

基于进化遗传算法的优化计算*

陈明

(石油大学计算机科学与技术系 北京 102200)

摘要 针对经典遗传算法在优化计算中存在的弊端,提出进化遗传算法.在染色体基因位编码方式、交叉与变异算子、适合度函数设计、候选解染色体选取方式及收敛准则等方面做了改进.通过对极难优化的 Shekel 类函数等的优化计算,说明该算法具有良好的全局搜索能力和较快的收敛速度.

关键词 遗传算法, Shekel 类函数, 优化计算, 进化策略, 收敛准则.

中图分类号 TP18

遗传算法(Genetic Algorithm)^[1]是近几年发展起来的一种随机全局优化算法.它是基于达尔文生物进化论的自然选择学说和群体遗传学原理而建立的.

在工程科学中存在很多困难的组合优化问题,这些问题大都是非线性的、甚至是不可微的,常规的优化技术无法有效地求解,只能对问题进行简化近似处理.遗传算法对求解问题只要求是可计算的,无可微性及其他要求,且较易获得全局最优解.^[2]

1 经典遗传算法

遗传算法是一个多点并行的迭代过程,在每次迭代中都进行如下操作:将一组以一定的基因形式描述的候选解进行交叉和变异操作以及适应环境能力的评价;选取参与产生后代的候选解.重复此过程,直至满足某种收敛准则而得到全局最优解.

对于求函数 F 的极小值问题,其目标函数为

$$\min F = F(X), \quad X = (x_1, x_2, \dots, x_i, \dots, x_n)^T, \quad X \in \Omega, \quad \forall x_i \in D_i, \quad D_i = [a_i, b_i],$$

其中 Ω 为自变量的定义域, F 为目标函数值,与解的优劣程度相关.

1.1 适合度函数

为使各候选解在定义域内的适合度函数值始终为正,并且目标函数值越小,则适合度函数值越大,可将适合度函数定义为 $\text{eval}(X) = C - F(X)$, C 为正常数.

进而优化问题作为求 $\max \text{eval}(X)$.

1.2 染色体

以一定长度的二进制数字串表示为染色体的一个候选解.染色体的长度与自变量的个数、自变量的取值范围及求解精度有关.

若 x_i 优化求解精度为小数点后 g_i 位,则其二进制编码长度为 $(b_i - a_i) \times 10^{g_i} \leq 2^{m_i} - 1$, 其中 m_i 为满足上述不等式的最小正整数.

x_i 的二进制编码 $\langle x_i \rangle_2$ 满足

$$\text{decimal}(\langle x_i \rangle_2) = \frac{x_i - a_i}{b_i - a_i} (2^{m_i} - 1).$$

将 $\langle x_1 \rangle_2, \langle x_2 \rangle_2, \dots, \langle x_i \rangle_2, \dots, \langle x_n \rangle_2$ 连接,构成候选解的染色体的二进制数字串长度为 $m = \sum_{i=1}^n m_i$, 其中 m_i 位对应 $[a_i, b_i]$ 的数据范围.

* 本文研究得到国家自然科学基金、国家 863 高科技项目基金、中国科学院机器人学开放研究实验室基金和德国科学联合会 DFG 基金资助.作者陈明,1949 年生,教授,主要研究领域为软计算与分布并行计算.

本文通讯联系人:陈明,北京 102200,石油大学计算机科学与技术系

本文 1997-03-28 收到原稿,1997-10-21 收到修改稿

1.3 初始化候选解

利用随机数发生器产生数量为 n_{pop} (染色体数) 的、位数为 m 的二进制数字串 U^t . U^t 表示初始代 $t=0$ 时的 n_{pop} 个候选解.

一般而言, 初始代候选解性能很差, 遗传算法能从这组候选解开始, 模拟进化过程, 择优汰劣, 最终得出优秀的群体与染色体, 完成优化计算.

1.4 候选解的适合度评价

按编码规则

$$x_{ki} = a_i + \text{decimal}(\langle x'_{ki} \rangle_2) \frac{b_i - a_i}{2^m - 1} \quad k=1, 2, \dots, n_{pop} \quad i=1, 2, \dots, n$$

将 t 代中候选解 k 的 $\langle x'_{ki} \rangle_2$ 解码, 并将对应数字代入 $\text{eval}(X) = C - F(X)$ 中, 计算各候选解的 $\text{eval}(X)_t$.

