leq y \leq 0$ |
| 子种群 4 所在的定义域 | $0 \leq x \leq 100, -100 \leq y \leq 0$ |
| 子种群规模 M | 100 |
| 记录优秀个体数 N | 50 |
| 小生境距离 L | 0.5 |
| 罚函数 penalty | 10^{-30} |
| 最大运行代数 MaxGen | 200 |
| 预定代数间隔 t | 10 |
| 反序交叉概率 p | 0.2 |
| 运行次数 | 30 |

使用三种算法分别对 Schaffer 函数进行 30 次仿真实验,若实验值与目标函数最优值之间的误差小于 0.1%,定义当次实验是成功的。实验结果如表 2 所示。

表 2 Schaffer 函数实验结果

| 函数 | 算法 | 最优值 | 平均值 | 平均演化代数 | 成功次数 |
|-------|------|------------------------------------|----------|--------|------|
| f_1 | SEA | $f_1(-0.1742, -3.1339) = 0.990284$ | 0.972729 | 90.3 | 0 |
| | NEA | $f_1(0.0036, 0) = 0.999988$ | 0.992784 | 173.2 | 10 |
| | 本文算法 | $f_1(0, 0) = 1$ | 0.999994 | 69.6 | 30 |

图 1 描述了 Schaffer 函数在不同进化代数后的种群个体分布情况。

2) Humpback 函数

$$f_2(x,y) = (4 - 2.1x^2 + x^4/3)x^2 + xy + (-4 + 4y^2)y^2 \quad -3 \leq x \leq 3, -2 \leq y \leq 2$$

函数 f_2 为 Humpback 函数,分别用三种算法求出该函数的最小值及最小值点。 f_2 是六峰值驼背函数,该函数共有六个局部极小值点;两个全局最大值点,分别为 $f_2(-0.0898, 0.7126) = -1.031628$ 和 $f_2(0.0898, -0.7126) = -1.031628$ 。Humpback 函数实验参数的选取如表 3 所示。

表 3 Humpback 函数实验参数

| 参数 | 参数值 |
|---------------|--------------------------------------|
| 子种群数 S | 2 |
| 子种群 1 所在的定义域 | $-3 \leq x \leq 3, 0 \leq y \leq 2$ |
| 子种群 2 所在的定义域 | $-3 \leq x \leq 3, -2 \leq y \leq 0$ |
| 子种群规模 M | 100 |
| 记录优秀个体数 N | 50 |
| 小生境距离 L | 0.5 |
| 罚函数 penalty | 10^{-30} |
| 最大运行代数 MaxGen | 200 |
| 预定代数间隔 t | 10 |
| 反序交叉概率 p | 0.2 |
| 运行次数 | 30 |

使用三种算法分别对 Humpback 函数进行 30 次仿真实验,实验结果如表 4 所示。

表 4 Humpback 函数实验结果

| 函数 | 算法 | 最优值 | 平均值 | 平均演化代数 | 成功次数 |
|-------|------|------------------------------------|-----------|--------|------|
| f_2 | SEA | $f_2(0.0897, -0.7109) = -1.031628$ | -1.027540 | 89.6 | 21 |
| | NEA | $f_2(-0.0898, 0.7126) = -1.031628$ | -1.029267 | 114.5 | 23 |
| | 本文算法 | $f_2(-0.0898, 0.7126) = -1.031628$ | -1.031589 | 47.2 | 30 |

图 2 描述了 Humpback 函数在不同进化代数后的种群个体分布情况。

3) Shubert 函数

$$f_3(x,y) = \left\{ \sum_{i=1}^5 i \cos[(i+1)x + i] \right\} \times \left\{ \sum_{i=1}^5 i \cos[(i+1)y + i] \right\} \quad -10 \leq x, y \leq 10$$

函数 f_3 为 Shubert 函数,分别用三种算法求出该函数的最小值及最小值点。定义域范围内,共有 18 个全局极小值点,局部极小值点也远远超过全局极小值点的数量,其全局最小值为 -186.730909 。Shubert 函数实验参数的选取如表 5 所示。

