

模糊自适应参数调整的改进遗传算法

罗 梅¹, 贾志娟²

(1 天津工程师范学院 计算机系, 天津 300222; 2 郑州师范学院 信息技术系, 河南 郑州 450044)

摘 要: 为提高遗传算法的优化性能, 构建了交叉及变异算子的模糊动态调整器, 给出了参数调整过程、模糊逻辑控制器的执行策略及控制过程. 采用标准的 Benchmark 测试函数比较了模糊控制器参数调整的遗传算法和简单遗传算法的性能, 结果表明该算法求解精度高, 优化效率高及进化代数少.

关键词: 遗传算法; 模糊逻辑控制器; 算子分析; 自适应参数调整

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2010)04-0001-04

The Improved Genetic Algorithm Based on Fuzzy Controller with Adaptive Parameter Adjustment

LUO Mei¹, JIA Zhi-juan²

(1 Dept. of Computer Science, Tianjin University of Technology and Education, Tianjin 300222, China;

2 Dept. of Information and Technology, Zhengzhou Teachers College, Zhengzhou 450044, China)

Abstract: For providing the optimization performance of the genetic algorithm, the fuzzy dynamic regulator for mutation and crossover operators is constructed, the adjusting process of parameter is specified, and the implementing tactics as well as the controlling process are also provided in detail. The performances of the improved genetic algorithm and simple genetic algorithm are compared by the standard Benchmark testing function, the result indicate that the proposed method has many advantages such as higher optimization accuracy, higher optimization efficiency and fewer evolution steps.

Key words: genetic algorithm; fuzzy logic controller; operator analysis; adaptive parameter adjustment

1 引言

遗传算法的局部搜索能力比较弱, 常常表现在容易陷入局部最优解^[1]. 为改变其局部搜索能力弱的缺点, 人们也提出了多种改进办法, 如从编码方法、适应度函数的定标, 从控制参数等方面加以改进, 但未能从算法参数变化上提高遗传算法的搜索能力. 基于此, 文中构建了交叉算子及变异算子的参数自适应动态调整控制器, 从而为采用遗传算法获得复杂多峰优化问题的全局优化解提供了很好的实现手段.

2 遗传算子的性能分析

标准遗传算法的交叉和变异算子的设计对遗传算法影响很大. 文中将从算子与群体多样性的关系、算子的搜索能力以及算子的适用范围三个方面分析各算子的性能.

2.1 交叉算子的性能分析

为探讨交叉算子与群体多样性之间的关系, 先给出一个群体多样性的计算方法.

假定任意优化问题的二进制编码空间为 $\{0, 1\}^L$, 群体规模设定为 N , 群体所包含个体集合 $P = \{a_1, a_2, \dots, a_N\}$, 其中, $a_j = (a_{j1}, a_{j2}, \dots, a_{jL})$, $j =$

1, 2, ..., N 其多样性测度为

$$m(p) = 1 - \frac{1}{l \times N} \sum_{i=1}^l (\max\{\sum_{j=1}^N (a_{ji}), \sum_{j=1}^N (1 - a_{ji})\} - \min\{\sum_{j=1}^N (a_{ji}), \sum_{j=1}^N (1 - a_{ji})\}) \quad (1)$$

由式(1)知, $m(p) \in [0, 1]$, 当 $m(p) = 0$ 时, 群体的多样性完全消失; 当 $m(p) = 1$ 时, 群体的多样性测度最大^[2].

为分析交叉算子与群体多样性的关系, 设初始群体中任意两个个体 $a_1, a_2 \in P, a_1 \neq a_2, a_1 = (a_{11}, a_{12}, \dots, a_{1l}), a_2 = (a_{21}, a_{22}, \dots, a_{2l})$; 假设采用单点交叉, 交叉点位置在 1 处, 经过交叉运算后, 两个新个体为: $a'_1 = (a_{11}, a_{22}, \dots, a_{2l}), a'_2 = (a_{21}, a_{12}, \dots, a_{1l})$, 形成的子代群体为 $P' = \{a'_1, a'_2, \dots, a'_N\}$, 其中当 $2 < j \leq N$ 时, $a'_j = a_j$. 根据式(1), 可以计算得

