文章编号: 1001-0920(2009)04-0542-05

一种基于有向交叉的遗传算法

范青武[a], 王 普[b], 高学金[b]
(北京工业大学 a. 实验学院, b. 电控学院, 北京 100124)

摘 要: 从解空间的角度分析了交叉算子的作用,针对其盲目搜索的缺陷,提出一种有向交叉遗传算子. 该算子通过优化控制交叉子代的落点位置,使交叉子代大概率地朝着最优解的方向进化. 实验表明,该算子显著地加快了遗传算法的寻优速度,提高了遗传算法定位最优解的精度.

关键词: 遗传算法; 交叉算子; 有向交叉
中图分类号: TP301.6 **文献标识码**: A

Improved genetic algorithm based on oriented crossover

FAN Qing-wu[a], *WANG Pu*[b], *GAO Xue-jin*[b]
(a. College of Pilot, b. College of Electrical Information and Control Engineering, Beijing University of Technology, Beijing 100124, China. Correspondent: FAN Qing-wu, E-mail: fqw@bjut.edu.cn)

Abstract: From the aspect of solution space, the role of crossover operators is analyzed. For the disadvantage of aimless serarch, an improved genetic algorithm based on oriented crossover is proposed, which can make the offspring individuals evolve towards the target value by optimizing their crossover positions. The evolving probability is very large. The simulation results show that the algorithm can improve greatly the efficiency and precision to find the optimum value.

Key words: Genetic algorithm; Crossover operators; Oriented crossover

1 引 言

几十年来,软计算一直是人工智能研究的热点. 遗传算法作为主要的软计算算法之一,具有较强的鲁棒性和通用优化能力,已广泛应用于材料、化工、交通等各个领域[1]. 作为一种通用优化算法[2],人们希望遗传算法能高效地处理所有的优化问题,但研究表明标准遗传算法存在收敛速度慢、精度不高等缺陷[3,4].

如何提高遗传算法的搜索能力和收敛速度是一个重要的问题. 目前有很多学者提出不同的改进方法,如自适应遗传算法、级连遗传算法、并行遗传算法、混合遗传算法等[5-9]. 但这些改进均受限于遗传算法基础理论的薄弱,局限于对某一特定算法或算法的某个环节、某种性能的研究,其普适性较差.

目前急需对遗传算法的本质与基本理论作透彻的研究. 在遗传算法中,交叉算子通过模拟自然界生物的杂交过程对个体进行杂交操作,不断产生新个体、增加种群的多样性、扩大寻优范围,从而使得遗传算法具有较强的搜索能力. 因此,交叉算子是遗传算法的核心,是决定算法收敛性能的关键. 本文首先从解空间的角度分析了交叉算子的实质,针对其盲目搜索的缺点,提出了一种有向交叉遗传算子. 测试算例表明,该算子显著加快了遗传算法的寻优速度,提高了遗传算法定位最优解的精度.

2 交叉操作特征分析

交叉操作的设计与问题的编码紧密相关. 遗传算法的编码策略包括至今仍在争论的两派: 一派根据模式定理建议用尽量少的符号编码,该派以二进制编码为代表; 一派以数值优化计算的方便和精度为准采用一个基因一个参数的方法,该派以十进制编码为代表. 实际应用表明,基于两派编码的交叉操作各有优缺点,均对某些优化问题表现出良好的效果. 探讨基于两派编码下交叉操作的共性特征,无疑有助于对遗传算法本质的认识.

二进制交叉通常使用的算子包括一点交叉、二点交叉和一致交叉等形式. 为了讨论方便,给出二进

收稿日期: 2008-03-14; **修回日期**: 2008-06-05.
基金项目: 国家自然科学基金项目(60704036); 北京工业大学青科基金项目(X1024000200801).
作者简介: 范青武(1977—),男,山西平遥人,讲师,博士,从事模式识别与智能控制等研究; 王普(1962—),男,合肥人,教授,博士生导师,从事人工智能、自动控制等研究.

制交叉的定义.

