

改进遗传算法与粒子群 优化算法及其对比分析

任斌, 丰镇平

(西安交通大学叶轮机械研究所, 710049, 西安)

[摘要] 进化算法作为一类新的优化搜索方法, 广泛应用于各种优化问题. 现对简单遗传算法进行了改进, 采用实值编码, 并与模拟退火算法及基于适值排序和随机选择的方法相结合, 形成了改进遗传算法. 同时还介绍了一种新的进化算法—粒子群优化算法. 将这两种优化算法应用于函数优化, 并对优化结果进行了对比分析. 比较结果表明, 改进遗传算法和粒子群优化算法都可以在函数优化方面表现出较好的健壮性, 但在找寻最优解的效率上, 粒子群优化算法较好.

[关键词] 函数优化, 改进遗传算法, 粒子群优化算法

[中图分类号] O224; [文献标识码] A; [文章编号] 1672-1292(2002)02-0014-07

0 引言

自然界经过数万年的演化, 存在的方式、规律都是最优的. 所以, 人们总是在找寻自然界的优化规律, 发展形成所谓的进化算法, 并应用到各类优化问题, 如本文介绍的遗传算法以及近年兴起的粒子群优化算法就是借鉴自然界优化规律而产生的进化算法.

遗传算法(Genetic Algorithm, 简称GA)是20世纪70年代由美国密歇根大学J. H. Holland教授提出的^[1], 它是一类基于自然选择和遗传学原理的有效寻优方法, 目前在很多方面都有成功应用^[1~4]. 粒子群优化算法最初由Jim Kennedy于1995年提出并成功地用于连续非线性函数的优化^[5], 它是对鸟群觅食过程中的迁徙和聚集的模拟, 也可以说是对社会心理学的一种模拟. 粒子群优化算法对优化问题求解来说, 最大特点就是收敛速度较快.

工程优化应用的大多数问题, 都需要先对其进行数学建模, 即向数学问题转化, 然后对函数进行优化. 工程优化所建立的数学函数, 往往是带有多种约束条件的复杂函数, 而且大都是既不连续、也不可微的隐函数. 对于这些复杂函数来说, 用常规的数学方法很难或根本无法处理. 本文所涉及的这两种进化算法都能处理这类复杂函数的优化问题, 但由于算法的原理不同, 在优化过程中所表现出来的特性和优点也会不同. 为了了解两种算法的基本特点, 本文在对简单遗传算法改进的基础上, 通过一个较为典型的数学问题的优化, 对这两种进化算法的性态进行了对比分析, 以此来检验它们在工程优化中的有效性.

1 简单遗传算法及其改进研究

遗传算法通过对参数的编码进行操作, 而不是对参数本身进行操作, 这使得它处理优化问题时, 既不要求函数连续, 更不要求可微. 因此, 这些函数既可以是数学解析式所表达的显函数, 也可以是映射矩阵甚至是神经网络等隐函数, 因而应用范围较广. 进一步, 该方法通过目标函数来计算适配值, 不需要其

* 收稿日期: 2002-09-11.

基金项目: 教育部高校骨干教师资助计划资助(GG-807-10698-1016).

作者简介: 任斌, 1974-, 西安交通大学叶轮机械研究所博士研究生, 从事叶轮机械气动热力学优化研究.

它的推导和辅助信息, 从而对问题的依赖较小. 遗传算法是从许多初始点开始并行操作, 而不是从一个点开始, 因而可以有效的防止搜索过程收敛于局部最优, 而且有较大的可能求得全部的最优解. 遗传算法具有并行计算的特点, 因而可通过大规模并行计算来提高计算速度. 上述特点使得遗传算法较适合复杂问题的优化.