(1) 当 $i = 1$ 时,

$$\begin{aligned} \sum_{j=1}^N a'_{j1} &= a'_{11} + a'_{21} + \dots + a'_{N1} \\ &= a_{11} + a_{21} + \dots + a_{N1} = \sum_{j=1}^N a_{j1} \end{aligned}$$

$$\text{同理可得 } \sum_{j=1}^N (1 - a'_{j1}) = \sum_{j=1}^N (1 - a_{j1});$$

(2) 当 $i = 2$ 时,

$$\begin{aligned} \sum_{j=1}^N a'_{j2} &= a'_{12} + a'_{22} + \dots + a'_{N2} \\ &= a_{12} + a_{22} + \dots + a_{N2} = \sum_{j=1}^N a_{j2} \end{aligned}$$

$$\text{同理可得 } \sum_{j=1}^N (1 - a'_{j2}) = \sum_{j=1}^N (1 - a_{j2})$$

$$(3) \text{ 当 } 2 < i \leq l \text{ 时, } \sum_{j=1}^N a'_{ji} = \sum_{j=1}^N a_{ji}, \sum_{j=1}^N (1 - a'_{ji}) = \sum_{j=1}^N (1 - a_{ji}).$$

由上述计算知, 群体多样性测度 $m(p) = m(p')$, 即单点交叉的交叉运算对群体多样性没有影响, 可以证明, 任意多点交叉运算均不改变群体的多样性, 但是交叉算子能够增强模式的重组能力, 提高搜索效率.

为探讨交叉算子的搜索能力, 假设模式空间为 $\{0, 1, *\}^l$, 阶为 R 定义距为 L 的模式在采用单点交叉算子的情况下, 被单点交叉算子破坏的概率上限为 $p_d(R, L) = p_c \times L / (l - 1)$, 存活概率的下限为 $p_s(R, L) = 1 - p_c \times L / (l - 1)$. 设 r_1, r_2 为两低阶不相交的模式, 分布在不同的父模式中, 它们组成 r

$= r_1 + r_2$ 阶不包含父模式的新模式的概率为 p_r , 即 $p_r = 1 - p_s(r_1, p_c) \times p_s(r_2, p_c)$. 当交叉概率较大时, 个体的重构能力增强, 提高了算法开辟新的搜索空间的能力. 若交叉概率太低, 个体的重构率较低, 算法的搜索可能陷入迟钝状态. 因此, 交叉概率是影响算子搜索能力的关键因素.

2.2 变异算子的性能分析

对于变异算子与群体多样性的关系, 通过对式(1)计算知, 群体多样性测度 $m(p) \neq m(p')$, 即单点变异运算对群体多样性有影响, 其他变异方式也能改善群体的多样性.

对于变异算子的搜索能力, 因变异算子是用来保证群体多样性的算子, 同交叉算子类似, 变异算子同样对个体具有重构和破坏的双重作用. 阶为 R 的模式在单点变异操作下的存活概率与重构概率分别为: $p_s(R, p_m) = (1 - p_m)^R, p_r(R, p_m) = 1 - p_s(R, p_m) = 1 - (1 - p_m)^R$, 故在变异算子的作用下, 随着进化代数的增加, 任意模式都将被搜索到, 其搜索能力与变异概率以及初始种群无关.

综上所述: 在进化初期一般需要提高模式的重组能力, 加大空间搜索的广度, 以便找到最优解所在的子空间, 跳出局部极值点; 在进化的后期则需要提高模式的存活能力, 以便在最优解子空间上进行局部搜索. 因此往往希望自适应地调节交叉率及变异率从而充分利用交叉算子及变异算子的搜索能力. 实际应用中发现: 对于交叉运算是产生新个体的主要方法, 它决定了遗传算法的全局搜索能力, 因此在进化开始时交叉算子 P_c 采用大概率而进化后期 P_c 采用小概率, 这样易于达到全局收敛性能. 当变异概率 P_m 很小时, 群体的稳定性好, 一旦陷入局部极值就很难跳出来, 易产生未成熟收敛; 而增大 P_m 的值, 可能破坏解群体的同化, 使解空间保持多样性, 搜索过程可以从局部极值点跳出来, 收敛到全局最优解. 在求解过程中也可以使用可变的 P_m , 即算法早期 P_m 取值较大, 扩大搜索空间; 算法后期 P_m 取值较小, 加快收敛速度^[3].

