

基于佳点集的多目标遗传算法

饶玉佳,程家兴,夏 军,李志俊

(安徽大学 计算机科学与技术学院,安徽 合肥 230039)

摘 要:佳点集遗传算法是利用数论中的佳点集的理论,对 GA 算法中的交叉操作进行了重新设计的 GA 算法。该算法不仅提高了求解的效率和精度,还能有效地避免“早熟”现象。对其部分机制作出改动,使其可以应用于多目标优化问题。并将其引入一个典型的多目标遗传算法 NPGA 中。文中给出的仿真算例证实了改进方法的有效性。

关键词:多目标优化;佳点集;NPGA

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2008)12-0067-03

Multi-Objective Optimization Genetic Algorithm Based on Good Point Set

RAO Yu-jia, CHENG jia-xing, XIA Jun, LI Zhi-jun

(Department of Computer Sci. & Techn., Anhui University, Hefei 230039, China)

Abstract: Good point set based GA is a GA which redesigned the crossover operation by using the principle of good point set in number theory. It has superiority in speed, accuracy and overcoming premature. In this paper, good point set based GA is modified to be applicable to Pareto multi-objective optimization problems. Then, it is introduced into NPGA (Niche Pareto Genetic Algorithm) to speed up the algorithm. The effectiveness of method is proved by the experiments.

Key words: multi-objective optimization problem; theory of good point set; NPGA

0 引 言

多目标优化问题^[1](MOP, Multi-objective Optimization Problem)经常出现于科学研究和工程实践中。多目标优化问题通常由多个目标组成,在问题的求解空间中各个目标之间往往是相互冲突的,因而多目标优化问题的最优解的定义较单目标优化问题解的定义要复杂。一般来说单目标优化问题的最优解可以是单个解,而多目标优化问题的解表现为一组均衡解或折衷解,即所谓的 Pareto 最优解集,解集中的解之间没有绝对的优劣之分,求解 MOP 问题的主要目的是要求到一组解。由于 MOP 问题的求解是一个较为困难的问题,所以在实际解决此类优化问题时,结果往往是一组近似 Pareto 最优解集。

遗传算法(GA)在单目标优化问题的解决上获得了成功,实践证明,以遗传算法求解多目标优化问题也是最有效的方法之一。

GA 能产生问题的群体解(而且常是在 Pareto 集中均匀分布的一组解),已显示出较古典方法优越得多的本质优点。近年来,已经提出了很多的多目标优化遗传算法^[2,3]。

在以遗传算法求解时,传统的交叉运算,是随机取两个染色体进行单点交叉操作(也可用其它的交叉操作,如多点交叉、树交叉以及部分匹配交叉、顺序交叉、周期交叉等),其交叉算子都是随机操作,后代简单地继承了“父亲”的一部分基因和“母亲”的一部分基因,并不能保证子代的性能优于父辈,而且以这种方式点对点的搜索范围有限,可能会忽略邻域内更好的点,而使结果收敛于局部最优。张玲、张钹教授利用数论中的佳点集理论和方法^[4],设计了一个新的交叉算法^[5],以提高 GA 算法的效率。实验证明了该算法在单目标优化方面的优越性。

文中利用其原理,在交叉操作后的 n 后代选择机制以及原有的染色体池劣者逐步淘汰的机制上做出修改,使其可以应用于多目标优化问题。并将其引入一个典型的多目标遗传算法 NPGA 中,与原有的 NPGA 的计算结果做出试验比较。证明了基于佳点集的交叉操作对于多目标优化问题的适用性和优越性。并认为其

收稿日期:2008-03-16

基金项目:教育部博士点基金(200403057002)

作者简介:饶玉佳(1984-),男,安徽安庆人,硕士研究生,研究方向为智能计算;程家兴,博导,教授,研究方向为智能计算。

它的一些典型多目标遗传算法也可以利用此算子来提高运算性能。

1 基于佳点集的多目标遗传算法

以下是佳点集的介绍以及具体的基于佳点集的遗传操作的描述。

1.1 佳点集的基本定义与性质

(1) 设 G_s 是 s 维欧氏空间中的单位立方体, 即 $x \in G_s, x = (x_1, x_2, \dots, x_s)$, 其中 $0 \leq x_i \leq 1, i = 1, 2, \dots, s$ 。

(2) 设 G_s 中有一点集 (n 个点), $P_n(k) = \{(x_1^{(n)}(k), \dots, x_s^{(n)}(k)), 1 \leq k \leq n\} \cap \{0 \leq x_i^{(n)}(k) \leq 1, 1 \leq i \leq s\}$ 。

(3) 对任一给定 G_s 中的点 $r = (r_1, r_2, \dots, r_s)$, 令 $N_n(r) = N_n(r_1, r_2, \dots, r_s)$ 表示 $P_n(k)$ 中满足不等式组 $0 \leq x_i^{(n)}(k) \leq r_i$ (其中 $i = 1, 2, \dots, s$) 的点的个数。
$$\phi(n) = \sup_{r \in G_s} \left| \frac{N_n(r)}{n} - |r| \right|$$
, 其中 $|r| = r_1 r_2 \dots r_s$, 则称点集 $P_n(k)$ 有偏差 $\phi(n)$ 。若对任一 n , 均有 $\phi(n) = O(1)$, 则称 $P_n(k)$ 在 G_s 上是一致分布。

