ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.7, No.19, July 2011.
E-mail: eduf@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

基于遗传算法的最优化问题求解

宋杰鹏[1,2]

(1.中国矿业大学,江苏 徐州 221116;2.徐州师范大学,江苏 徐州 221116)

摘要:遗传算法是一种有效的解决最优化问题的方法,在解决复杂的全局优化问题方面,遗传算法已得到了成功的应用。对遗传算法的基本步骤进行总结,通过最优化问题求解实例描述了遗传算法的具体运行过程,包括产生初始染色体、染色体评价、选择、交叉、变异等。分别应用 VC 和 VB 两种语言进行编程实现,结果表明,VC 语言在运算效率和结果优度方面均比 VB 语言要好。

关键词:遗传算法;最优化问题;最优解;求解步骤;运算效率

中图分类号:TP312 **文献标识码**:A **文章编号**:1009-3044(2011)19-4654-02

Solving Optimization Problems Based on Genetic Algorithm

SONG Jie-peng[1,2]

(1. China University of Mining and Techonlogy, Xuzhou 221116, China; 2. Xuzhou Normal University, Xuzhou 221116, China)

Abstract: Genetic algorithm is a kind of effective method to solve optimization problems, and in solving complex global optimization problem, it has been successfully used. The basic steps of genetic algorithm are summarized, and through a real example, the detailed operation steps of genetic algorithm are described, which includes initial chromosomes generation, chromosomes evaluation, selection, crossover and mutation. VC and VB programming languages are applied for realizing the above steps, and the results show that VC is better than VB in operation efficiency optimality.

Key words: genetic algorithm; optimization problem; optimal solution; solving steps; operation efficiency

遗传算法是一种通过模拟自然进化过程搜索最优解的方法[1]。遗传算法本身并不要求对优化问题的性质作一些深入的数学分析,从而对那些不太熟悉数学理论和算法的使用者来说,无疑是方便的[2]。本文结合实例探讨了遗传算法求解最优化问题的具体步骤,并同时应用 VC 和 VB 进行编程,对其运算结果进行比较。

1 遗传算法的求解过程

生物遗传物质的主要的载体称为染色体,在遗传算法中,染色体通常是一串数据(或数组),它用来作为优化问题的解的代码,但它本身并不一定是解[3]。应用遗传算法求解一般要经过这样的几个过程[4]:首先,随机产生一定数目的初始染色体,由这些染色体组成一个种群,其中,种群中染色体的数目被称为种群的大小或规模。然后,用评价函数来评价得到的每个染色体的优劣,即染色体对环境的适应程度(称为适应度),用来作为以后遗传操作的依据。接着,进行选择过程,选择过程的目的是为了从当前种群中选出优良的染色体,使它们成为新一代的染色体,判断染色体是否优良的标准就是它们各自的适应度,也就是染色体的适应度越高,它被选择的机会就越多。通过选择过程,产生新的种群。再对这个新的种群进行一定概率的交叉操作,这个操作是遗传算法中的主要操作之一,类似于生物的杂交。然后进行变异操作,以挖掘种群中不同个体的多样性,克服其有可能陷入局部解的问题[5]。这样,经过上边的各种运算后产生的新的染色体称为后代。然后再对产生的后代重复进行选择、交叉和变异这几个操作,在迭代到设定的次数后,把生成的最好的染色体拿来作为优化问题的最优解。

2 用遗传算法求解最优化问题实例

应用遗传算法求解极大化问题:

$$\begin{cases} \max \sqrt{x_1}+\sqrt{x_2}+\sqrt{x_3} \\ s.t. \\ \quad x_1^{\,2}+x_2^{\,2}+x_3^{\,2}\le 1 \\ \quad x_1,x_2,x_3\ge 0 \end{cases}$$