1.1 简单遗传算法

简单遗传算法一般由编码机制、适应值函数、遗传算子、控制参数等四个部分组成^[1]. 简单遗传算法由 0 和 1 进行编码, 0 和 1 称为基因, 由基因组成二进制串, 二进制串称为染色体(或称之为个体), 染色体与优化问题的解相对应, 并通过编码和解码过程实现基因空间和解空间的转换. 遗传算法中的优胜劣汰的评价标准是通过适应值函数来体现的, 通过适应值函数的好坏可以判断解的好坏, 从而判断染色体的优劣. 对于优化问题, 适应值函数是目标函数. 遗传算法中的遗传算子(genetic operator)主要有三种: 交叉算子(crossover)、变异算子(mutation)、选择算子(selection). 交叉和变异算子就是在算法中采用何种交叉和变异策略, 即采用单点或是多点交叉或变异. 选择算子也称复制算子(reproduction), 它的作用在于根据个体的优劣程度决定它在下一代是被复制还是被淘汰. 一般来说, 通过选择, 适应值较高的个体有较大的复制机会; 而适应值较小的个体即低劣的个体在下一代中存在的几率也就较小. 简单遗传算法中的控制参数包括, 染色体的串长(string)、种群大小(pop_size)、交叉概率(p_c)、变异概率(p_m)、最大遗传代数(maxgen)等.

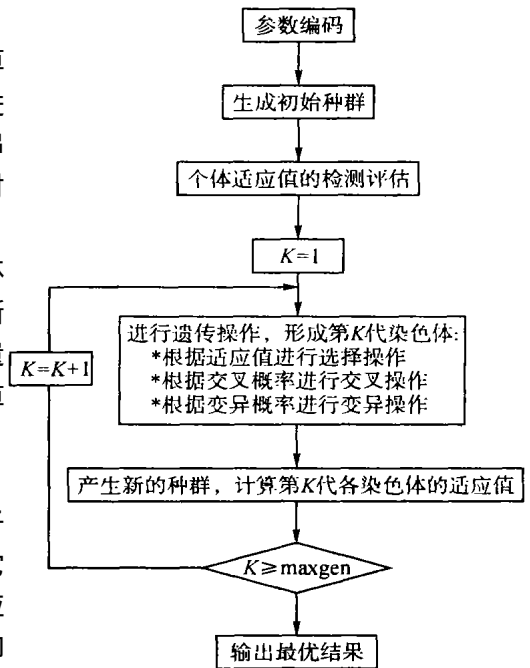


图1 简单遗传算法(SGA)计算流程图

1.2 改进遗传算法

本文在简单遗传算法的基础上, 通过采用实数编码机制, 并在遗传算子的改进等策略方面对简单遗传算法进行了改进.

采用实数编码机制, 即直接在优化问题的解空间进行编码, 不但省去了基因空间与解空间之间的编码和解码过程, 而且可以提高解的精度及减少优化过程的计算量. 相应地, 本文优化的二元函数采用如下的交叉和变异算子.

$$\begin{aligned}\alpha^* &= \alpha + (\beta - \alpha) * \omega; \\ \beta^* &= \beta + (\alpha - \beta) * (1 - \omega)\end{aligned}$$

$$\begin{aligned}\alpha^* &= \alpha * (1 - \omega * (1 - \text{gen}/\text{maxgen}) * * 2); \\ \beta^* &= \beta * (1 - \omega * (1 - \text{gen}/\text{maxgen}) * * 2).\end{aligned}$$

以上算子中, ω 为[0, 1]之间的随机数, gen、maxgen 分别为进化的当前代数和最大代数. 以此实现从上代染色体(α, β)到下代染色体(α^*, β^*)的进化.

在对选择算子的改进方面, 本文采用如下的选择方法产生下代种群.

- ①将杂交和变异产生的新染色体和父代所有染色体组合成新的被选种群;
- ②对被选种群按升序排列;
- ③删去所有重复的染色体;
- ④在被选种群中按顺序挑选两个染色体, 计算其概率 $\min\{1, \exp(-\Delta f/T_k)\}$ 当概率大于 random [0, 1] 时, 接受前一染色体进入新种群, 否则, 接受后一染色体进入新种群, 按照这种选择, 直到选择 pop

_size 个染色体组成新的种群.

