

# 基于改进拥挤距离的多目标进化算法

汪文彬<sup>1</sup>, 钟声<sup>2</sup>

(1. 琼州学院计算机科学与技术系, 五指山 572200; 2. 海南大学信息科学与技术学院, 海口 570228)

**摘要:** 针对多目标进化算法的拥挤距离截断算子的分布度保持不足以及在二进制编码情况下较难收敛的缺点, 提出一种改进的多目标进化算法, 使用改进的拥挤距离截断算子和自适应变异算子, 与经典的多目标进化算法进行对比, 实验表明, 该算法得到的 Pareto 解集具有良好的收敛性和分布性。

**关键词:** 多目标进化算法; 拥挤距离; 变异算子

## Multi-Objective Evolution Algorithm Based on Improved Crowding-distance

WANG Wen-bin<sup>1</sup>, ZHONG Sheng<sup>2</sup>

(1. Dept. of Computer Science and Technology, Qiongzhou University, Wuzhishan 572200;

2. College of Information Science and Technology, Hainan University, Haikou 570228)

**【Abstract】** In terms of the inadequacy of Multi-Objective Evolution Algorithm(MOEA) with the crowding-distance truncation operator to preserve the distribution and the deficiency of the distribution that is hard to get near to the true Pareto front under the binary condition, an improved MOEA is proposed. The improved algorithm includes the improved crowding-distance truncation operator and the self-adaptive mutation operator. Compared to other classical MOEA, experiment analysis proves that the improved algorithm achieving the final Pareto solutions qualified the better convergence and the good distribution to the true Pareto front.

**【Key words】** Multi-Objective Evolution Algorithm(MOEA); crowding-distance; mutation operator

求解多目标问题(Multi-Objective Problem, MOP)需要对多个相互冲突的目标函数同时优化。基于种群操作的进化计算可以隐并行地搜索解空间中的多个解, 因此, 进化计算非常适合求解 MOP。在多目标进化算法中<sup>[1]</sup>, 后决策技术运用得比较广泛, 它通过运行多目标进化算法(Multi-Objective Evolutionary Algorithm, MOEA)产生一组解集供决策者选择。其中, 具有代表性的 MOEA 有 NSGA2<sup>[2]</sup>, SPEA2<sup>[3]</sup>等。为使算法得到的解集更好地均匀分布在 Pareto 前沿面上, 本文对拥挤距离<sup>[2]</sup>加入更新算子, 设计一种动态的自适应变异算子。

### 1 Pareto最优的相关描述

通常一个 MOP 可表示为

$$\text{Max } y = f(x) = (f_1(x), f_2(x), \dots, f_n(x))$$

$$\text{s.t. } g_i(x) \leq 0$$

其中, 决策向量  $x \in R^m$ ; 目标向量  $y \in R^n$ ; 目标函数  $f_i(x), i=1, 2, \dots, n$ ;  $g_i(x) \leq 0, i=1, 2, \dots, h$  是约束条件。

对 MOP 的非劣最优解定义如下:

**定义(非劣最优解)** 若  $x^*$  是搜索空间中一点, 设  $x^*$  为非劣最优解, 当且仅当不存在  $i$  使  $f_i(x) > f_i(x^*)$  成立。

相应非劣最优解的目标向量称为非支配目标向量(non-dominator)。由所有非支配的目标向量构成 MOP 的非劣最优目标域(Pareto front), 也称为 Pareto 最优前端。

### 2 改进的拥挤距离

当所要求解多目标优化问题的 Pareto 前沿面具有连续的特征时, 其 Pareto 解集的数目趋于无穷大, 这就需要修剪算子对外部集进行截断, 以保持外部集的均匀性。现今使用较

多、影响较大的外部集保持策略为拥挤距离策略<sup>[2]</sup>, 其拥挤距离的计算方式为: 设有 2 个子目标  $f_1$  和  $f_2$ , 设  $P[i]_{distance}$  为个体  $i$  的聚集距离,  $P[i].m$  为个体  $i$  在子目标  $m$  上的函数值, 则个体  $i$  的拥挤距离为

$$P[i]_{distance} = (P[i+1].f_1 - P[i-1].f_1) + (P[i+1].f_2 - P[i-1].f_2) \quad (1)$$

