第 4期
2006年 4月
电 子 学 报
ACTA ELECTRONICA SINICA
Vol.34 No.4
Apr. 2006

基于违约解转化法的遗传算法及其性能分析

高玉根[1],程 峰[1],王 灿[1],王国彪[2]
(1.浙江科技学院机械与汽车工程学院,浙江杭州 310038;2.北京科技大学土木与环境工程学院,北京 100083)

摘 要: 遗传算法在求解约束优化问题时,面临的关键问题之一就是如何处理约束条件.本文提出了一种基于违约解转化法的遗传算法(CIFGA),也就是遗传算法在处理约束条件时,在每一进化代遗传操作后,把所有违反约束条件的个体逐个转化成满足约束条件的个体,整个遗传群体保持不变,经过一代代的进化,最终求出约束问题的最优解.对于采用二进制编码和实数编码的CIFGA,理论证明了其收敛性.测试试验结果表明:CIFGA有较好的算法性能和解决约束优化问题的能力.

关键词: 遗传算法;收敛性;约束优化
中图分类号: TP18 文献标识码: A 文章编号: 0372-2112(2006)04-0638-04

A New Improved Genetic Algorithms Based on Converting Infeasible Individuals into Feasible Ones and Its Property Analysis

GAO Yu-gen[1], CHENG Feng[1], WANG Can[1], WANG Guo-biao[2]
(1. Zhejiang University of Science and Technology, Hangzhou, Zhejiang 310023, China;
2. University of Science and Technology Beijing, Beijing 10083, China)

Abstract As Genetic Algorithms handles the constraint optimization problem, the difficulty is how to solve the constraints. According to this problem, a new improved Genetic Algorithms (CIFGA) is proposed. The key strategy in CIFGA is that the whole infeasible individuals, appeared after each generation, will be transformed into feasible ones. Go through generation after generation, the optimum solution of optimization problem can be founded. The CIFGA, with either binary coding or real coding is proved to converge to global optimum solution. The experimental results show that CIFGA has great advantage of convergence property over the GAs based on Penalty Function (PFGA), and has good ability of solving constrained optimization in general purpose.

Key words genetic algorithms; convergence property; constrained optimization

1 引言

遗传算法(Genetic Algorithms)求解无约束优化问题已取得了较好的效果,但求解约束优化问题还有一定的困难.在工程优化问题中,大多数带有约束条件,遗传算法在求解这类问题的关键之一是如何处理约束条件[1,2].目前遗传算法已有一些处理约束条件的方法,如惩罚函数法、可行解搜索法、解码器法、算子修正法、IFDNAGA等,其中惩罚函数法最为常用[3].这些算法在处理约束条件时各有不同的方法,但在应用时都受到一定的限制.本文提出了一种较为通用的处理约束条件的方法——违约解转化法,并把此方法应用到遗传算法中,就形成了基于违约解转化法的遗传算法(Genetic Algorithms based on Converting Infeasible individuals into Feasible ones,简称CIFGA).

2 基于违约解转化法的遗传算法(CIFGA)

2.1 基本思想

其基本思想是:在遗传算法的每一进化代中,经过遗传操作之后,将产生许多个体后代,有的在可行域内,有的在可行域外.然后,在可行域内找出一个或几个适应值较好的个体,每个可行域外的个体都被拉向好的个体,使它逐渐进入可行域内,转变成可行个体.因此,经过这样的处理后,本次进化代的所有个体都在可行域内,至此,这一代进化完毕.经过一代代的进化,每一代都采用违约解转化法把违反约束的个体进行转化成可行域内的个体,直到进化终止,最后求出问题的最优解.

收稿日期: 2005-05-20;修回日期: 2005-09-26
基金项目:国家自然科学基金(No.95705010)

现以一个二维问题进行说明. 如图 1所示, 若 Ω 表示遗传空间, Φ 表示可行域, $\overline{\Phi}$ 表示不可行域, 则 $\Phi\subset\Omega$, $\overline{\Phi}\subset\Omega$. 设个体 $\boldsymbol{S}\in\Phi$, 而且是可行域中适应值较好的; 个体 $\boldsymbol{R}\in\overline{\Phi}$. 连接 $\boldsymbol{R}$ 和 $\boldsymbol{S}$ 成直线, 在直线上计算一点 $\boldsymbol{R}_1$, 且满足公式:

$$\boldsymbol{R}_1=\boldsymbol{S}-\alpha(\boldsymbol{S}-\boldsymbol{R}) \tag{1}$$

式中转化系数 α 取值为: $0.6<\alpha<0.8$. 若 $\boldsymbol{R}_1$ 不在可行域, 则将 $\boldsymbol{R}_1\Rightarrow\boldsymbol{R}$, 按公式(1)重新计算 $\boldsymbol{R}_1$; 如此进行下去, 直至 $\boldsymbol{R}_1$ 成为可行个体为止. 然后其余不可行域的个体都按照此方法拉向个体 $\boldsymbol{S}$, 使其成为可行个体. 经过这样的转化处理后, 所有的不可行个体都变成可行的. 最后, 所有变成可行的个体替换所有原来不可行的个体, 遗传算法的群体规模保持不变.