3 模糊逻辑自适应的遗传算法参数调整方法

3.1 遗传算法的参数调整过程

由于在遗传进化过程中, 系统对进化的参数比较敏感, 而且在进化的过程中如采用固定的控制参数则不利于收敛到全局最优解, 为了动态调整交叉

概率 P_c 及变异概率 P_m , 系统采用了两个模糊控制器进行自适应的调整^[4]. 在进化过程中, 系统连续计算三代的平均适应度值即 $k - 2\Delta\overline{\text{eval}}(V; k - 2)$ 、 $k - \Delta\overline{\text{eval}}(V; k - 1)$ 和 $k - \Delta\overline{\text{eval}}(V; k)$. 且

$$\begin{aligned} & \Delta^2\overline{\text{eval}}(V; k - 1) \\ &= (\Delta\overline{\text{eval}}(V; k - 2) - \Delta\overline{\text{eval}}(V; k - 1)) \\ &= \left[\frac{\sum_{j=1}^N \text{eval}(v_j; k - 2)}{N} - \frac{\sum_{j=1}^N \overline{\text{eval}}(v_j; k - 1)}{N} \right] \times \\ & \quad \lambda \\ & \Delta^2\overline{\text{eval}}(V; k) \\ &= (\Delta\overline{\text{eval}}(V; k - 1) - \Delta\overline{\text{eval}}(V; k)) \\ &= \left[\frac{\sum_{j=1}^N \text{eval}(v_j; k - 1)}{N} - \frac{\sum_{j=1}^N \text{eval}(v_j; k)}{N} \right] \times \lambda \end{aligned}$$

式中, $V = [v_1 \cdots v_2 \cdots v_j]^T$, λ 为模糊化因子.

这样, P_c 动态调整过程即顺序判断下列规则:

if $\epsilon \leq \Delta^2\overline{\text{eval}}(V; k - 1) \leq \gamma_c$ and $\epsilon \leq \Delta^2\overline{\text{eval}}(V; k) \leq \gamma_c$

then 增大 P_c 并产生下一代;

if $-\gamma_c \leq \Delta^2\overline{\text{eval}}(V; k - 1) \leq -\epsilon$ and $-\gamma_c \leq \Delta^2\overline{\text{eval}}(V; k) \leq -\epsilon$

then 减小 P_c 并产生下一代;

if $-\epsilon \leq \Delta^2\overline{\text{eval}}(V; k - 1) \leq \epsilon$ and $-\epsilon \leq \Delta^2\overline{\text{eval}}(V; k) \leq \epsilon$

then 快速增大 P_c 并产生下一代.

上述规则中, ϵ 为一个事先给定的近零实数值; γ_c 为一个事先设置的模糊隶属函数最大值, $-\gamma_c$ 为模糊隶属函数的最小值.

P_m 的动态调整过程与 P_c 的调整过程类似, 只是将规则中 $\pm \gamma_c$ 变换为 $\pm \gamma_m$.

3.2 模糊逻辑控制器的执行策略

对于 P_c 的执行策略分为 (P_m 控制器的执行策略类同):

① 模糊控制器 P_c 的输入和输出: 该控制器的输入为 $\Delta^2\overline{\text{eval}}(V; k - 1)$ 和 $\Delta^2\overline{\text{eval}}(V; k)$; 输出为 P_c 的变化量 $\Delta P_c(k)$. ② 设置隶属函数 $\Delta^2\overline{\text{eval}}(V; k - 1)$ 、 $\Delta^2\overline{\text{eval}}(V; k)$ 及 $\Delta P_c(k)$, 其中模糊逻辑隶属函数的输入和输出语言变量如图 1、图 2 所示, 且 $\Delta^2\overline{\text{eval}}(V; k - 1)$ 及 $\Delta^2\overline{\text{eval}}(V; k)$ 被事先标准化到 $[-1.0, 1.0]$ 区间内, $\Delta P_c(k)$ 被标准化到 $[-0.1, 0.1]$ 区间内. ③ 模糊决策表的建立: 基于经验和领域专家的意见, 建立的模糊决策表如表 1 所示^[5]. ④ 控制器反模糊化信息表的建立: 其具体信息如表 2 所示. 图 1、图 2 中, NR 为负大; NL 为负较大; NM 为

负中; NS 为负小; ZE 为零; PS 为正小; PM 为正中; PL 为正较大; PR 为正大.