1.5 适合度比例信息选择

以一定概率从候选集 U^t 中选取 n_{pop} 对染色体作为双亲繁殖下一代候选解, 其概率与各候选解的适合度成比例, 适合度大的候选解得到更多的机会繁殖后代, 使优良特性得以遗传. 体现了适者生存、不适者淘汰的思想.

1.6 交叉(Crossover)

简单交叉分为两步. 首先以一定的概率 p_c 随机选取两个染色体数字串作为双亲, 然后进行交叉操作. 设串长为 l , 随机选取一整数 $k, k \in [1, l-1]$, 通过交换串中从 $k+1$ 至 l 的所有字符, 从而产生两个新串.

1.7 变异

以一定概率 p_m 随机选取交叉操作后的某一染色体数字串, 对其中的某一或几个基因取反运算后的结果就是变异操作后得到的新数字串. 变异操作体现了生物进化过程中的基因突变.

由上述可以看出, 基于二进制染色体描述方式的经典遗传算法, 求解数值优化问题存在以下不足:

- 候选解的编码过程是优化变量的离散化过程, 这必然影响优化精度;
- 求解过程中频繁编码、解码将导致增加计算工作量;
- 为了增加精度势必增加数字串长度, 因此, 计算精度与计算量之间存在着矛盾;
- 对于多变量优化问题, 其候选解染色体数字串过长, 这将影响算法特性;
- 算法求解过程不直观, 无法实时监视优化过程;
- 对于解空间域不定的优化问题求解困难.

为了克服经典遗传算法求解数值优化问题的不足, 本文提出采用实数描述染色体的进化遗传算法.

2 进化遗传算法

进化遗传算法采用实数数码染色体基因以及实数码交叉和变异遗传算子.

定义 2.1. 染色体向量 $X = (x_1, x_2, \dots, x_i, \dots, x_n)^T$, $\forall x_i \in R, i=1, 2, \dots, n$, 其中 x_i 表示染色体 X 的集合的第 i 个基因位.

定义 2.2. 种群 U 为染色体的集合, 即 $U = \{X_j\}$, $j=1, 2, \dots, n_{pop}$, 其中 n_{pop} 为每一代染色体数.

定义 2.3. 交叉操作(凸组合)为

$$C = A \otimes_{rd} B,$$

$$c_i = \begin{cases} \alpha a_i + (1-\alpha)b_i, & \text{if } rd=0 \\ \alpha b_i + (1-\alpha)a_i, & \text{if } rd=1 \end{cases}$$

$$i=1, 2, \dots, n \quad \forall a_i \in A, \forall b_i \in B, \quad A, B \in U, 0 < \alpha < 1.$$

其中 α 可以是确定的或随机选取之值; rd 为随机选取的二进制数; \otimes_{rd} 为交叉操作符.

由凸组合的特点可知, 若 $A, B \in D^n$, 则 $C \in D^n$, 可以使解的搜索在一定的空间内进行. 因此, 进化遗传算法的交叉操作保证在已搜索到的解空间不断寻优.

定义 2.4. 变异操作为

$$E = \nabla_{rd} \cdot D,$$

$$e_j = \begin{cases} d_j + \Delta(t, \delta), & \text{if } rd=0 \\ d_j - \Delta(t, \delta), & \text{if } rd=1 \end{cases} \quad j \in \{1, 2, \dots, n\} \quad D \in U, \forall d_i \in D.$$

其中 ∇_{rd} 为变异操作符, rd 为随机选取的二进制数; $\nabla(t, \delta)$ 为变异的调整函数, 是进化代和目标函数波动度 δ 的

函数.

变异操作能够不断开拓问题解的新空间,体现了算法的全局搜索能力.

定义 2.5. 适合度函数为 $eval(X) = f(MAX) - f(X)$, 其中 MAX 为当前种群 U 中的目标函数值最大的染色体.

这样定义的适合度函数能够保证各候选解染色体适合度函数值的非负及极大化,并使环境条件随遗传代的加深而变得苛刻,体现了自然界的变化情况.