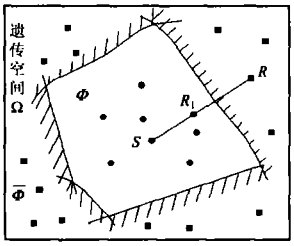

图 1　违约解转化法的示意图

违约解转化法的实现步骤和 CIFGA 的算法流程图见文献[4].

2.2　个体 $\boldsymbol{S}$ 和转化系数 α 的选取

个体 $\boldsymbol{S}$ 的选取对遗传算法的性能稍有影响. 若 $\boldsymbol{S}$ 的适应值选择最大, 生成的 $\boldsymbol{R}_1$ 群体的整体适应值偏高, 有可能会引起遗传算法的早熟现象; 若选择最小, 生成的群体的整体适应值偏低, 可能会使得遗传算法收敛速度降低, 影响运行效率. 因此, 建议个体 $\boldsymbol{S}$ 选择可行域中适应值较好的一个个体.

转化系数 α 影响着不可行个体转化成可行个体的转化效率, 若选择的过小, 转化的效率高, 但生成的 $\boldsymbol{R}_1$ 群体的基因模式相似, 这会降低遗传算法的局部搜索能力, 即使已经搜索到最优解附近, 但很难达到最优解; 若选择的过大, 转化的效率低, 不可行个体进入可行域的时间长, 多少会影响遗传算法的运行效率. α 一般选在 0.6~0.8.

总之, 个体 $\boldsymbol{S}$ 和转化系数 α 的选取不会影响遗传算法的收敛性. 在运行效率方面, 虽然不起主要作用, 但也会稍影响遗传算法的收敛速度.

2.3　收敛性

在标准的遗传算法(简称 SGA)中, 对个体有三种遗传操作: 选择、交叉和变异; 此外, 还需要对所求解问题的解空间状态变量进行编码, 解输出时还要进行解码; 另外, 还要建立适应度函数, 用它对 SGA 中的个体进行评价. 因此 SGA 可概括成如下的五元体:

$$\mathrm{SGA}=\{S_e, C_r, M_u, D_e, f_v\} \tag{2}$$

集合 SGA 的元是: S_e—选择操作; C_r—交叉操作; M_u—变异操作; D_e—编码、解码; f_v—适应度评价.

在 SGA 应用过程中, 人们往往结合问题的特征和领域知识对 SGA 进行各种改变, 形成了各种各样的具体的 GAs. 其中, 最常见的是在 SGA 的基础上加入杰出个体保留策略. 这样改进的 GAs 可表示成如下的六元体:

$$\mathrm{GAs}=\{S_e, C_r, M_u, D_e, f_v, E_l\} \tag{3}$$

其中, E_l—杰出个体保留策略.

CIFGA 也属于改进的 GAs, 它是在杰出个体保留策略的基础上又增加了一项对不满足约束的个体(infeasible individual)的转化处理, 因此, CIFGA 可表示成如下的七元体:

$$\mathrm{CIFGA}=\{S_e, C_r, M_u, D_e, f_v, E_l, M_i\} \tag{4}$$

式中: M_i—违约解转化操作. 在对违约解转化操作过程中使用了式(1). 式(1)进一步变换为如下形式:

$$\boldsymbol{R}_1=(1-\alpha)\boldsymbol{S}+\alpha\boldsymbol{R} \tag{1}$$

如果把公式(5)中的个体 $\boldsymbol{S}$ 和 $\boldsymbol{R}$ 看作两个父代, 那么等号(=)右边部分 $\boldsymbol{S}$ 和 $\boldsymbol{R}$ 的线性组合则可以表示两个父代进行一次算术交叉运算, $\boldsymbol{R}_1$ 则可看作是他们交叉后产生的一个子代, 假设两个父代个体为: $\boldsymbol{S}=[x'_1, x'_2, \cdots, x'_m]^T$; $\boldsymbol{R}=[x''_1, x''_2, \cdots, x''_m]^T$, 按照式(5)进行运算, 得到 R_1 为:

$$\boldsymbol{R}_1=(1-\alpha)\begin{bmatrix}x'_1\\x'_2\\\vdots\\x'_m\end{bmatrix}+\alpha\begin{bmatrix}x''_1\\x''_2\\\vdots\\x''_m\end{bmatrix}=\begin{bmatrix}(1-\alpha)x'_1+\alpha x''_1\\(1-\alpha)x'_2+\alpha x''_2\\\vdots\\(1-\alpha)x'_m+\alpha x''_m\end{bmatrix}=\begin{bmatrix}r_{11}\\r_{12}\\\vdots\\r_{1m}\end{bmatrix} \tag{6}$$