改进后的遗传算法, 由于采用了实数编码, 避免了编码和解码的过程, 有利于提高计算效率. 又由于采用了合并种群的排序、剔除重复染色体以及在这个基础上的模拟退火原理的概率接受机制选择下代染色体^[6], 这样, 就可以以一定的概率接受较差解, 即适应值较差的染色体. 遗传算法在原理上是一种概率搜索方法, 所以, 只要所给的种群足够大, 遗传代数足够多, 原则上它是可以找到最优值的, 但这必须以牺牲时间为代价; 而且, 遗传算法作为一种优化算法, 虽然其健壮性较好, 但遇到多峰函数的时候, 有时候也难免收敛到局部最优值, 而不是全局最优值. 存在这两种问题的原因, 往往是在选择的过程中, 发生了有价值的染色体信息的丢失. 所以, 以一定的概率接受较差解, 就保证了种群的多样性, 也保存了有价值的染色体信息, 从而可以使搜索最优解的时间缩短且可以防止遗传算法收敛到局部最优解, 同时也增强了其健壮性.

2 粒子群优化算法

粒子群优化算法是对鸟群觅食过程中的迁徙和群集的模拟^[5, 7, 8]. 鸟群在觅食时, 在它们找到有食物的地方之前的迁徙过程中, 既有分散又有群集的特点. 所谓鸟语花香, 鸟也有鸟的语言, 它们总是通过自己一套独有的方式在无时无刻地相互传递着信息, 其实, 这种个体之间信息的交换在群居的生物群体中都存在, 如蜜蜂、蚂蚁, 也包括人类. 对于鸟群来说, 在它们找到食源之前的从一地到另一地的迁徙过程中, 总是有那么一只鸟对食物的嗅觉较好, 即对食源的大致方面具有较好的洞察力, 从而, 这只鸟就拥有食源的较好信息. 由于在找寻食源的途中, 它们又在无时无刻地相互传递信息, 特别是这种较好的信息. 所以, 在这种“好消息”的指引下, 最终导致了鸟群“一窝蜂”的奔向食源, 达到了在食源的群集. Jim Kennedy 从这种自然现象中受到启发, 提出了粒子群优化算法. 解群相当于一个鸟群, 一地到另一地的迁徙相当于解群的进化, “好消息”相当于解群每代进化中的最优解, 食源相当于全局最优解.

粒子群优化算法的数学抽象和实现步骤如下:

设定有 N 个解的解群 $X_i[d]$, $i \in [1, N]$, $d \in [1, M]$, d 为每个解的维数, 即每个解可以是个解向量. 然后, 对每个解定义一个速度矢量 V_i , 该速度矢量相当于 ΔX_i , 该速度矢量在第一代中与解群一样, 是随机初始生成的. 对于进化的第 k 代来说, 对于每一个解向量来说, 通过如下关系式生成下代解向量.

$$X_i(k+1) = X_i(k) + V_i(k). \quad (1)$$

其次, 定义 pbest (personal best) 和 gbest (global best) 分别代表个体 $k-1$ 代和 k 代的较好值和 k 代解群中的最好值, 即 $X_{\text{pbest}, i}$ 和 $X_{\text{gbest}, i}$. 而 $X_{\text{pbest}, i} - X_i$ 和 $X_{\text{gbest}, i} - X_i$ 分别表示每代个体解向量位置和个体最优位置及每代中最优个体位置的距离. 这种距离在速度矢量中体现. 如下式:

$$V_i(k+1) = V_i(k) + \eta(X_{\text{pbest}, i} - X_i) \quad (2)$$

$$V_i(k+1) = V_i(k) + \eta(X_{\text{gbest}, i} - X_i) \quad (3)$$

综合(2)式和(3)式可得到速度矢量的迭代变化:

$$V_i(k+1) = \omega V_i(k) + C_1 \times \eta(X_{\text{pbest}, i} - X_i) + C_2 \times \eta(X_{\text{gbest}, i} - X_i) \quad (4)$$

这样, 新解可以这样产生:

$$X_i(k+1) = X_i(k) + V_i(k+1). \quad (5)$$

以上各表达式中: k 为进行代数; C_1 , C_2 分别为自定义常数; ω , η 分别为 $[0, 1]$ 之间的随机数.

以本文函数优化为例, 粒子群优化算法可以通过流程图来实现(见图2).