进化遗传优化算法描述如下.

- S1: 随机产生初始解向量集 $U^t = \{X_j^t, j=1, 2, \dots, npop, t$ 表示进化代,
- S2: 根据适合度函数定义, 计算染色体(候选解)适合度函数值 $eval(X_j^t), j=1, 2, \dots, npop$.
- S3: 根据各染色体的比例信息

$$P_j^t = eval(X_j^t) / \sum_{l=0}^{npop} eval(X_l^t)$$

选取参与产生下一代染色体的解向量 $X_j^{t+1} (j=1, 2, \dots, npop)$.

显然, 适合度较大的候选解参与产生下一代的概率大, 为了加快算法的收敛速度, 将每代中适合度值最大的候选解强行传到下一代, 而不受选择过程的限制.

- S4: 在向量 $X^1, X^2, \dots, X^j, \dots, X^i, \dots, X^{npop}$ 中以一定的概率 p_c 随机选择两向量 X^i, X^j 交叉操作. 随机保留交叉操作后的一向量而丢弃另一向量, 得到候选解 $\{X^1, X^2, \dots, X^i, \dots, X^j, \dots, X^{npop}\}$.
- S5: 从交叉操作后得到的向量集中, 以一定的概率 p_m 随机选取某一向量中的某一分量 X^m , 并对其实行变异操作, 得到下一代的候选解集

$$\{X_1^{t+1}, X_2^{t+1}, \dots, X_{npop}^{t+1}\}.$$

- S6: 设 t 和 $t+1$ 代候选解集中适合度值最高的两解向量 X_{max}^t 与 X_{max}^{t+1} 间的欧氏距离为

$$d = \|X_{max}^t - X_{max}^{t+1}\| = \sqrt{(X_{max1}^t - X_{max1}^{t+1})^2 + (X_{max2}^t - X_{max2}^{t+1})^2 + \dots + (X_{maxn}^t - X_{maxn}^{t+1})^2},$$

若 $d \leq \epsilon$, 则 $X^* = X_{max}^{t+1}$, 得到全局最小点, 算法收敛; 否则, 转至 S2 继续进行解的优化搜索.

可以看出, 基于实数码候选解染色体基因的进化遗传算法在求解数值优化问题时, 不受解空间域的限制, 故其适于解空间域不定的优化问题的求解.

3 实例计算

下面通过几个实例计算, 说明进化遗传算法的能力.

3.1 Goldstein & Price 函数

$$\begin{aligned} \min f(x_1, x_2) = & (1 + (x_1 + x_2 - 1)^2)(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \cdot \\ & (30 + (2x_1 - 3x_2)^2)(18 - 32x_1 + 12x_1^2 + 48x_2^2 - 36x_1x_2 + 27x_2^2) \\ & - 2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2 \end{aligned}$$

上述函数在定义域内存在 4 个局部极小点, 最优解为 $X^* = (0, -1)$, $f_{min} = 3$.

采用进化遗传算法, 不论初始代候选解处于何种位置, 最后都进化, 收敛到最优解. 如图 1 所示.

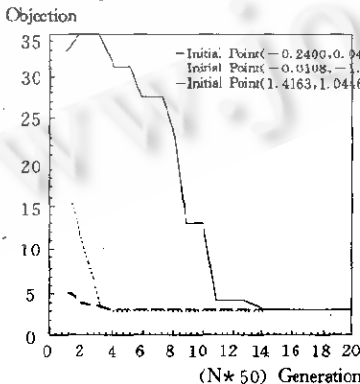


图1 目标函数值随遗传代的进化

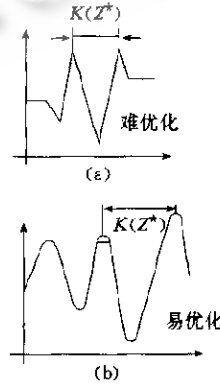


图2 难与易优化函数

3.2 Shekel 函数

Shekel 类函数是一类极难优化的函数. 其原因是, 相对于整个变量域, 其全局极小点的吸引域范围很窄. 图 2 所示的是难、易优化函数的比较. 其中 $K(Z^*)/K(Z) < \epsilon, K(Z^*)$ 为全局最小点的吸引域, $K(Z)$ 为整个变量区.