定义 1　设 $S=\{0,1\}^l$ 为个体空间,$U=S$ 为指标集,对于 $(X,Y)\in S^2$,其中 $X=(x_1,x_2,\cdots,x_l)$,$Y=(y_1,y_2,\cdots,y_l)$,记

$$X\oplus Y=(x_1\oplus y_1,x_2\oplus y_2,\cdots,x_l\oplus y_l),$$

$$X\otimes Y=(x_1\otimes y_1,x_2\otimes y_2,\cdots,x_l\otimes y_l),$$

$$\overline{X}=(1\oplus x_1,1\oplus x_2,\cdots,1\oplus x_l)=1\oplus X.$$

其中

$$x_i\oplus y_i=\begin{cases}1, & x_i\neq y_i;\\ 0, & x_i=y_i;\end{cases}$$

$$x_i\otimes y_i=\begin{cases}1, & x_i=y_i=1;\\ 0, & \text{otherwise}.\end{cases}$$

对于 $u\in U$,记 $\Phi_u(X)=u\otimes X$,$\bar{\Phi}_u(X)=(1\oplus u)\otimes X$,$h(X,Y)=X\oplus Y$,其中 $1=(1,1,\cdots,1)\in S$ 表示各分量恒为 1 的向量. 于是称 $X'=C_u(X,Y)=h(\Phi_u(X),\bar{\Phi}_u(Y))$ 和 $Y'=\overline{C}_u(X,Y)=h(\Phi_u(Y),\bar{\Phi}_u(X))$ 为母体 $(X,Y)\in S^2$ 的交叉后代.

定义 2

$$X\times Y=G;$$

$$G_i=\begin{cases}x_i, & x_i=y_i;\\ *, & \text{otherwise}.\end{cases}$$

其中符号"*"代表任意字符,即 0 或 1. 则称 G 所代表的模式为以染色体 X 和 Y 为祖先的"家族".

根据二进制交叉的定义,有如下关系:$x_i'=u_i\otimes x_i\oplus\overline{u_i}\otimes y_i$,$y_i'=u_i\otimes y_i\oplus\overline{u_i}\otimes x_i$. 若 $x_i=y_i$,则有 $x_i'=y_i'=x_i$.

可见,二进制交叉的子代个体必属于模式 G. 由模式定义可知,每个模式实际上定义了一组具有共同特征的位串集合,所以 G 为二进制交叉的子代可行解集合.

十进制交叉操作有点式交叉、离散交叉和多种数值型交叉. 因为点式交叉和离散交叉都是分量值的交换,而且实数个体每位代表的空间大,这两种交叉方法表达的精细度差、搜索效率低,所以数值型交叉算子应该作为实数遗传算法的基本交叉方法. 同样给出十进制交叉的定义.

定义 3　设 $X=(x_1,x_2,\cdots,x_k)$,$Y=(y_1,y_2,\cdots,y_k)$ 为两个十进制编码父代个体,$\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_k)$ 为算术交叉因子,其中 $0\leq\alpha_i\leq 1$,k 为问题维数.

记

$$x_i'=\alpha_i x_i+(1-\alpha_i)y_i,\quad i=1,2,\cdots,k,$$

$$y_i'=(1-\alpha_i)x_i+\alpha_i y_i,\quad i=1,2,\cdots,k.$$

称 $X'=C_\alpha(X,Y)=(x_1',x_2',\cdots,x_k')$ 和 $Y'=\overline{C}_\alpha(X,Y)=(y_1',x_2',\cdots,x_k')$ 为父代 X,Y 的交叉后代. 不妨设 $x_i<y_i$,令 $y_i-x_i=\Delta$,则根据十进制交叉算子的定义,有

$$x_i'=x_i+(1-\alpha)\Delta,\quad y_i'=x_i+\alpha\Delta.$$

由于 $0\leq\alpha\leq 1$,有

$$x_i\leq x_i'\leq y_i,\quad x_i\leq y_i'\leq y_i.$$

可见,十进制交叉操作后子代个体必落在区间 $[x_i,y_i]$ 中.