粒子群优化算法具有和遗传算法某些相似的特征, 所以 Jim Kennedy 也称之为“一类进化算法”. 该算法的最大特点就是对函数优化显示了较高的效率, 据本文作者的理解, 是因为它借鉴了梯度优化方面的一些理念.

3 两种进化算法的对比分析

3.1 数学问题

本文中为了对改进遗传算法和粒子群优化算法进行对比分析, 采用了以下数学问题^[9].

$$\begin{aligned} \min f(x, y) &= (x-10)^3 + (y-20)^3 \\ \text{s. t. } (x-5)^2 + (y-5)^2 - 100 &\geq 0 \\ 0 - (x-6)^2 - (y-5)^2 + 82.81 &\geq 0 \\ 13 \leq x \leq 100, \quad 0 \leq y \leq 100 \end{aligned}$$

该问题为一带边界约束和非线性约束的二元函数的优化问题. 已知全局最优解为 $(x, y) = (14.095, 0.84296)$, $F(x, y) = -6961.813381$.

3.2 两种进化算法的优化对比

在两种进化优化方法对上述函数结果的对比作图中, 为了清楚地看出解群随进化代数的变化情况, 作者有意使其图形的坐标刻度随着进化代数的增加, 逐渐减小, 类似于一个解群变化的放大过程. 图中标号(gen)表示进化代数.(见图3, 图4).

3.2.1 对改进遗传算法的考查

对于改进遗传算法而言, 作如下的控制参数设定: 群体规模大小(pop_size)为20; 个体串长度(string)为2; 杂交概率(p_c)为0.98, 变异概率(p_m)为0.01; 最大遗传代数(maxgen)为500; 初始温度(T_0)为100.

3.2.2 对粒子群优化算法的考查

对于粒子群优化算法而言, 作如下的参数设定: 粒子群的最大进化代数为20; 粒子群的种群大小为20, 维数为2; 粒子群算法速度矢量中的参数为: $C_1: 3.0, C_2: 3.0, \omega: 3.0$.

3.2.3 对比分析

对于这两种进化算法对上述数学问题的优化, 采用的计算机配置为: CPU: 赛扬 1G; 内存: 128 Mb; 在这种配置下, 两种算法的对比如表1所示.

表1 两种进化算法的性能对比

项目	进化代数	耗费CPU时间	最优解	最优解出现代数
改进遗传算法	500	48 h	$(x, y) = (14.095010, 0.8429802)$ $F(x, y) = -6961.792000$	488
粒子群优化算法	20	10 s	$(x, y) = (14.095000, 0.8429654)$ $F(x, y) = -6961.809000$	17

通过数学算例的考查, 可以看出, 改进遗传算法和粒子群优化算法都能在有限的进化代数内找到其最优解. 从这两种进化算法的解群随进化代数的变化情况可以看出, 粒子群优化算法在较少的进化代数内, 其解群就向最优解的方向收敛, 这说明粒子群优化算法的优化效率较高. 改进遗传算法花费了较长的计算时间, 进行了较多的进化迭代, 最终找到了最优解. 其解群是逐渐收敛到最优解的, 收敛速度较慢. 这也正是遗传算法这种随机搜索算法的“特点”所在. 从数学算例的考查来看, 粒子群优化算法和改进遗传算法都表现了较好的健壮性.

从两种进化算法的原理看来, 粒子群优化算法优化初期, 其解群随进化代数而言表现了更强的随机性, 正是由于其产生下一代解群的较大的随机性, 以及每代所有解的“信息”的共享性和各个解的“自我素质”的提高, 如解群随进化代数的变化过程中存在“掉队”个体的“归队”现象, 这样, 就使得每代种群中

