

文章编号: 1001-0920(2007)11-1317-04

多目标函数优化的元胞蚂蚁算法

朱刚^{1,2}, 马良¹

(1. 上海理工大学 管理学院, 上海 200093; 2. 上海第二工业大学 电子电气工程学院, 上海 201209)

摘要: 提出一种求解多目标函数优化的元胞蚂蚁算法. 该方法将元胞自动机演化规则引入蚂蚁算法, 给出了在连续空间多目标函数优化的算法描述, 定义了与蚂蚁信息素释放有关的元胞演化规则及蚂蚁邻域的转移概率, 并实现了算法的具体步骤. 在 Matlab 环境下, 采用该算法对一些典型的测试函数进行求解和验证. 实验结果表明, 该方法具有向真实的 Pareto 前沿逼近的效果, 是一种求解多目标优化的有效方法.

关键词: 蚂蚁算法; 元胞自动机; 多目标函数优化

中图分类号: O22 **文献标识码:** A

Cellular ant algorithm for multi-objective function optimization

ZHU Gang^{1,2}, MA Liang¹

(1. College of Management, University of Shanghai for Science and Technology, Shanghai 200093, China; 2. College of Electronic and Electric Engineering, Shanghai Second Polytechnic University, Shanghai 201209, China. Correspondent: ZHU Gang, E-mail: zhug123@sh163.net)

Abstract: A cellular ant algorithm for solving multi-objective function optimization is presented. The proposed algorithm introduces the rule of cellular automata into the ant algorithm and gives a cellular ant algorithm with its mathematical description that can be used for solving the multi-objective function optimization problem of continuous systems. The rule of cellular automata with ants pheromone and the transition probability are described. The algorithm is coded in Matlab, by which the series of typical problem instances are tested. The simulation results show that the algorithm can efficiently reach the true Pareto frontier.

Key words: Ant algorithm; Cellular automata; Multi-objective function optimization

1 引言

近年来,随着智能优化方法在各应用领域取得成功,用其求解复杂的多目标优化问题已成为研究热点.很多学者使用遗传算法^[1-3]和蚂蚁算法^[4]对求解多目标问题的 Pareto 最优解进行了深入地探讨.

蚂蚁算法是一种源于自然界中的仿生类算法^[5],已在一系列困难的组合优化问题求解中取得了成效.然而在求解多目标函数优化问题方面还处于探索阶段,有关此类问题的论文甚少.对于多目标优化问题,不仅要求所得的解能收敛到 Pareto 前沿,而且要有效地保持群体的多样性.

元胞自动机是冯诺伊曼^[6]最早提出的,沃尔夫勒姆等^[7]将动力系统方法和计算理论及形式语言方法用于元胞自动机的研究中,促进其广泛应用.元胞空间内的每个元胞变化取决于其周围半径为 r 的邻

居中的元胞,它们遵循同样的演化规则,作同步局部更新.大量的元胞通过简单的相互作用而构成动态系统的演化.

针对多目标优化的特殊性,本文基于蚂蚁算法和元胞自动机的原理,提出一种求解多目标连续系统优化问题算法——元胞蚂蚁算法.蚂蚁在搜索区域内按元胞自动机的演化规则进行局部优化,再按不同情况释放不同的信息素;区域之间蚂蚁通过信息素的交流,进行优化搜索.这样既能达到 Pareto 前沿,又能保证解的多样性.最后对部分典型的二元的多目标连续函数优化问题进行了测试和验证.

2 多目标函数优化的元胞蚂蚁算法

在科学管理与经济决策的许多应用领域中,有大量的多目标优化问题.对很多决策而言,需优化的目标往往不止一个.如何在多个目标中寻找折衷最

收稿日期: 2006-07-23; 修回日期: 2006-10-30.

基金项目: 国家自然科学基金项目(70471065); 上海市重点学科建设项目(T0502).