可以看出, 违约解转化法对不满足约束个体的处理相当于进行了一次交叉操作. 所以, CIFGA 集合中的元 C_r 和 M_i 可以合并成 C'_r, 即:

$$C_r\cup M_i=C'_r \tag{7}$$

于是, CIFGA 进一步变成:

$$\mathrm{CIFGA}=\{S_e, C'_r, M_u, D_e, f_v, E_l\} \tag{8}$$

由式(8)和式(3)看出, CIFGA 和改进的 GAs 一样也是一个六元体的集合. 若 CIFGA 采用二进制编码, 文献[5]已经证明: 采用式(3)操作元的遗传算法是收敛的; 若 CIFGA 采用实数编码, 文献[6]也证明了具有式(3)操作元的实数编码的遗传算法(RGA)是收敛的. 也就是说, 不论是二进制编码, 还是实数编码, CIFGA 同样可以收敛到全局的最优解.

3　CIFGA 的试验结果及其评价

为了评价 CIFGA, 本文采用了 3个典型的具有约束的测试实例[7], 对 CIFGA 进行测试. 这 3个测试实例为:

$$(1)\ \max F_1(\boldsymbol{X})=4x_1+3x_2 \tag{9}$$

$$\text{s.t.}\quad 6-2x_1-3x_2\geqslant 0 \tag{10}$$

$$3+3x_1-2x_2\geqslant 0 \tag{11}$$

$$4-2x_1-x_2\geqslant 0 \tag{12}$$

$$0\leqslant x_i\leqslant 2,\ i=1, 2 \tag{13}$$

此实例一个线性规划优化问题. 已知全局的最优解为: $\boldsymbol{X}^*=[1.5, 1.0]^T$, $F(\boldsymbol{X}^*)=9.0$, 并且最优解位于搜索空间的边界上.

$$(2)\ \min F_2(\boldsymbol{X}) = -x_1 - x_2 \tag{14}$$

$$\text{s.t. } 2x_1^4 - 8x_1^3 + 8x_1^2 - x_2 + 2 \geqslant 0 \tag{15}$$

$$4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 - x_2 + 36 \geqslant 0 \tag{16}$$

$$0 \leqslant x_1 \leqslant 3 \tag{17}$$

$$0 \leqslant x_2 \leqslant 4 \tag{18}$$

从数学模型可知, 约束函数全部为非线性约束不等式. 已知全局的最优解为 $\boldsymbol{X}^* = [2.3295, 3.1783]^T$, $F(\boldsymbol{X}^*) = -5.5079$.

$$(3)\ \min F_3(\boldsymbol{X}) = (x_1 - 10)^3 + (x_2 - 20)^3 \tag{19}$$

$$\text{s.t. } (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geqslant 0 \tag{20}$$

$$-(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \geqslant 0 \tag{21}$$

$$13 \leqslant x_1 \leqslant 100 \tag{22}$$

$$0 \leqslant x_2 \leqslant 100 \tag{23}$$

此实例的所有约束是非线性约束, 可行空间比较狭窄, 约束条件较强, 获得全局的最优点较困难. 已知问题的全局解为

$$\boldsymbol{X}^* = (14.095, 0.84296)^T, F(\boldsymbol{X}^*) = -6961.8138$$

为了评价 CIFGA, 将 CIFGA 与惩罚函数法的遗传算法 (本文简称 PFGA) 一起对上述 3 个测试函数进行测试实验. 在计算时, 两种算法都采用相同的遗传参数和运行参数, 即采用实数编码、杰出个体保护, 群体规模 $M = 50$, 进化代数 $N = 100$, $P_c = 0.55$, $P_m = 0.1$.

3.1 CIFGA 的收敛频率和收敛速度

(1) 收敛频率

用不同算法进行独立运行 100 次, 设全局最优点所在的容差区间为 $[F^* - 0.005F^* + 0.0005]$. 收敛频率见表 1.

表 1　两种算法的收敛频率比较 (单位: 次)

函数 / 算法	F_1	F_2	F_3
CIFGA	100	100	100
PFGA	100	67	26

从表中可以看出, 对于 $F_1 \sim F_3$, CIFGA 运行 100 次全部收敛, 且收敛频率为 100%. 除 F_1 外, PFGA 收敛频率没有达到 100%, 而且对于可行域空间较窄的 F_3, 收敛率只有 26%. 可见 CIFGA 的收敛频率远优于 PFGA.