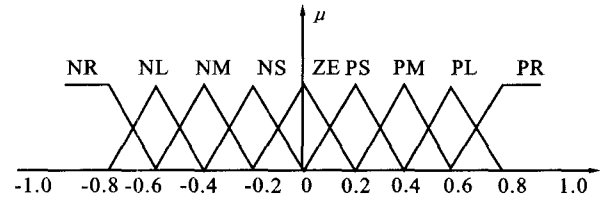


图 1 $\Delta^2\overline{\text{eval}}(V; k - 1)$ 和 $\Delta^2\overline{\text{eval}}(V; k)$ 的隶属函数

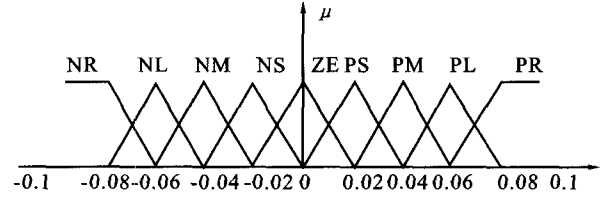


图 2 $\Delta P_c(k)$ 及 $\Delta P_m(k)$ 的隶属函数

表 1 参数 $P_c(m)$ 及 P_m 的模糊决策表

$\Delta^2\overline{\text{eval}}(V; k)$	$\Delta^2\overline{\text{eval}}(V; k - 1)$								
	NR	NL	NM	NS	ZE	PS	PM	PL	PR
NR	NR	NL	NL	NM	NM	NS	NS	ZE	NE
NL	NL	NL	NM	NM	NS	NS	ZE	ZE	PS
NM	NL	NM	NM	NS	NS	ZE	ZE	PS	PS
NS	NM	NM	NS	NS	ZS	ZE	PS	PS	PM
ZE	NM	NS	NS	ZE	PE	PS	PS	PM	PM
PS	NS	NS	ZE	ZE	PS	PS	PM	PM	PL
PM	NS	ZE	ZE	PS	PS	PM	PM	PL	PL
PL	ZE	ZE	PS	PS	PM	PM	PL	PL	PR
PR	ZE	PS	PS	PM	PM	PL	PL	PR	PR

表 2 参数 P_c 及 P_m 控制的反模糊化表

$Z(i, j)$	i								
	-4	-3	-2	-1	0	1	2	3	4
j	-4	-4	-3	-3	-2	-2	-1	-1	0
	-3	-3	-3	-2	-2	-1	-1	0	1
	-2	-3	-2	-2	-1	-1	0	0	1
	-1	-2	-2	-1	-1	0	0	1	2
	0	-2	-1	-1	0	2	1	1	2
	1	-1	-1	0	0	1	1	2	3
	2	-1	0	0	1	1	2	2	3
	3	0	0	1	1	2	2	3	4
	4	0	1	1	2	2	3	3	4

3.3 模糊逻辑器的控制过程

对 P_c 、 P_m 两个模糊控制器的具体控制过程为:

(1) 输入迭代 3 次 ($k - 2, k - 1, k$) 的平均适应度值 $\Delta^2\overline{\text{eval}}(V; k - 1)$ 及 $\Delta^2\overline{\text{eval}}(V; k)$.

(2) 标准化 $\Delta^2\overline{\text{eval}}(V; k - 1)$ 及 $\Delta^2\overline{\text{eval}}(V; k)$, 并设置下标 i 和 j 对应相关的控制参量, 其中反模糊化信息如表 2 所示.