根据上面的讨论可以发现,二进制交叉和十进制交叉有一个共同的特点:交叉结果实质上是在父代个体隐含的解空间中取值,即父代个体唯一地决定了子代个体的落点范围.

为了研究父代个体所隐含的解空间是否代表了最优解的集合,作如下统计分析:随机生成两个个体,分别考察在二进制交叉和十进制交叉作用下生成交叉优秀解的概率,其中交叉优秀解指经交叉操作后生成的性能优于父代个体的子代个体. 显然,生成交叉优秀解的概率越大,交叉操作的进化能力便越强,优化效率越高. 所采用的考察函数分别为文献[10]中的 $F0$,$F1$ 和 $F13$. 其中:$F0$ 为典型的单调函数,$F1$ 为典型的单峰函数,$F13$ 为典型多峰函数. 重复上述操作 10000 次,对结果进行概率统计分析,统计结果列于表 1.

表 1　各种交叉性能统计结果

	二进制交叉	十进制交叉
$F0$	0.2395	0
$F1$	0.2900	0.3091
$F13$	0.2961	0.3010

结果表明,从统计的角度而言,对于单调函数,二进制交叉的性能优于十进制交叉的性能,十进制交叉搜索到种群优秀解的概率为 0,这时,种群的进化将不得不依赖于变异操作;对于单峰函数和多峰函数,十进制交叉生成种群优秀解的能力优于二进制交叉. 但无论是二进制交叉还是十进制交叉,均表现出随机取值的特点.

综上所述,可以得出如下结论:交叉算子的实质是在父代个体隐含的解空间中随机取值,因而不能保证交叉操作后的子代个体优于父代个体,体现出盲目搜索的特点.

3　有向交叉算法原理

一个重要的问题是:如何以优化控制的方式产生交叉后代,使得交叉子代可以大概率地朝着最优解的方向进化?下面进一步分析交叉子代落点位置的特点. 不妨设 $a\leq x\leq y\leq b$,$a\leq x'\leq y'\leq b$,如图 1 所示. 研究发现交叉子代的落点位置有如下两种情况:

1)　Ⅰ区:$x\leq x'\leq y'\leq y$;

2) Ⅱ区: $a \le x' \le x$ 且 $y \le y' \le b$. 即子代个体要么都落在父代个体之间,要么都落在父代个体两侧.

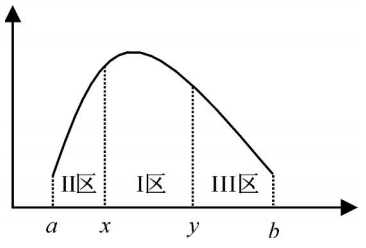

图 1 交叉子代落点特性

对于单调函数,如图 2 所示. 显然,交叉后必在Ⅱ区找到优秀解.

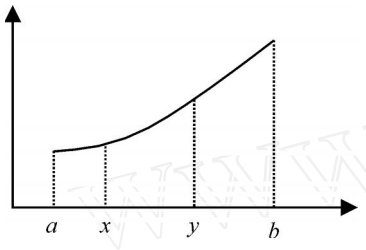

图 2 单调函数交叉特性

对于单峰函数,如图 3 所示,其中 x^* 为峰顶点对应的个体. 分以下几种情况考虑:

1) $x \le y \le x^*$ 或 $x^* \le x \le y$. 该情况类似于单调函数,故交叉后将在 Ⅱ区找到优秀解.

2) $x < x^* < y$. 不妨设 $f(x) > f(y)$,由于父代个体落于峰顶的两侧,必能找到一点 x'',使得 $x' \in [x, x'']$ 时,$f(x') > f(x)$; $x' \in [x'', y]$ 时,$f(x') < f(x)$. 令 $|f(x) - f(y)| = \Delta$,要使 x' 大概率地落入 $[x, x'']$ 区间内,即使得交叉子代大概率地优秀,必须满足 Δ 很小. 由于选择算子的本质是使种群趋向于种群中适应值大的个体的集合[11],在选择算子的不断作用下,种群中个体间的 Δ 将趋于 0. 由此可见,该情况经交叉后大概率地在区间 Ⅰ找到优秀解.