(4) 令 $r \in G_s, \epsilon$ 是任意小的正数, 形为 $P_n(k) = \{(\{r_1 * k\}, \dots, \{r_s * k\}), k = 1, 2, \dots, n\}$ 的偏差满足 $\phi(n) = C(r, \epsilon) n^{-1+\epsilon}$, 其中 $C(r, \epsilon)$ 是只与 r, ϵ 有关的常数, 则称 $P_n(k)$ 为佳点集, r 称为佳点。

(5) 取 $r_k = \{2 \cos \frac{2\pi k}{\rho}\} 1 \leq k \leq s; \rho$ 是满足 $(\rho - s)/2$ 的最小素数, 则 r_k 是佳点; 或取 $r_k = \{e^k\} 1 \leq k \leq s$, 则 r_k 是佳点。

1.2 佳点集交叉操作与染色池淘汰机制

原有的交叉操作: 取两个染色体 A, B 进行交叉操作, 方法如下:

设 $A = a_1 a_2 \dots a_L, B = b_1 b_2 \dots b_L$, 交叉得 $C = c_1 c_2 \dots c_L$, 其中

$$c_k = \begin{cases} a_k & \text{若 } a_k = b_k \\ * & \text{若 } a_k \neq b_k \quad (* \text{ 表示待定}) \end{cases}$$

计算一下 $*$ 的个数, 设其为 S , 用到佳点集 G_s 中去。

在 S 维空间中作含有 n 个点的佳点集 $p_n(i) = \{(\{r_1, i\} \dots \{r_s, i\}), 1 \leq i \leq n\}$

其中 $r_1 \dots r_s$ 可按第(5)步去取, 若 $\{a\} < 0.5$, 则取 0, 否则取 1。

在其“家族”中产生出 n 个后代, 取其中适应度最大者(或最大的几个)作为交叉操作后的子代。

原有的淘汰机制:

把经过遗传操作(交叉、变异)后得到的染色体都

放到染色体池中, 对新得到的染色体, 计算其适应度值, 若假定染色体池的容量一定, 当染色体的个数超过容量时, 就将适应度小的染色体从池中删去。

1.3 将基于佳点集的交叉操作引入多目标优化

将 1.2 的方法引入多目标优化会产生问题:

在多目标优化中, 染色体个体之间比较计算较为复杂, 对于“当染色体的个数超过容量时, 就将适应度小的染色体从池中删去。”这种操作, 每次染色体池超过容量时, 就需要对染色池中所有个体进行一次级别排序, 删除级别低的个体。由于染色体池的容量一般较大, 所以对所有个体级别排序是十分花时间的, 之后当有新的染色体进入, 染色体池又会变满, 又要重新排序删除级别低的。总体上的计算量很可观。

修改使其适用于多目标优化:

对交叉操作的修改: 取两个染色体 A, B 进行交叉操作, 在其“家族”中产生出 n 个后代后, 对这 n 个后代的集合进行以下操作(n 为较小常数, 此操作计算量不大)。

操作如下:

(1) 确定 n 个后代中的 Pareto 最优个体;

(2) 如果 Pareto 最优个体数量等于两个, 取此两个, 退出; 否则, 继续;

(3) 如果 Pareto 最优个体数量大于两个, 任取其中的两个, 退出; 否则, 继续;

(4) 如果 Pareto 最优个体数量只有一个, 从群体中去掉此个体, 在剩下个体中确定 Pareto 最优个体;

(5) 从新的 Pareto 最优个体中任选一个, 与(4)中去掉的个体一起, 作为被选的子代。这样操作, 在两染色体 A, B 交叉运算的 n 个后代中, 选出两个最好个体, 直接放入下一代种群。避免了染色体池超过容量时的大量计算。

2 NPGA 算法简介

Jeffrey Horn 等提出了 NPGA(Niched Pareto Genetic Algorithm), 使用基于 Pareto 支配的锦标赛选择模式。

算法的基本思想非常巧妙: 从种群中随机选择两个个体, 与来自种群的一个子集比较(典型地, 子集占整个种群的 10%), 若其中一个个体优于子集中所有个体, 而另一个个体被子集支配, 则非劣个体获胜; 所有其它情况被视为一结(tie, 即互不支配)。当有结存在时, 通过适应度共享确定锦标赛结果。因为该算法的非劣最优解选择是基于种群的部分而非全体, 所以其优点是能很快找到一些好的非劣最优解域, 并能维持一个较长的种群更新期^[6]。

3 基于佳点集的 NPGA 算法流程

将修改后的交叉操作,引入多目标遗传算法 NPGA 中。

算法流程如下:

Step1:生成初始群体 $Pop(0)$,最大进化代数 T ,计数器 $t = 0$;