**算法 1** 基于拥挤距离的截断算法

**Step1** 初始化每个个体的拥挤距离,  $P[i]_{distance} = 0$ ;

**Step2** 对每个目标进行排序, 根据式(1)计算每个个体的拥挤距离;

**Step3** 将边界点赋予最大值以确保进入下一代;

**Step4** 对个体的拥挤距离进行排序, 并选择拥挤距离大的个体进入下一次迭带。

算法选择拥挤距离较大的个体进入下一代, 当一个拥挤距离较小的个体淘汰之后, 会造成其他个体的拥挤距离变化。从而不能真实地反映个体之间真实的密度关系, 如图 1 所示, 在截断算法运行时存在 10 个 Pareto 解集, 对其进行排序, 对其中的拥挤距离较小的个体进行截断, 从而造成框中没有个体存在, 如图 2 所示, 其解集分布并不均匀。

为解决此问题, 本文加入了一个动态的更新算子, 使加入外部集的个体的拥挤距离最大, 从而使种群保持较好的分布度。

**基金项目:** 海南省教育厅高校科研基金资助项目(Hj2008-117)

**作者简介:** 汪文彬(1970-), 男, 讲师、硕士, 主研方向: 进化计算; 钟声, 教授、博士

**收稿日期:** 2008-10-10 **E-mail:** wwb0419@126.com

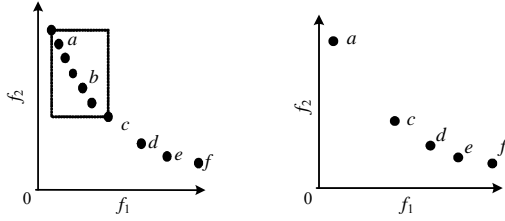


图1 须截断的 Pareto 解集 图2 算法1产生的精英集

### 算法2 基于改进拥挤距离的截断算法

**Step1** 初始化每个个体的拥挤距离,  $P[i]_{distance}=0$ , 对每个目标进行排序, 根据式(1)计算每个个体的拥挤距离;

**Step2** 将边界点赋予最大值以确保进入下一代, 查找精英集中拥挤距离最小的个体  $P$ , 并将其从精英集中淘汰;

**Step3** 重新更新个体的拥挤距离(更新每个个体的拥挤距离大小时, 记录其每个目标的最近距离);

**Step4** 精英集满足预定大小后停止。

由于改进的算法采用了动态的更新算子, 每淘汰一个个体重新计算与其相邻的个体, 从而使拥挤距离能够真实地反映出个体之间的密度关系, 其最终的解集如图3所示。

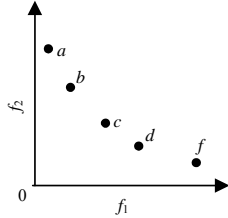


图3 算法2产生的精英集

## 3 动态变异算子

多数遗传算法采用交叉、变异作用于对应其搜索空间的二进制编码, 完成对空间的搜索。采用二进制编码的多目标进化算法<sup>[4-5]</sup>, 其变异算子都采用了比较小的概率(一般而言是其串长度的倒数), 相对于巨大的搜索空间, 其交叉、变异的手段很难搜索到所有潜在解的空间, 将会使算法陷入到局部最优。现今大多数的多目标进化算法采用了类似于贪婪策略的选择手段, 具有代表性的有 Pareo rank<sup>[2]</sup>, Pareto count<sup>[3]</sup>, 将适应值较大的个体优先选入进化池中, 这类个体将具有较大的概率将结构保存下来, 从而使种群陷入到局部最优中, 因此, 加大其变异概率, 完成其对空间的广度搜索(exploration)就成为算法避免陷入到局部最优的一个手段。但是较大的变异概率将打破个体的子结构, 使之无法进行空间的开发(exploitation), 因此, 本文设计了一种根据种群中的非支配解的数目进行变异的策略。

在本文中根据种群中的非支配个体对变异概率进行自适应调整, 其策略为当种群非支配集所占比率不大时, 加大其变异概率, 从而使算法能够搜索解空间, 也用以避免适应值较大的个体的子结构占据种群而陷入局部最优, 当种群的非支配集较多时, 对空间进行开发。

对每一个个体变异的概率取值为

$$control = \lfloor nds \rfloor / popsize \quad (2)$$

如果  $control$  小于 0.9, 则  $p\_mutation = 1 - control$ ; 否则:

$$p\_mutation = 1 / bitlength \quad (3)$$

变异步骤如下:

**Step1** 对种群中每一个个体, 根据式(2)计算其是否变异;

**Step2** 随机选一个变量的基因串;

**Step3** 对所选基因串的每一位根据式(3)进行变异。

在此选择一个变量的基因串的原因是在算法早期其变异的概率比较大, 而对其所有的基因串进行变异会使得个体的平衡完全打破, 因此, 本文选择个体的一个子结构(变量的基因串)。

## 4 数值实验

为测试本文算法的性能, 采用如下 5 个测试函数<sup>[4]</sup>:

(1)ZDT1: 决策空间为  $x \in [0, 1]^{30}$ , 目标函数为

$$\begin{aligned} \min f_1(x) &= x_1 \\ \min f_2(x) &= g(x) \left( 1 - \sqrt{x_1/g(x)} \right) \end{aligned}$$

$$g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

(2)ZDT2: 决策空间为  $x \in [0, 1]^{30}$ , 目标函数为

$$\begin{aligned} \min f_1(x) &= x_1 \\ \min f_2(x) &= g(x) \left( 1 - (x_1/g(x))^2 \right) \end{aligned}$$

$$g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

(3)ZDT3: 决策空间为  $x \in [0, 1]^{30}$ , 目标函数为

$$\begin{aligned} \min f_1(x) &= x_1 \\ \min f_2(x) &= g(x) \left( 1 - \sqrt{x_1/g(x)} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right) \end{aligned}$$

$$g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

(4)ZDT4: 决策空间为  $x_1 \in [0, 1], x_i \in [-5, 5]^9$ , 目标函数为

$$\begin{aligned} \min f_1(x) &= x_1 \\ \min f_2(x) &= g(x) \left( 1 - \sqrt{(x_1/g(x))} \right) \\ g(x) &= 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \end{aligned}$$

(5)ZDT6: 决策空间为  $x_i \in [0, 1], x = (x_1, x_2, \dots, x_m)$ , 目标函数为

$$\begin{aligned} f_1(x_1) &= 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ g(x_2, x_3, \dots, x_m) &= 1 + 9 \cdot \left( \sum_{i=2}^m x_i / (m-1) \right)^{0.25} \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned}$$

SPEA2 和 NSGA2 算法采用二进制编码, 运行 20 次, 每次迭代 250 代。

### 4.1 算法性能评价标准

本文用以下 2 项指标来评价算法的性能<sup>[5]</sup>。对于所测试的问题, 假设算法所得的非劣最优目标域为  $Z$ , 理论上的非劣最优目标域为  $Z'$ 。

(1)收敛性指标: 算法的收敛性可以通过式(4)计算  $Z$  和  $Z'$  间的最小距离的平均值来度量。

$$\gamma = \frac{1}{|Z|} \sum_{z \in Z} \min \{ \|z - \bar{z}\|, \bar{z} \in Z' \} \quad (4)$$

(2)多样性指标: 多样性可以通过式(5)来度量。

$$\Delta = \frac{d_f + d_i + \sum_{i=1}^{|Z|-1} |d_i - \bar{d}|}{d_f + d_i + (|Z|-1)\bar{d}} \quad (5)$$

其中,  $d_i$  是非劣最优目标域  $Z$  中连续 2 个非劣解向量间的欧氏距离;  $\bar{d}$  是所有  $d_i$  的均值;  $d_f$  和  $d_i$  分别表示非劣最优目标域  $Z'$  中极值目标向量和非劣最优目标域  $Z$  边界向量间的距离。

## 4.2 实验结果及分析

收敛性指标  $\gamma$  的值如表 1 所示。可以看出, 本文提出的算法对 5 个 ZDT 测试函数在收敛性方面表现的都比其他 2 种算法好, 尤其是对于 ZDT4, ZDT6 这 2 个比较难以收敛的函数具有很明显的优势, 说明算法能够较有效地摆脱局部最优面的吸引, 对解空间能够执行有效的搜索。对于其他函数优势不是很明显, 主要原因是这 2 个函数的优化求解相对来说

不是很困难。

多样性指标  $\Delta$  的值如表 2 所示。可以看出, 算法对 Pareto 前沿面的多样性能保持是非常有效的, 优于这 2 个经典的算法, 特别是对 NSGA2 算法在各个测试函数中都占优, 这说明了改进的拥挤距离的保持策略要大大优于原有的拥挤距离。综上所述, 本文提出的算法在评价 MOEA 的上述 2 个标准时都表现得很好。

表 1 收敛性指标  $\gamma$  的均值与偏差

算法	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6	
	均值	偏差	均值	偏差	均值	偏差	均值	偏差	均值	偏差
SPEA2	0.023 285	0	0.167 620	0.000 815	0.018 409	0.000 000	4.927 100	2.703 000	0.232 551	0.004 945
NSGA2	0.000 894	0	0.000 824	0.000 000	0.043 411	0.000 042	3.227 636	7.307 630	7.806 798	0.001 667
本文算法	0.004 781	0	0.001 015	0.000 000	0.019 870	0.000 000	0.437 910	0.003 410	0.015 684	0.003 214