(3) 计算控制参数 P_c 及 P_m 的变化量 $\Delta P_c(k)$ 和 $\Delta P_m(k)$, 其中 $\Delta P_c(k) = Z(i, j) \times 0.02$, $\Delta P_m(k) = Z(i, j) \times 0.002$, 而 $Z(i, j)$ 为 $\Delta^2 \text{eval}(V; k-1)$ 及 $\Delta^2 \text{eval}(V; k)$ 反模糊化的相应值, 具体对应关系如表 2 所示; 用 0.02 及 0.002 分别调整 P_c 和 P_m 的变化范围。

(4) 更新 P_c 和 P_m 的值

$$\begin{cases} P_c(k) = P_c(k-1) + \Delta P_c(k) \\ P_m(k) = P_m(k-1) + \Delta P_m(k) \end{cases} \quad (2)$$

式(2)中规定, $P_c(k)$ 的值不超出 $[0.5, 1.0]$ 的范围, $P_m(k)$ 的值不超出 $[0.0, 0.1]$ 的范围。

4 改进的遗传优化算法的验证

为了验证模糊逻辑控制器参数自适应调整的改进遗传优化算法的求解能力, 采用 Benchmark 中的测试函数对其进行验证, 即

$$\min f(x, y) = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1.0 + 0.001(x^2 + y^2)]^2}, \quad -100 \leq x, y \leq 100 \quad (3)$$

通过对公用的 Benchmark 函数(式(3))分析知, 函数 $f(x, y)$ 是极为困难的多峰值函数, 该函数仅在 $(0, 0)$ 处取最小值, 最小值为 0, 尽管该函数仅含一个最小值的解, 但在此解附近有无穷多个取局部极小值的解, 从而算法搜索极为困难^[6]。其具体的函数图像如图 3 所示。文中采用传统遗传算法及对交叉概率、变异概率采用模糊自适应参数动态调整的遗传算法进行对比计算。其试验结果如表 3 所示。从表 3 中数据可以看出, 采用模糊自适应参数动态调

整的遗传算法 4 次统计的优化结果的最大值为 0.030, 优化计算的结果基本在允许的误差范围内, 这一结果远比简单遗传算法得到的结果精度高; 从每次优化获得理想结果时算法运算的时间对比上看, 改进的算法基本上在 25s 以内, 而简单遗传算法占有的时间在 40s 以上; 同时进化最后收敛的角度看, 改进后的算法所用的进化代远少于传统遗传算法。

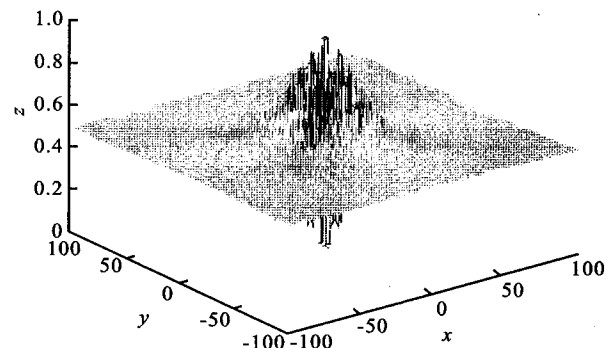


图 3 测试函数图像

5 结束语

采用遗传算法优化函数问题时, 为了获得复杂多峰函数的优化结果, 在进化的早期应动态地提高交叉算子 P_c 及变异算子 P_m 的值, 在优化的后期要动态地减少交叉算子 P_c 及变异算子 P_m 的值。为了动态的调整两者的变化值, 建立了 P_c 及 P_m 的两个模糊逻辑参数控制器, 依据进化个体变化的情况动态自适应的调整, 通过对测试函数的优化结果对比可知, 采用这种模糊逻辑自适应参数调整的遗传算法对优化复杂多峰函数能得到较高精度的优化解, 且进化的代数明显减少, 进化时所占用的时间也少, 因此优化效率高。

表 3 测试函数值对比表

简单遗传算法					模糊自适应参数动态调整的遗传算法				
进化代	x	y	进化时间 /s	$f_{\min}$	进化代	x'	y'	进化时间 /s	$f'_{\min}$
1013	-0.006	0.092	41.18	0.084	297	0.002	0.029	23.07	0.028
1089	-0.016	0.000	50.57	0.062	301	-2.026	-0.017	21.64	0.030
927	0.008	-0.004	45.35	0.095	287	-0.014	0.000	23.91	0.019
1102	-0.045	0.008	47.80	0.045	276	-0.003	-0.020	20.56	0.021