作者简介: 朱刚(1963—),男,上海人,副教授,博士生,从事系统工程、智能优化的研究; 马良(1964—),男,上海人,教授,博士生导师,从事系统工程、智能优化的研究.

优解是一个比较复杂的问题。

在多目标优化问题中,由于决策者对目标的选择出于不同目的,多个目标往往不可能同时达到最优,有时甚至是相反的。此外,不同决策者由于对各个目标的偏好不同,也会得到不同的最优解集。常用的优化技术本身并不能直接处理多目标优化问题,往往将其转化为较容易处理的单目标或双目标问题。典型的方法有:主要目标法、线性加权法、平均和加权法、理想点法、乘除法和几何平均法等^[8]。这些优化技术一般每次只能得到 Pareto 解集中的一个,或是相对于 Pareto 最优解的一个折衷解。而 Pareto 解往往是多目标优化问题的一个解集,因此群体搜索策略是非常合适的解决方案,其每次搜索的结果是一群可行解,能更好地逼近 Pareto 最优解集。

一个典型的带约束函数优化问题可使用常规的罚函数法将所有约束方程转入目标函数中。这样,在算法的具体搜索过程中就不必考虑约束是否满足的问题,而直接考虑目标函数优化。

蚂蚁算法中蚂蚁个体之间是通过信息素进行信息传递的。一个孤立的蚂蚁随机移动时,能检测到其他同伴所释放的信息素,并沿着该路线移动,同时又释放自身的信息素,从而增强了该路线上的信息素数量。随着越来越多的蚂蚁通过该路线,一条最佳的路径就会逐渐形成。

元胞自动机是一个时间和空间都离散的动力系统,由元胞、元胞空间、邻居及规则4部分组成。元胞自动机可视为由一个元胞空间和定义于该空间的变换函数组成。它可模拟复杂结构和过程的模型。

元胞是元胞自动机最基本的组成部分,元胞空间是元胞分布空间的网点集合。针对多目标的函数优化,元胞蚂蚁算法将元胞自动机的元胞、元胞空间及元胞邻居与蚂蚁算法的区域搜索相结合。对于连续函数 $f_i(x_1, x_2, \dots, x_n), i = 1, 2, \dots, k$, 其定义域集合为 n 维欧氏空间,可将其中的任一元素视为元胞,而所有的元素构成元胞空间。

元胞状态的局部演化是一个时间函数,元胞下一时刻的状态由元胞的当前状态和其邻居的当前状态决定。元胞邻居定义有多种方法,本文选用扩展的 Moore 邻居类型。蚂蚁搜索的区域为元胞空间中某个元胞及其扩展的 Moore 邻居,其中邻居的半径 r 的值取决于函数定义域的取值范围和蚂蚁的数量。

蚂蚁在搜索区域内释放信息素,而这些信息素可被整个元胞空间中其他蚂蚁所感知。蚂蚁在搜索区域内按元胞自动机的演化规则进行局部优化,即按不同情况释放不同的信息素。

定义1 元胞演化规则如下:

(1) 初始化:选择区域 N_i 的中心元胞 c_{i0} , 计算

$$Z_{i0pt} = (f_1(c_{i0}), f_2(c_{i0}), \dots, f_k(c_{i0})),$$

比较所有区域得元胞空间的初始最佳值 Z_{i0pt} 。

(2) 选取区域 N_i 中任意元胞 c_i , 计算

$$Z_i = (f_1(c_i), f_2(c_i), \dots, f_k(c_i)).$$

1) 若 $Z_i = Z_{i0pt}$, 不释放信息素。

2) 若 $Z_i < Z_{i0pt}$, 替换区域的最佳值。

若 $Z_i < Z_{i0pt}$, 替换空间最优值, 释放信息素

Q :

若 Z_i 与 Z_{i0pt} 无法比较, 则将其作为新的最优值保存, 并释放信息素 Q ;