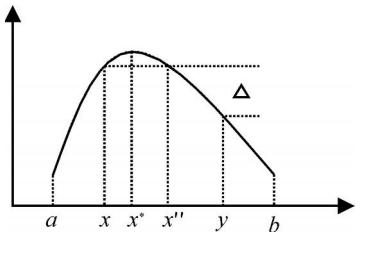

图 3 单峰函数交叉特性

对于多峰函数(如图 4 所示),在进化初期,种群中代表各峰的个体依据适应值的大小进行竞争生存,这个时期在 Ⅰ区和 Ⅱ区均有可能找到优秀解;随着进化的进行,种群中的个体逐渐聚集于某一个适应值高的山峰,多峰函数转变为单峰特点或单调特点.

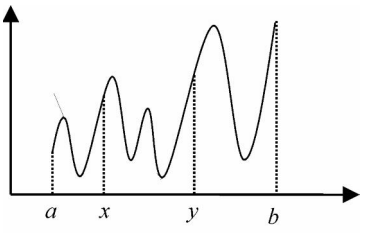

图 4 多峰函数交叉特性

通过上述分析,得到一个启发,如果能优化控制交叉子代的落点位置,将交叉定向到能获得优秀解的 Ⅰ(或 Ⅱ) 区间,那么交叉后的子代将大概率地优于父代个体. 根据这个启发,设计出一种新的交叉 - 有向交叉算法,其基本思想是:对父代个体分别在 Ⅰ和 Ⅱ区进行交叉生成准子代个体,并选择性能好的两个个体作为本次交叉的子代个体. 其算法流程如下:

Step1: 确定种群规模 N,交叉概率 p_c,变异概率 p_m,变量取值范围 $[a, b]$ 等参数.

Step2: 置 $k = 0$,随机产生初始种群.

Step3: 基于适应值建立群体中每一个个体的选择概率 $f_i / \sum f_i$,采用蒙特卡罗方法对群体作 N 次随机选择,将选出的个体放入"配对池".

Step4: 配对池中的个体两两结合形成 $N/2$ 对家庭.

Step5: 对每一个家庭以概率 p_c 进行有向交叉操作:

1) $\alpha_1 = \text{rand}(1)$,随机生成 $\alpha_1 \in [0, 1]$;

max = max(X, Y),取父代个体中的较大值赋予 max;

min = min(X, Y),取父代个体中的较小值赋予 min;

temp1 = b - max,计算较大个体与上限的距离;

temp2 = min - a,计算较小个体与下限的距离;

delt = min(temp$_1$, temp$_2$),取距离的较小值;

$X_1' = \text{max} + \alpha_1 \text{delt}$,生成中间个体 X_1'; $Y_1' = \text{min} - \alpha_1 \text{delt}$,生成中间个体 Y_1'.

2) $\alpha_2 = \text{rand}(1)$,随机生成 $\alpha_2 \in [0, 1]$;

$X_2' = \alpha_2 X + (1 - \alpha_2) Y$,生成中间个体 X_2';

$Y_2' = \alpha_2 Y + (1 - \alpha_2) X$,生成中间个体 Y_2'.

3) 比较 X_1', X_2', Y_1', Y_2' 的适应值大小,取两个最大值为子代个体.

Step6: 以概率 p_m 对交叉后的个体进行变异操作.

Step7: 检验停止准则,若满足,则停止;否则 k

$= k + 1$ 并返回 Step3.

4　实验验证

为了验证算法的效果,选择 4 个典型的遗传算法(GA)性能测试函数,分别采用标准遗传算法(SGA),自适应遗传算法(AGA)和本文提出的有向交叉遗传算法(DGA)进行测试,并比较它们的性能.因为很多应用(如实时控制)往往要求优化算法在给定的有限时间内求得最优解,所以在测试中,比较两种算法在给定进化代数内求得最优解的能力.考察的指标为搜索到最优解的概率和最优解的平均值.搜索到最优解的概率越高,表明算法的搜索效率越高,性能越稳定;最优解的平均值与理论最优解越接近,表明算法定位最优解的精度越高.表 2 为所选用的测试函数列表.