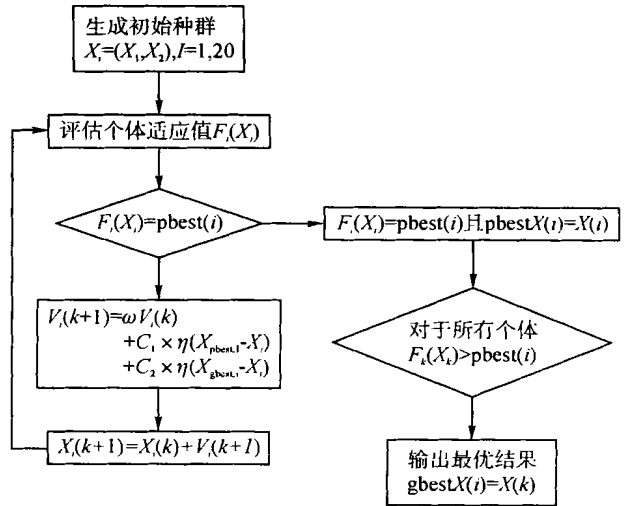


图2 粒子群优化算法流程图

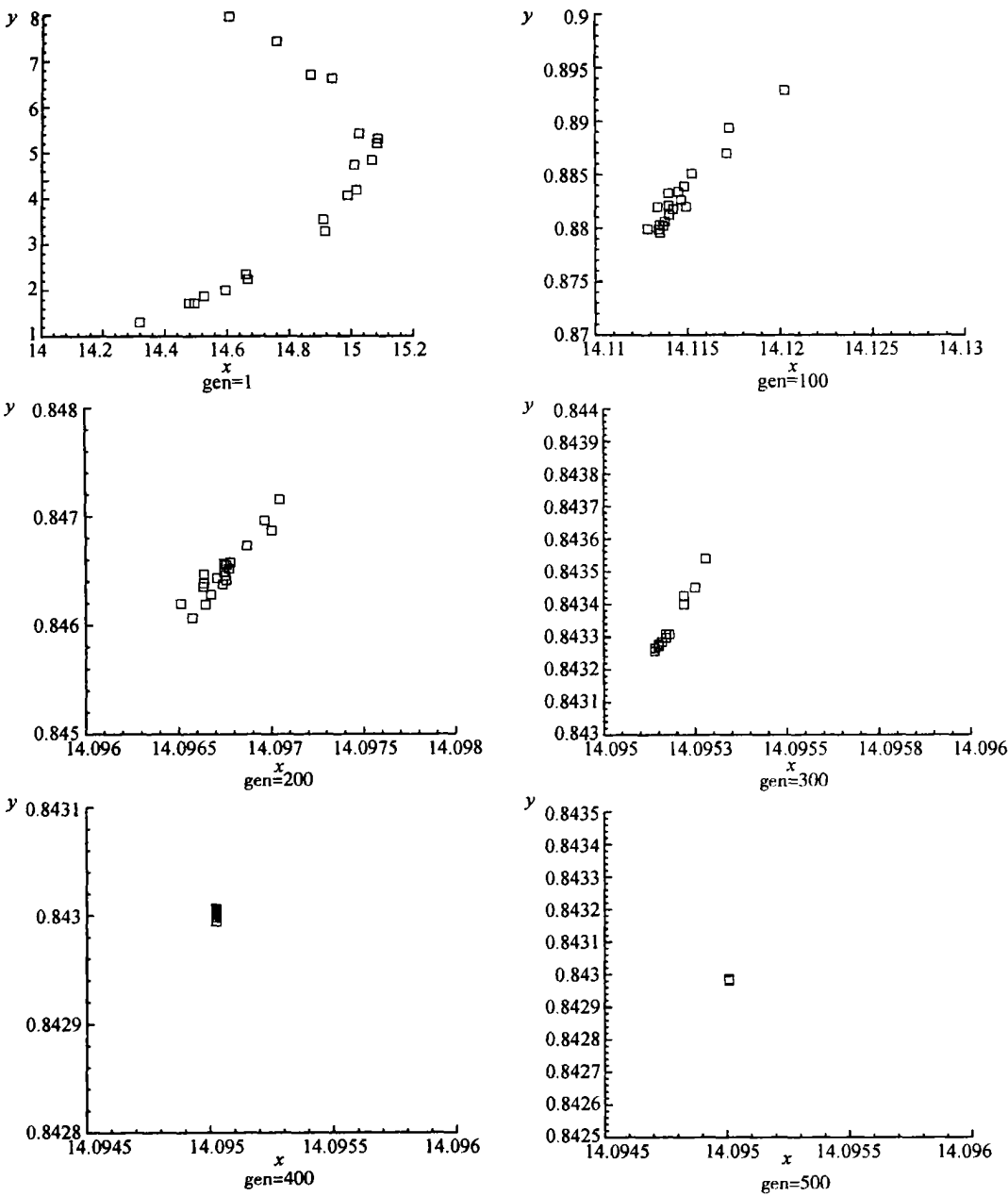


图 3 改进遗传算法的解随遗传代数的变化情况

的解具有“自我”学习提高和向“他人”学习的双重优点, 这样, 使得其下代解有针对性的从“先辈”那里继承更多的信息. 在这种“自我提高”和“取人之长, 补己所短”的前提下, 很快就达到了群体最优, 从而能在较少的代数内找到最优解.

4 结论

本文提出了一种改进的遗传算法, 介绍了一种新的进化算法——粒子群优化算法, 并用相同的二元函数数学问题对其进行了对比分析. 结果表明, 改进遗传算法和粒子群优化算法在带约束条件数学函数的优化方面都显示了较好的优化特性. 改进遗传算法和粒子群优化算法在函数优化方面显示的特征不同, 前者耗时较长, 在算法的改进方面还有待于做进一步工作, 以提高其优化效率; 后者的优化效率较高, 但