参考文献:

- [1] 英海燕, 王友波. 基于自适应遗传算法的模板匹配研究[J]. 微电子学与计算机, 2008, 25(1): 163-165.
- [2] 杨国军, 崔平远, 李琳琳. 遗传算法在神经网络控制中的应用与实现[J]. 系统仿真学报, 2001, 13(5): 567-570.
- [3] 夏冰, 冯建华. 基于遗传算法的测试访问机制最优化[J]. 微电子学与计算机, 2007, 24(6): 207-210.
- [4] Mitsuo G, Young S Y. Soft computing approach for reliability optimization: state-of-the-art survey[J]. Reliability Engineering & System Safety, 2006, 91(9): 1008-1026.
- [5] 过润秋, 洪旭, 苏旺旺. 基于模糊控制理论的二级倒立摆控制算法[J]. 西安电子科技大学学报, 2006, 33(1): 111-115.

(下转第 8 页)

表 2 测试函数(4) 与传统方法结果的比较

	文中算法	最好解			
		自适应 乘法法	Gen	Homaifar	GRG
x_1	78.08228	78.008	81.490	78.000	78.620
x_2	33.031439814512858	33.011	34.090	33.000	34.440
x_3	27.071944261449740	27.109	31.240	29.995	31.070
x_4	44.893914606454146	45.000	42.200	45.000	44.180
x_5	44.879107075616076	44.822	44.822	34.370	35.220
$f(x)$	- 3.103115687213954e + 004	- 31025.1412	- 30183.586	- 30665.609	- 30373.949

5 结束语

文中根据可行性规则求解约束优化的方法出发,利用自适应惯性权因子,适当应用罚函数确定其特殊适应度值,结合模拟退火的局部搜索算法,提出一种改进的混合粒子群算法来解决约束优化问题.从实验结果可以看出,文中算法有效可行,计算精度较高且解的稳定性较好,为解决复杂约束函数优化问题提供了一种行之有效的方法.但对算法的收敛性理论证明有待于进一步研究.

参考文献:

[1] Eberhart R C, Kennedy J. A new optimizer using particles swarm theory[C]// Proc. Sixth International Symposium on Micro Machine and Human Science. Japan, Nagoya, 1995:30-43.

[2] 王凌,刘波.微粒群优化与调度算法[M].北京:清华大学出版社,2008:59-93.

[3] 胡宏梅,董恩清.基于粒子群算法的码书设计研究[J].

微电子学与计算机,2009,26(1):97-100.

[4] 周鲜成,申群态,王俊年.基于微粒群的颜色量化算法[J].微电子学与计算机,2008,25(3):51-34.

[5] Deb K. An efficient constraint handing method for genetic algorithms[J]. Computer Methods in Applied Mechanics and Engineering, 2000(186):311-338.

[6] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Trans. on Evolutionary Computation, 2000(4):284-294.

[7] 谢柏桥,戴光明,石红玉.一种改进的求解约束函数优化问题的演化算法[J].计算机应用与软件,2008,25(7):48-50.

[8] 李炳宇,萧蕴诗,汪镭.PSO算法在工程优化问题中的应用[J].计算机工程与应用,2004,40(18):74-76.

作者简介:

裴胜玉 男,(1986-),硕士.研究方向为计算智能及其应用.

周永权 男,(1962-),博士,教授.研究方向为计算智能、神经网络及应用.

罗淇方 (1976-),工程师.研究方向为并行计算.

(上接第 4 页)

[6] Le Changyong, Yao Xin. Evolutionary programming using mutations based on the levy probability distribution[J]. IEEE Transactions on evolutionary computation, 2004, 8 (1):1-13.

作者简介:

罗 梅 女,(1974-),硕士,讲师.研究方向为软件工程、计算机算法及应用.

贾志娟 女,(1973-),硕士,副教授.研究方向为智能算法及信息挖掘.