表2中,$F0$为严格单调函数类代表,在 $x^* = 1$ 处达到理论极大值 1;$F13$ 为典型的不等高多峰问题,在 x^* 等于0.0797,0.2467,0.4506,0.6814,0.9339 处表现为多个非等高、非等距的极大值 0.9991,0.9545,0.7662,0.4809,0.2217,这类问题的 GA 搜索取决于峰顶和谷底的高度差距,以及谷底之间的高度差距,但区分能力比较弱,也视为高程度的 GA-困难问题;$F2$ 是一个典型的非线性可分函数,在 $x^* = (1,1,\cdots,1)$ 处达到极小值 0,以与最优解位串适应值之差小于 0.01 为满意最优解,出现 GA 早熟的比例非常高,属于高程度 GA-困难问题,采用 GA 搜索到全局最优解将是一个小概率事件;$F7$ 函数是模式欺骗很大的一类线性不可分函数,在 $x^* = (0,0)$ 时达到极大值1,属于高难度 GA-困难问题,采用 GA 搜索到全局最优解将是一个小概率事件.

表 2　选用的测试函数

	采用的测试函数
$F0$	$\max f(x) = x^{1/5}, x \in [0,1]$
$F13$	$\max f(x) = e^{-2\times\ln 2\times((x-0.1)/0.8)^2}\sin^6(5\pi(x^{3/4}-0.05))$, $x_i \in [-10,10]$
$F2$	$\min f(x) = \sum_{i=1}^{n-1}[100\times(x_{i-1}-x_1^2)^2+(x_i-1)^2]$, $x_i \in [-5.12,5.12]$
$F7$	$\max f(x) = 0.5 - \frac{\sin\sqrt{x_1^2+x_2^2}-0.5}{[1+0.001\times(x_1^2+x_2^2)]^2}$, $x_i \in [-10,10]$

测试参数设置为:$F0$ 和 $F13$ 的染色体串长设定为 $L = 50$,$F2$ 和 $F7$ 的染色体串长设定为 $L = 80$;$F0$ 的群体规模设定为 $N = \{10,20,30\}$,$F13$ 的群体规模设定为 $N = \{20,40,60\}$,$F2$ 和 $F7$ 的群体规模设定为 $N = \{40,80,120\}$;设定交叉概率为 $p_c = 0.6$,变异概率 $p_m = 0.02$;设定最大进化代数为 120,每种情况下的实验计算重复 100 次.测试结果列于表 3 ～ 表 6.

通过测试结果,可以得出如下结论:

1)在其他条件相同的情况下,随着种群规模的增大,各种遗传算法所表现出的性能均有不同程度的改善;随着函数复杂程度由低到高变化,其种群规模设定值也应由小到大变化.

表 3　SGA,AGA 和 DGA 对 F0 的优化性能比较

种群规模	最优解	收敛率/%			最优解的平均值		
		SGA	AGA	DGA	SGA	AGA	DGA
10	> 0.99999	1	12	100	0.9904611787	0.999466743	0.9999998069
20	> 0.99999	1	21	100	0.9949213587	0.999825582	0.9999999610
30	> 0.99999	3	23	100	0.9980845922	0.999861088	0.9999999879

表 4　SGA,AGA 和 DGA 对 F13 的优化性能比较

种群规模	最优解	收敛率/%			最优解的平均值		
		SGA	AGA	DGA	SGA	AGA	DGA
20	> 0.9991	41	56	99	0.9867883370	0.9927995566	0.9993260948
40	> 0.9991	52	78	100	0.9945768731	0.9977253440	0.9996120529
60	> 0.9991	72	84	100	0.9973695117	0.9981061258	0.9996122764