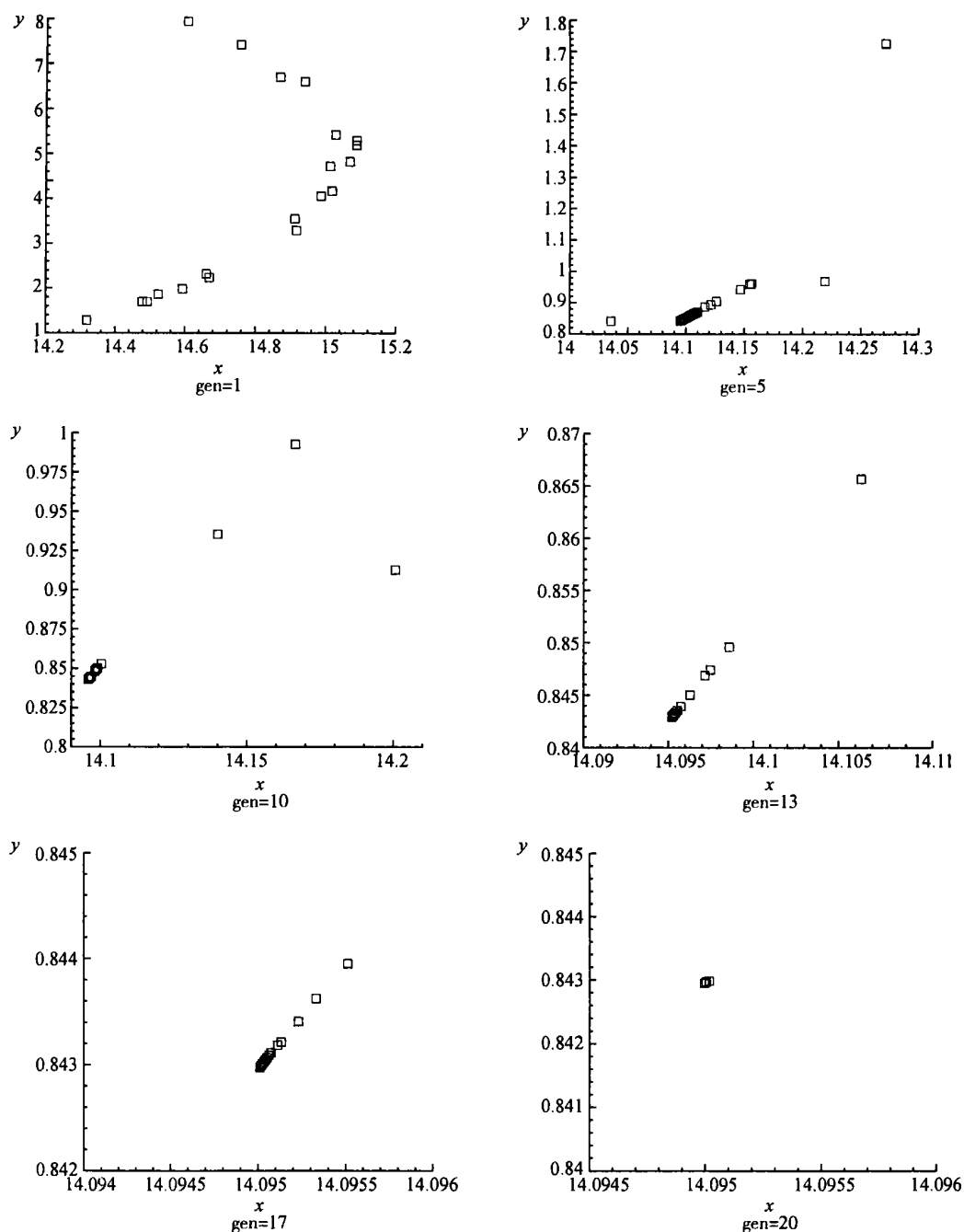


图4 粒子群优化算法的解随进化代数的变化情况

还未见在实际的优化问题中广泛应用, 所以还有待于与不同的实际优化问题相结合, 以此验证其在实际优化应用中的广泛性和健壮性.

[参考文献]

- [1] 任平. 遗传算法(综述)[J]. 工程数学学报, 1999, 16(1): 1~ 8.
- [2] 丰镇平, 李军, 沈祖达. 遗传算法及其在透平机械设计中的应用[J]. 燃气轮机技术, 1997, 11(2): 13~ 22.
- [3] Li Jun, Feng Zhenping, *et al.* Aerodynamic Optimum Design of Transonic Turbine Cascades Using Genetic Algorithms[J].

Journal of Thermal Science, 1997, 6(2): 364~ 368.

- [4] 董彤, 丰镇平, 李军. 遗传算法在透平叶栅多目标优化设计中的应用[J]. 中国电机工程学报, 1999, 19(6): 74~ 76.
- [5] V. Tandon. NC End Milling Optimization Using Evolutionary Computation[J]. International Journal of Machine Tools and Manufacture, 2001, 42: 595~ 605.
- [6] 王雪梅, 王义和. 模拟退火算法和遗传算法的结合[J]. 计算机学报, 1997, 20(4): 381~ 384.
- [7] F Zhang, D Xue. Optimal Concurrent Design Based upon Distributed Product Development Life-cycle Modeling[J]. Robotics and Computer Integrated Manufacturing, 2001, 17: 469~ 486.
- [8] A R Cockshott, B E Hartman. Improving the Fermentation Medium for Echinocandin B Production Part II: Particle Swarm Optimization[J]. Process Biochemistry, 2001, 36: 661~ 669.
- [9] [美] Z. 米凯利维茨, 著. 演化程序——遗传算法和数据编码的结合[M]. 周家驹, 等译. 北京: 科学出版社, 2000, 103.

Improved Genetic Algorithm and Particle Swarm Optimization as well as Comparison between Them

Ren Bin, Feng Zhenping

(Institute of Turbomachinery, Xi'an Jiaotong University, 710049, Xi'an, PRC)

Abstract: As a new kind of optimization search techniques, the evolutionary algorithms are widely used to solve different problems in optimal areas. After a careful research, the simple genetic algorithm has been improved by adopting float-coding method, simulated annealing algorithm and sorted stochastic fitness selection strategy; and has been applied to mathematic function optimization. In addition, a new evolutionary algorithm—Particle Swarm Optimization is introduced and applied to the same mathematic function optimization. The optimization results are compared with each other in this paper. The comparative result indicates that the Improved Genetic Algorithm and Particle Swarm Optimization are both robust. But Particle Swarm Optimization can obtain the optimum solutions more easily than the Improved Genetic Algorithm, and it is a good optimization method with strong competitiveness.

Key words: function optimization, improved genetic algorithm, particle swarm optimization

[责任编辑: 刘健]