(2) 收敛速度

在同一台计算机上, 相同的运行环境, 相同的容差区间条件下 (同上), 两种算法独立执行 100 次, 得到函数 $F_1 \sim F_3$ 的收敛速度、最优解见表 2. 从表中可以看出, CIFGA 的收敛速度远远高于 PFGA. 并且对于可行域较窄的函数 F_3, CIFGA 的收敛速度更加明显.

表 2　两种算法的收敛速度比较 (单位: 秒)

函数	最优函数值	CIFGA		PFGA	
		平均收敛时间	收敛时的最优解	平均收敛时间	收敛时的最优解
F_1	9	5.7	8.9998	9.4	8.9997
F_2	-5.5079	6.6	-5.5078	13.4	-5.5084
F_3	-6961.8138	23.5	-6961.8137	#	#

注: 表中#表示进化 500 代时该算法仍无法收敛于最优点

3.2 在线性能和离线性能

选取线性规划问题函数 F_1 和非线性规划问题函数 F_2, 对 CIFGA 和 PFGA 进行在线性能、离线性能测试, 如图 2, 图 3 所示.

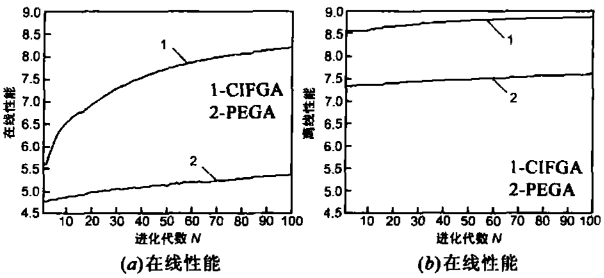

图 2　函数 F_1 与算法的性能关系

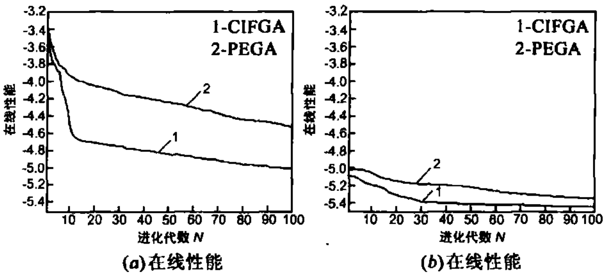

图 3　函数 F_3 与算法的性能关系

从图中可以看出, 在进化过程中, CIFGA 的动态性能和收敛性能都优于 PGA, 而且能够迅速地逼近最优值. 因此, CIFGA 的性能要高于 PGA.

4　结论

本文提出了一种基于违约解转化法的遗传算法 (CIFGA), 为遗传算法求解约束优化问题提供了一种新的、较为一般性的解法. CIFGA 简单、易行、易于实现. 文中给出了约束方程中只有不等式约束的几个测试函数, 从测试试验结果可以看出, CIFGA 且具有较好计算稳定性和收敛速度, 同时在线性能和离线性能也表明了 CIFGA 有较好的算法性能. 因此, 在求解只有不等式约束的优化问题方面, CIFGA 是可行的.

参考文献:

[1] Michalewicz Z, Dasgupta D, Le R, et al. Evolutionary algorithm for constrained engineering problems[J]. Computers & Industrial Engineering, 1996, 30(4): 851-870

[2] 王跃宣, 刘连臣, 等. 处理带约束的多目标优化进化算法[J]. 清华大学学报(自然科学版), 2005, 45(1): 103-106

[3] 潘正君, 康立山, 陈毓屏, 著. 演化计算[M]. 北京: 清华大学出版社, 1998. 74-84

[4] 高玉根, 王国彪, 丁预展. 一种改进的遗传算法及其在

约束优化中的应用 [J]. 淄博学院学报 (自然科学与工程版), 2002, 4(2): 17- 20.
[5] Rudolph G. Convergence analysis of canonical genetic algorithms[J]. IEEE Transactions on Neural Network, 1994, 5(1): 63- 66
[6] 林丹, 李敏强, 寇纪淞. 基于实数编码的遗传算法的收敛性研究 [J]. 计算机研究与发展, 2000, 37(11): 1321 - 1327.
[7] [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]. 北京: 科学出版社, 2000.

作者简介:

高玉根 男, 出生于山东省嘉祥县, 博士, 浙江科技学院副教授, 主要从事现代设计理论与方法的研究工作. E-mail: gaoyugen@ 163. com

程 峰 男, 出生于江苏沛县, 硕士, 浙江科技学院教授级高工, 主要从事机械设计理论与方法的研究.