Step2:对当代群体 $Pop(t)$ 中每一个体,计算其各目标函数值。

Step3:采用 Pareto 支配的锦标赛方法,从当前群体选择个体,放入交配池;

Step4:在交配池中依次取出两个体,进行基于佳点集的交叉(见 1.3)和变异操作。将生成子代群体作为下一代的种群 $Pop(t+1)$;

Step5:中止准则。如果 $t \geq T$,输出 $Pop(t)$ 中的所有非支配个体,作为问题最终的 pareto 最优解集,算法中止;否则, $t = t + 1$, 转 step2。

4 实验

采用 shaffer 测试函数进行比较, minimize. $F = (f_1(x), f_2(x))$, 其中 $f_1(x) = x^2$, $f_2(x) = (x-2)^2$, 变量范围: $-10^3 \leq x \leq 10^3$ 。

算法群体大小:80,运行代数:40,交叉概率:0.8,变异率:0.1, niche 半径:0.2。

实验对比结果见图 1~3。

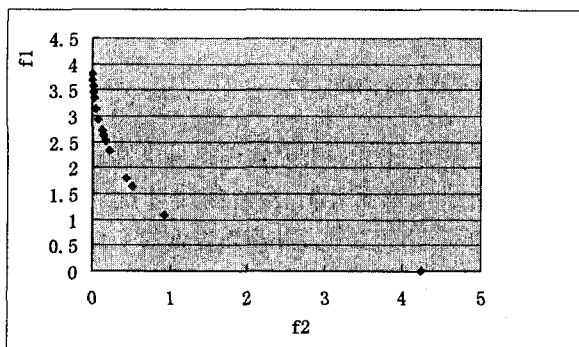


图 1 改进前的计算结果(第 20 代)

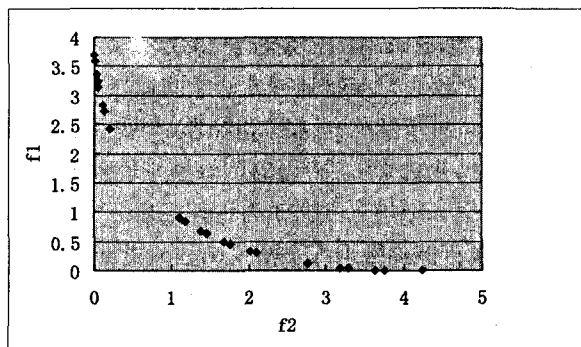


图 2 改进前的计算结果(第 40 代)

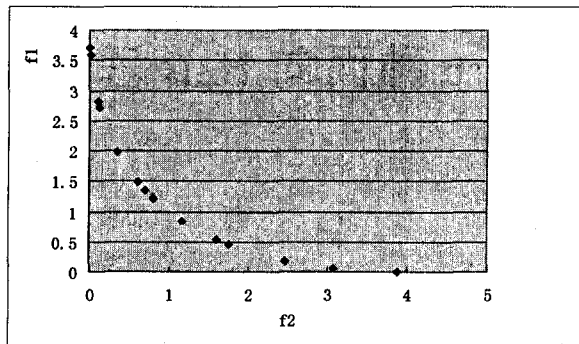


图 3 改进后的计算结果(第 20 代)

就本例而言,改进前的算法至少需要运行到 40 代以后才能算出达到要求的解集,而改进后的算法只需 20 代左右。利用佳点集改进算法后,提高了求解的效率。

此外,就非劣解前端的均匀性来讲,改进后的算法,求得的解集在分布上也比较均匀。

5 结束语

用佳点法求到的子集(在均值意义下)最能代表其“家族”性能,故用佳点法构造的交叉算法,从理论上看,更符合遗传算法的机理。其次,从钟开莱定理知,用佳点集法取 n 点比用随机法取 n 点,其偏差要小得多(差平方倍),这是佳点集方法收敛速度快的理论依据。将佳点集遗传算法的机制加以改动,将其引入 NPGA 算法的交叉操作中,经过实验表明,取得比原 NPGA 算法更好的性能(收敛明显加快,解的分布较均匀)。故将佳点集理论与其它的一些多目标优化遗传算法结合,能提供一个更加合理高效的交叉操作。

参考文献:

- [1] 谢涛,陈火旺,康立山. 多目标优化的演化算法[J]. 计算机学报,2003,26(8):997-1003.
- [2] 王小平,曹立明. 遗传算法——理论、应用与软件实现[M]. 西安:西安交通大学出版社,2002:115-122.
- [3] 崔逊学. 多目标进化算法及其应用[M]. 北京:国防工业出版社,2006.
- [4] 华罗庚,王元. 数论在近代分析中的应用[M]. 北京:科学出版社,1978.
- [5] 张铃,张钊. 佳点集遗传算法[J]. 计算机学报,2001,24(9):917-922.
- [6] Horn J, Nafpliotis N, Goldberg D E. A Niche Pareto Genetic Algorithm for Multiobjective Optimization[C] // In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence. Piscataway, New Jersey:[s.n.], 1994:82-87.