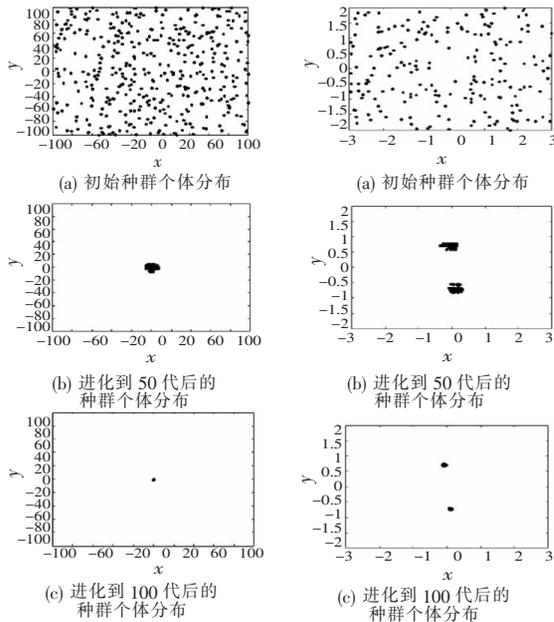


图 1 Schaffer 函数实验在不同时期的种群个体分布

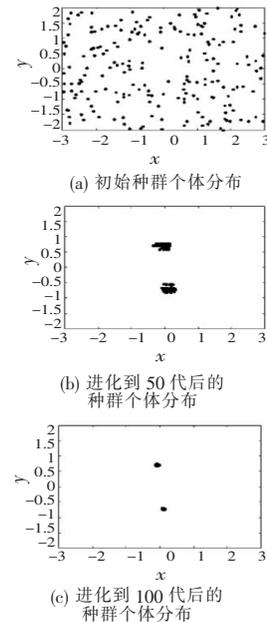


图 2 Humpback 函数实验在不同时期的种群个体分布

表 5 Shubert 函数实验参数

| 参数 | 参数值 |
|---------------|--|
| 子种群数 S | 4 |
| 子种群 1 所在的定义域 | $-10 \leq x \leq 0, 0 \leq y \leq 10$ |
| 子种群 2 所在的定义域 | $0 \leq x \leq 10, 0 \leq y \leq 10$ |
| 子种群 3 所在的定义域 | $-10 \leq x \leq 0, -10 \leq y \leq 0$ |
| 子种群 4 所在的定义域 | $0 \leq x \leq 10, -10 \leq y \leq 0$ |
| 子种群规模 M | 100 |
| 记录优秀个体数 N | 50 |
| 小生境距离 L | 0.5 |
| 罚函数 penalty | 10^{-30} |
| 最大运行代数 MaxGen | 200 |
| 预定代数间隔 t | 10 |
| 反序交叉概率 p | 0.2 |
| 运行次数 | 30 |

使用三种算法分别对 Shubert 函数进行 30 次仿真实验,实验结果如表 6 所示。

表 6 Shubert 函数实验结果

| 函数 | 算法 | 最优值 | 平均值 | 平均演化代数 | 成功次数 |
|-------|------|--------------|--------------|--------|------|
| f_3 | SEA | -186.730 857 | -181.138 785 | 55 | 6 |
| | NEA | -186.730 909 | -186.700 827 | 106.8 | 21 |
| | 本文算法 | -186.730 909 | -186.727 382 | 17 | 30 |

运用本文中的算法可以得到所有的全局最优点,分别为 $(-7.08350 -7.70831)$; $(-0.80032 -7.70831)$; $(5.48286 -7.70831)$; $(-7.70831 -7.08350)$; $(-1.42513 -7.08350)$; $(4.85805 -7.08350)$; $(-7.08350 -1.42513)$; $(-0.80032 -1.42513)$; $(5.48286 -1.42513)$; $(-7.70831 -0.80032)$; $(-1.42513 -0.80032)$; $(4.85805 -0.80032)$; $(-7.08350 4.85805)$; $(-0.80032 4.85805)$; $(5.48286 4.85805)$; $(-7.70831 5.48286)$; $(-1.42513 5.48286)$; $(4.85805 5.48286)$ 。而运用其他两种方法得到所有全局最优点的概率比较低,甚至可能得不到全局最优点。图 3 描述了 Shubert 函数在不同进化代数后的种群个体分布情况。