若 $Z_i = Z_{i0pt}$, 释放信息素 $Q/2$ 。

3) 若 Z_i 与 Z_{i0pt} 无法比较, 则:

若 $Z_i < Z_{i0pt}$, 替换空间最优值, 释放信息素

Q ;

若 Z_i 与 Z_{i0pt} 无法比较, 则将其作为新的最优值保存, 并释放信息素 Q ;

若 $Z_i = Z_{i0pt}$, 释放信息素 $Q/2$ 。

蚂蚁的寻优方向与空间中其他区域的信息素浓度以及它们之间的目标函数的平均差异程度有关。

定义2 蚂蚁邻域的转移概率定义为

$$P_{ij} = \frac{[i_j][j_i]}{[i_k][j_k]} \quad (1)$$

其中: j 为区域 j 的信息素强度, i_j 为目标函数平均差异程度

$$i_j = \sqrt{\frac{1}{Nk} \sum_{t=1}^k (f_t(c_i) - f_t(c_j))^2}, \quad (2)$$

为邻域吸引强度的相对重要性(> 0), 为目标函数平均差异的相对重要性(> 0)。

定义3 信息素强度更新方程为

$$j_j^{\text{new}} = j_j^{\text{old}} + k \frac{j_j}{j_j} \quad (3)$$

其中: j_j^k 为第 k 个蚂蚁在区域 N_j 每迭代一次所释放的信息素, 体现强度的持久性。

该算法的主要步骤如下:

Step1: $nc = 0$ (nc 为迭代步数或搜索次数), 将 j 和 j 初始化, 并按蚂蚁个数及空间的大小确定区域的大小, 将 n 个蚂蚁置于 m 个搜索区域的中心, 元胞演化规则初始化;

Step2: 对每个蚂蚁 $k (k = 1, 2, \dots, n)$ 进行区域搜索, 计算各蚂蚁的目标函数值 Z , 按元胞演化规则释放信息素, 更新当前的最好解或添加新的最好解;

Step3: 按更新方程修改轨迹强度;

Step4: 计算转移概率 P_{ij} , 确定蚂蚁是否转移;

Step5: 置 $j = 0; nc = nc + 1$;

Step6: 若 $nc <$ 预定的迭代次数且无退化行为

(即找到的都是相同解), 则转 Step2;

Step7: 输出目前的最好解集.

随着蚂蚁在区域内信息素的释放, 依据蚂蚁的转移概率, 蚂蚁将集中于非劣解较多的区域, 获取更多的非劣解; 同时蚂蚁在不同的区域内搜索, 使所得解有一定的分布度.

元胞蚂蚁算法的计算时间复杂度与元胞空间的划分和每个区域的蚂蚁个数有关. 为降低计算时间复杂度, 在处理实际问题时, 采用均衡划分元胞空间, 且每个区域的蚂蚁个数不超过总区域数, 即 $n \leq m$, 则此时算法的时间复杂度为 $O(nc * n^2 * m^2)$ $O(nc * m^4)$ (nc 为迭代次数); 若每个区域放一个蚂蚁, 则时间复杂度为 $O(nc * m^3)$, 与传统的蚂蚁算法相似.

3 实例验证

为验证方法的有效性, 本文求解了一系列算例, 这里仅给出 3 组二元目标优化测试函数及其计算效果. 在每个测试实例中, 以图形化的方式给出了该方法生成的 Pareto 前沿, 并对解的分布性能及散布范围^[9] 进行比较.

定义 4 间距评估

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}. \quad (4)$$

其中

$$d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|),$$
$$i, j = 1, 2, \dots, n,$$
$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i.$$

n 为算法获得 Pareto 前沿上向量的个数, 当所获得的解越接近均匀散布时, 间距 S 的值越小.