表 2 多样性指标  $\Delta$  的均值与偏差

算法	ZDT1		ZDT2		ZDT3		ZDT4		ZDT6	
	均值	偏差	均值	偏差	均值	偏差	均值	偏差	均值	偏差
SPEA2	0.154 72	0.000 68	0.339 45	0.001 75	0.469 10	0.005 265	0.823 90	0.002 883	1.044 220	0.158 106
NSGA2	0.463 29	0.041 62	0.435 12	0.026 07	0.575 60	0.005 070	0.479 47	0.009 840	0.644 477	0.035 042
本文算法	0.146 62	0.032 31	0.301 43	0.002 98	0.245 81	0.006 560	0.451 12	0.024 650	0.477 170	0.023 236

## 5 结束语

本文提出的基于改进的拥挤距离多目标进化算法采用了分布度保持策略和自适应的变异算子, 相比经典的多目标进化算法具有更好的性能, 实验说明了其有效性。下一步的研究方向是把该算法应用于解决多目标离散优化问题上。

### 参考文献

- [1] Coello C, Carlos A. Evolutionary Algorithms for Solving Multiobjective Problems[M]. [S. l.]: Kluwer Academic/Plenum Publishers, 2002.
- [2] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II[J]. IEEE Trans. on Evolutionary Computation, 2002, 6(2): 182-197.

- [3] Zitzler E. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization[C]//Proceedings of EUROGEN'01. Athens, Greece: [s. n.], 2001.
- [4] Kukkonen S, Lampinen J. An Extension of Generalized Differential Evolution for Multiobjective Optimization with Constraints[C]//Proceedings of the 8th International Conference on Parallel Problem Solving from Nature. Birmingham, UK: [s. n.], 2004.
- [5] Zitzler E, Deb K, Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results[J]. Evolutionary Computation, 2000, 8(2): 173-195.

编辑 顾姣健

(上接第 210 页)

运动的过程中, 马达声也随之由左耳向右耳移动, 开始的时候左耳音量大于右耳音量, 随着与听者之间位置的变化, 两耳的音量先增大再减小, 当小车运动到听者右边, 右耳的音量开始高于左耳。整个过程一共播放了 7 个声音, 符合 XML 文档中的描述。测试结果显示, 该系统可以实现包括碰撞发声在内的多种音效, 对多声源播放中具体某个声音能够进行很好的控制。

为了检测有没有加载多余声音文件, 继续载入“训练”场景, 但是在程序中播放“迷宫”场景中的声音。测试结果是, 声音文件未能找到, 可见“迷宫”场景中的声音文件没有被加载, 有效地节约了计算机资源。

在测试过程中发现, 小车的多普勒效应不是很明显, 可能是速度较低的原因。该声音系统还有许多需要改进的地方: 比如添加 DSP 处理功能, 这样可以实现丰富的数字音效; 自动为声音设置 FMODE\_MODE 类型, 可以方便不熟悉 FMODE 的用户使用; 能够实时切换各种音效的开/关状态, 以满足不同用户的需要。

## 5 结束语

本文介绍了一种基于 FMODE 引擎的声音系统, 采用 XML 技术解决了声音资源加载中的问题, 实现了包括碰撞发声在内的多种音效; 通过对播放通道的管理, 实现了对多声源的

管理。该系统能够满足教育机器人物理仿真对声音的需求, 为 FMODE 声音引擎在其他仿真环境中的应用提供了参考。

### 参考文献

- [1] Pohl E D. CROBOTS: A Robot Kinematic Simulator and Animator[C]//Proc. of the 1990 ASME International Computers in Engineering Conference and Exposition. Boston, USA: [s. n.], 1990: 355-360.
- [2] Cook G E, Biegl C, Springfield J F, et al. An Intelligent Robotics Simulator[C]//Proc. of the 29th IAS Annual Meeting. Piscataway, USA: IEEE Press, 1994: 1793-1800.
- [3] Olivier M. Webots™: Professional Mobile Robot Simulation[J]. International Journal of Advanced Robotic Systems, 2004, 1(1): 39-42.
- [4] 李鸣华. 面向中小学的虚拟机器人教学平台设计[J]. 中国教育信息化, 2008, (4): 58-61.
- [5] 沈靖, 吕恬生. 教学机器人三维仿真与远程控制软件实现[J]. 计算机仿真, 2006, 23(10): 289-292.
- [6] 宋洪军, 马昕, 李贻斌, 等. 教育机器人三维软件系统的设计与实现[J]. 山东大学学报, 2007, 37(4): 34-38.
- [7] 丁跃潮, 张涛. XML 实用教程[M]. 北京: 北京大学出版社, 2006.

编辑 任吉慧