表 5　SGA,AGA 和 DGA 对 F2 的优化性能比较

种群规模	最优解	收敛率/%			最优解的平均值		
		SGA	AGA	DGA	SGA	AGA	DGA
40	< 0.01	7	40	91	0.1778727400	0.0278092600	0.0037800700
80	< 0.01	24	62	97	0.0791409200	0.0125244300	0.0020749800
120	< 0.01	34	68	99	0.0373975900	0.0112046200	0.0015494500

表 6 SGA, AGA 和 DGA 对 F7 的优化性能比较

种群规模	最优解	收敛率 / %			最优解的平均值		
		SGA	AGA	DGA	SGA	AGA	DGA
40	> 0.999	1	25	59	0.9909798321	0.9932268488	0.9965209272
80	> 0.999	6	37	74	0.9921302394	0.9944016154	0.9986545033
120	> 0.999	9	48	92	0.9921708686	0.9958255727	0.9993284684

2) 在相同的种群规模下,对 4 个测试函数分别运行 100 次,相对于 SGA 和 AGA,DGA 的收敛率均有较大的提高,表明 DGA 鲁棒性更强,性能更稳定;同时,相对于 SGA 和 AGA,DGA 的解的质量均有数量级的提高,表明 DGA 不仅具有更快地寻找最优解的速度,而且具有更精确地定位最优解的能力.

5 结 论

作为一种具有全局寻优能力的随机优化算法,遗传算法能以较大的概率搜索到全局最优解,但它收敛速度慢,精度不高,有必要寻求速度快、精度高的改进算子来弥补这一缺陷. 分析得出交叉算子的实质是在父代个体隐含的解空间中随机取值,因而其不能保证交叉操作后的子代个体优于父代个体,体现出盲目搜索的特点. 针对这一缺点,本文提出了一种有向交叉遗传算法,该算法通过优化控制交叉子代的落点位置,使交叉子代得以大概率地朝着最优解的方向进化. 测试算例表明,该算法操作简单,并且显著地加快了遗传算法的寻优速度,提高了遗传算法定位最优解的精度,具有较好的应用前景和推广价值.

参考文献(References)

[1] Man K F, Tang K S, Kwong S. Genetic algorithms: Concepts and applications[J]. IEEE Trans on Industrial Electronics, 1996, 43(5): 519-534.

[2] Goldberg D E. Genetic algorithms in search, optimization and machine learning[R]. Addison Wesley, 1989.

[3] Forgel D B. An introduction to simulated evolutionary optimization [J]. IEEE Trans on Neural Networks, 1994, 5(1): 85-95.

[4] Xu Z B, Li G. The simulated-biology-like algorithms for global optimization[J]. J of Operation Research, 1995, 14(2): 1-13.

[5] Bingul Z, Sekmen A, Zein-Sabatto S. Evolutionary approach to multi-objective problems using adaptive algorithms[C]. Proc of the IEEE Int Conf on Systems, Man and Cybernetics. Piscataway, 2000: 1923-1927.

[6] Gautam G, Chaudhuri B B. A cascaded genetic algorithm for efficient optimization and pattern matching [C]. Proc of the 2nd Int Conf on Advances in Pattern Recognition. Berlin: Springer-Verlag, 2001: 32-39.

[7] Shisanu T, Prabhas C. Parallel genetic algorithm with parameter adaptation [J]. Information Processing Letters, 2002, 82(1): 47-54.

[8] Bianco C G L, Piazzi A. A hybrid algorithm for infinitely constrained optimizaton[J]. Int J of Systems Science, 2001, 32(1): 91-102.

[9] Renders J M, Flasse S P. Hybrid methods using genetic algorithms for global optimization[J]. IEEE Trans on System, Man and Cybernetics, 1996, 26(2): 243-258.

[10] 李敏强, 寇纪淞, 林丹, 等. 遗传算法基本理论与应用[M]. 北京: 科学出版社, 2002.
(Li M Q, Kou J S, Lin D, et al. Genetic algorithm theory and applications[M]. Beijing: Science Press, 2002.)

[11] 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.
(Wang X P, Cao L M. Genetic algorithm-theory, application and software implementation[M]. Xi'an: Xi'an Jiaotong University Press, 2002.)