定义 5 最大散布范围评估

$$D = \sqrt{\max_{i=1}^n f_1^i - \min_{i=1}^n f_1^i)^2 + (\max_{i=1}^n f_2^i - \min_{i=1}^n f_2^i)^2}. \quad (5)$$

该评估定义了两个极值解的距离, 其值越大, 表明算法的解散布范围越广.

表 1 给出了这 3 组测试函数, 算法用 Matlab 编程实现并进行了计算测试. 测试中所用的参数为 $\alpha = 1, \beta = 0.7, Q = 1$. 元胞的区域分割数为 10, 则每个实例为 100 个区域. 每个区域的中心为元胞, 其余部分为元胞邻居, 而区域之外的部分为该区域的邻域.

编号 1 的测试函数迭代 200 次后的求解结果如图 1 所示. 在每个区域放 5 个蚂蚁的情况下, 获得 Pareto 解的个数为 5 709 个, 解的间距为

表 1 测试函数

编号	函 数
1 ^[1]	$\min f_1(x) = \begin{cases} -x, & x < 1 \\ -2+x, & 1 < x < 3 \\ 4-x, & 3 < x < 4 \\ -4+x, & x > 4 \end{cases}$ $\min f_2(x) = (x-5)^2, \quad -5 \leq x \leq 10$
2 ^[4]	$\min f_1(x) = 4x_1^2 + 4x_2^2$ $\min f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$ s. t. $e_1(x) = (x_1 - 5)^2 + x_2^2 \leq 25$ $e_2(x) = (x_1 - 8)^2 + (x_2 + 3)^2 \leq 7.7$ $0 \leq x_1 \leq 5, 0 \leq x_2 \leq 3$
3 ^[4]	$\min f_1(x) = x_1 \quad \min f_2(x) = x_2$ s. t. $e_1(x) = x_1^2 + x_2^2 - 1 \leq 0$ $e_2(x) = 0.1 \cos(16 \arctan(x_1/x_2)) - 0.5 \leq 0$ $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$ $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$

0.001 135 2, 最大散布范围为 2.827 1. 表 2 给出了与文献[1]中算法的间距的比较结果. 从表 2 可见, 本算法的结果优于文献[1]中所提的算法.

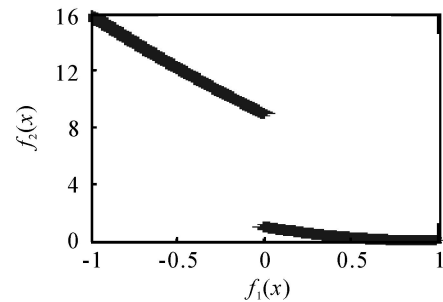


图 1 编号 1 函数求解结果

表 2 编号 1 的函数与文献[1]中算法在间距上的比较

算法	NSGA	PAES	Micro-GA	本文算法
间距 S	0.001 641	0.067 520	0.001 530	0.001 135

编号 2 的测试函数的求解结果如图 2 所示. 测试初始状态为每个区域放 5 个蚂蚁, 算法迭代次数为 10. 获得 Pareto 解的个数为 468 个, 解的间距为

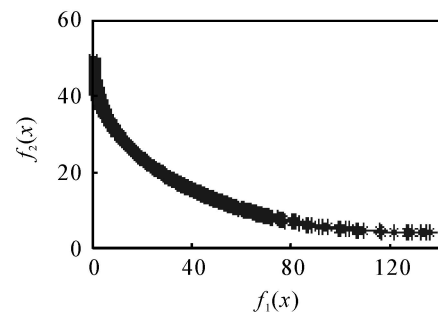


图 2 编号 2 函数求解结果

0.331 61,最大散布范围为 191.表 3 中数据表明,元胞蚂蚁算法的结果均优于文献[4]的结果,能很好地逼近 Pareto 的前沿.