从表 2、4 和 6 可以看出,使用新算法的实验成功率为 100%,收敛速度比其他两种算法快,能在较低的代数内收敛到最优解,得到的解不仅精确度高,而且能找到所有的全局最优解。图 1~3 是新算法在求解函数最优解的过程中种群个体分

布的情况,从这些图中可以看出,对于三个实验函数,都能在 100 代以内演化出所有的全局最优解。这些实验说明了文中算法的性能优于其他两种算法,该算法对于求解复杂的多峰函数优化问题是很有效的。

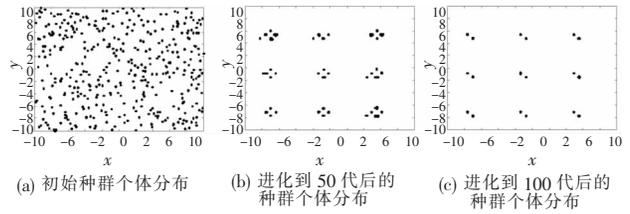


图 3 Shubert 函数实验在不同时期的种群个体分布

4 结束语

本文在小生境演化算法的基础上引入并行演化算法的思想,将种群分成若干个子种群,各子种群并行工作,独立进化,其中子种群根据定义域划分,并且采用一种新的反序—交叉算子,从而保证了种群中个体的多样性;同时加强算法的全局搜索能力和局部搜索能力,弥补了传统算法在求解函数优化问题,特别是多峰函数优化问题中存在的早熟现象、精度差和收敛速度慢等问题。通过实验结果表明,该算法与传统的算法相比具有明显的优越性。

参考文献:

- [1] HOLLAND J H. Adaptation in natural and artificial systems[M]. Ann Arbor, USA: University of Michigan Press, 1975.
- [2] 王小平,曹立明. 演化算法理论、应用与软件实现[M]. 西安:西安交通大学出版社, 2006.
- [3] 谢凯. 排挤小生境遗传算法的研究与应用[D]. 淮南:安徽理工大学, 2005.
- [4] WEI Ling-yun, ZHAO Mei. A niche hybrid genetic algorithm for global optimization of continuous multimodal functions[J]. Applied Mathematics and Computation, 2005, 160(3): 649-661.
- [5] YU Xin-jie, YU Zan-ji. Improved sequential niche genetic algorithm for multimodal optimization [J]. Tsinghua University Journal of Nature Science, 2001, 41(3): 17-20.
- [6] EI IMRANI A, BOUROUMI A, EI ABIDINE Z H, et al. A fuzzy clustering-based niching approach to multimodal function optimization[J]. Journal of Cognitive System Research, 2000, 1(2): 119-133.
- [7] JIANG Da-zhi, WU Zhi-jian, KANG Li-shan. A simple and effective evolutionary algorithm for multimodal function optimization[C]//Progress in Intelligence Computation and Applications. 2005.
- [8] 华洁,崔社武. 基于个体优化的自适应小生境演化算法[J]. 计算机工程, 2010, 36(1): 194-196.
- [9] 李智勇,童调生. 基于多种群进化小生境遗传算法的神经网络进化设计方法研究[J]. 控制与决策, 2003, 18(5): 607-610.
- [10] MÜHLENBEIN H, SCHOMISCH M, BORN J. The parallel genetic algorithm as function optimizer[J]. Parallel Computing, 1991, 17(6-7): 619-632.
- [11] 高家全,何桂霞. 并行遗传算法研究综述[J]. 浙江工业大学学报, 2007, 35(1): 56-59.
- [12] 邓长春,朱儒明,李咏霞,等. 一种求解 TSP 问题的多种群并行遗传算法[J]. 计算机仿真, 2008, 25(9): 187-190.
- [13] GUO Tao, MICHALEWICZ Z. Inver-over operator for the TSP [C]//Proc of the 5th International Conference on Parallel Problem Solving from Nature. London: Springer-Verlag, 1998: 803-812.
- [14] 危明,李元香,姜大志,等. 基于精英策略的反序—杂交算法[J]. 武汉理工大学学报:信息与管理工程版, 2008, 30(4): 514-518.
- [15] 向佐勇,刘正才. 几种局部优化算子在求解 TSP 中的性能比较[J]. 计算机工程与设计, 2009, 30(8): 1950-1953.