为了更为直观, 选用 shekel 类二维函数, 其函数曲面图和等值线图如图 3 所示. 此函数存在 3 个极小点, 分别在 (1, 1), (4, 4) 和 (8, 8) 附近, 其中全局最优点在 (4, 4) 附近.

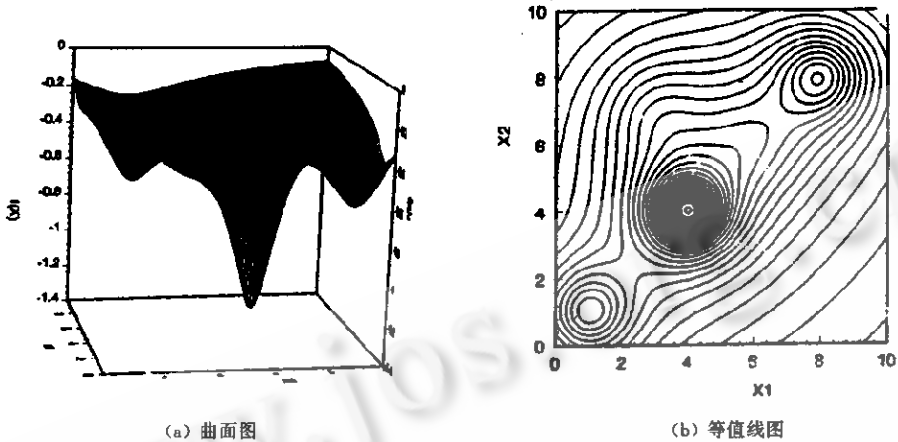


图 3 二维 shekel 类函数

在远离全局最优点和局部极小点的吸引域内选取初始代候选解, 经过不同的进化途径, 均能收敛于全局最优点 $X^* = (4.010\ 386, 4.009\ 744)$, $f_{\min} = -1.23$.

4 结束语

典型实例的优化计算表明, 本文提出的进化遗传算法具有良好的全局搜索能力和较快的收敛速度, 并在参数选择方面获得下述一些结果.

(1) n_{pop} 的选取对遗传算法性能影响很大, n_{pop} 选择过小, 算法的全局搜索能力将受到影响, 可能会过早地收敛于局部最小点; n_{pop} 选择过大, 将降低算法的收敛速度, 造成计算资源上的浪费.

(2) 交叉操作体现了算法的局部搜索特性. 对不同问题, 交叉操作概率 P_c 的最佳数值不尽相同, 但其最佳范围为 $P_c = 0.4 \sim 0.7$, P_c 过大或过小都将影响算法特性. 如对 Shekel 类函数, 若 P_c 选择过大, 进化搜索过程将产生振荡, 若 P_c 选择过小, 将降低算法的搜索速度.

(3) 变异操作体现了算法的空间搜索能力, 它不断开拓解的新空间并具有使解跳出局部极小的功能. 对不同问题, P_m 的最佳值不尽相同. 一般说来, P_m 的选取小于 0.2. 对于 Shekel 函数, 若 P_m 选择过大, 则进化搜索过程将发生振荡甚至发散; 若 P_m 选择过小, 将陷入局部最小.

参考文献

- 1 Holland J H. Adaptive of Natural and Artificial Systems. Ann Arbor: The University of Michigan Press, 1975
- 2 Goldberg D. Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison Wesley, 1989

Optimization Computing Based on Evolution Genetic Algorithm

CHEN Ming

(Department of Computer Science and Technology University of Petroleum Beijing 102200)

Abstract Deeply analyzed the conventional genetic algorithm and for its shortcomings on numerical optimization, evolution genetic algorithm (EGA) is proposed. EGA makes some improvements on description of genes of chromosome, genetic operators of crossover and mutation, designing of fitness function, selection method on chromosome of candidate solutions and convergence criterion. The optimization results of some functions (including hard Shekel' function) show that EGA has fine ability of global searching and speedy convergence.

Key words Genetic algorithm, Shekel' function, optimization computing, evolution storage, convergence criterion.