表 3 编号 2 的函数与文献[4]中算法的比较

算法	迭代次数	解的个数	间距 s	最大散布范围 D
NSGA	10	97	0.85	115.44
文献[4]算法	10	238	0.37	142.59
本文算法	10	468	0.33	191

图 3 给出了编号 3 的测试函数在每个区域放 5 个蚂蚁且迭代 200 次后的求解结果.获得 Pareto 解的个数为 98 个,解的间距为 0.008 6,最大散布范围为 1.429 6.表 4 中数据表明,本文算法和文献[4]的结果较为接近,都能较好地逼近 Pareto 的前沿.

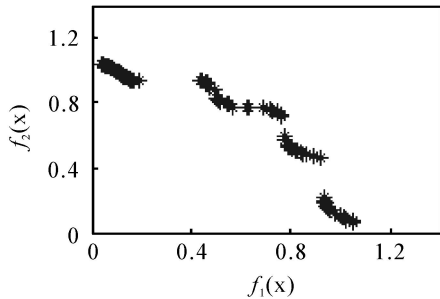


图 3 编号 3 函数的求解结果

表 4 编号 3 的函数与文献[4]中算法的比较

算法	迭代次数	解的个数	间距 s	最大散布范围 D
NSGA	100	78	0.007 8	1.21
文献[4]算法	100	136	0.005 1	1.41
本文算法	200	98	0.008 6	1.43

4 结 语

本文基于元胞自动机的区域演化规则,提出了用于求解多目标函数优化的元胞蚂蚁算法.将演化规则与信息素释放相结合,可更好地引导蚂蚁的行走方向.算例求解与测试结果表明,本文算法能很好地逼近 Pareto 前沿,同时具有一定的散布广度.

参考文献(References)

[1] Coello C A , Toscano Pulido G . Multiobjective

optimization using a micro-genetic algorithm[C]. Proc of the Genetic and Evolutionary Computation Conf. San Francisco: Morgan Kaufmann Publishers, 2001: 274-282.

[2] Mu Sheng-jing, Su Hong-ye, Wang Yue-xuan, et al. An efficient evolutionary multi-objective optimization algorithm [C]. Proc of the IEEE Congress on Evolutionary Computation. Canberra: IEEE Press, 2003: 914-920.

[3]王跃宣,刘连臣,牟盛静,等.处理带约束的多目标优化进化算法[J].清华大学学报,2005,45(1):103-106. (Wang Yue-xuan, Liu Lian-chen, Mu Sheng-jing, et al. Constrained multi-objective optimization evolutionary algorithm[J]. J of Tsinghua University, 2005, 45(1): 103-106.)

[4]张勇德,黄莎白.多目标优化问题的蚁群算法研究[J].控制与决策,2005,20(2):170-173. (Zhang Yong-de, Huang Sha-bai. On ant colony algorithm for solving multiobjective optimization problems[J]. Control and Decision, 2005, 20(2): 170-173.)

[5]Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [C]. Proc of the 1st European Conf on Artificial Life. Paris: Elsevier Publishing, 1991: 134-142.

[6] Von Neumann J, Burks A W. Theory of self-reproducing automata [M]. Urbana: University of Illinois Press, 1966.

[7] Wolfram S. Theory and application of cellular automata [M]. Singapore: The World Scientific Publishing Company Limited, 1986.

[8]《运筹学》教材编写组.运筹学[M].北京:清华大学出版社,2005:440-447.

(The Group of "Operations Research". Operations research [M]. Beijing: Tsinghua University Press, 2005: 440-447.)

[9] Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.

(上接第 1316 页)

[5] Hu X, Eberhart R C. Multiobjective optimization using dynamic neighborhood particle swarm optimization[C]. Proc of the IEEE Congress on Evolutionary Computation. Honolulu, 2002: 1677-1681.

[6] Joanna L, Eiben A E. A multi-sexual genetic algorithm for multiobjective optimization[C]. Proc of the 1996 Int

Conf on Evolutionary Computation. Nagoya, 1996: 59-64.

[7] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results [R]. Zurich: Swiss Federal Institute of Technology, 1999.