2.1 初始产生 pop_size 个染色体

用染色体 $V=(x_1, x_2, x_3)$来作为解的代码。解的可行性由下面的检验函数检验:如果$(x_1<0 \parallel x_2<0 \parallel x_3<0)$ 则不可行;如果 $x_1^2+x2^2+x_3^2>1$ 则不可行。可知此模型的可行集包含于下列超几何体中

$$X=\{(x_1,x_2,x_3 \mid 0\le x_1\le 1,0\le x_2\le 1,0\le x_3\le 1)\}$$

这样,可以从这个超几何体中抽取初始染色体 $x_1=u(0,1)$, $x_2=u(0,1)$, $x_3=u(0,1)$

收稿日期:2011-05-25

作者简介:宋杰鹏(1977-),男,山东省莱阳市人,讲师,工程硕士在读,主要研究方向为计算机应用。

其中 u(0,1)表示服从区间[0,1]上的均匀分布的随机数。如果生成的染色体不可行,则拒绝接受,再重新生成一个新的染色体,如果可行,则接受它作为种群的一名成员。经过有限次抽样后,得到 30 个(设定种群的规模为 30,即 pop_size=30)可行的染色体。

2.2 计算每个染色体的评价函数值

计算产生的 30 个染色体的 $\sqrt{x_1}+\sqrt{x_2}+\sqrt{x_3}$ 的函数值,并按照由高到低的顺序排序,记函数值最大的染色体编号为 V_1,其余的顺次为 $V_2, V_3, \cdots, V_{30}$。

定义基于序的评价函数为[6] $$eval(V_i)=a(1-a)^{i-1}, \quad i=1,2,\cdots,30$$

其中,a 为给定的一个介于 0 到 1 之间的数据,例如 a=0.05,当 i=1 时,意味着染色体是最好的,i=30 时是最差的。

2.3 选择过程

选择过程可以通过如下四步给出[7]:

步骤 1 对每个染色体 V_i,计算累积概率 q_i:

$$\begin{cases} q_0=0 \\ q_i=\sum_{j=1}^{i} eval(V_j), \quad i=1,2,\cdots,30 \end{cases}$$

步骤 2 从区间$(0,q_{30}]$中产生一个随机数 r。

步骤 3 若 $q_{i-1}<r<q_i$,则选择第 i 个染色体 $V_i(1\leqslant i\leqslant 30)$。

步骤 4 重复步骤 2 和步骤 3 共30 次,得到 30 个复制的染色体。

2.4 交叉操作

定义参数 P_c 作为交叉操作的概率,从 i=1 到 30 重复执行如下步骤:产生一个[0,1]之间的随机数 r,若 $r<P_c$,那么选 V_i 作为父代中的一个染色体。 对于生成的父代中的任意两个染色体 V_m 和 V_n 进行交叉操作

$$X=c\cdot V_m+(1-c)\cdot V_n, \quad Y=(1-c)\cdot V_m+c\cdot V_n$$

c 为在(0,1)区间中产生的一个随机数,如果产生的两个新的后代 X 和 Y 均可行,则用它们代替父代,否则再一次产生随机数 c,再进行染色体的交叉,直到得到的 X 和 Y 均可行或循环到给定的次数后结束。

2.5 变异操作

定义参数 P_m 作为遗传系统中的变异概率,从 i=1 到 30 重复执行如下步骤:产生一个[0,1]之间的随机数 r,若 $r<P_m$,那么选择 V_i 这个染色体作为变异操作的父代。

用 $V=(x_1,x_2,\cdots,x_n)$表示上面的过程中生成的父代,再按照以下方法对生成的父代进行变异[8]。设 M 为在初始化的过程中任意定义的一个较大的数,在 n 维欧氏空间中任意产生一个 d 作为变异的方向,计算染色体 V+M·d 的值并将其代入约束条件中,如果判断后是不可行的,那么将 M 调整为 0 和 M 之间的任意一个随机数,再重新计算新染色体 V+M·d 的值,然后再判断它的可行性,重复上述过程直到得到的新染色体通过检验为止。若在设定的迭代次数内未能得到新的可行解,那么将 M 赋值为 0。上述操作结束后,用 X=V+M·d 这个新得到的染色体替换原先的染色体 V。重复进行选择、交叉、变异操作,直到给定的迭代次数为止。

2.6 结果

将上边的步骤用 C 语言写成程序,并在 VC++6.0 下运行,当设定交叉概率为 0.3,变异概率为 0.2,迭代 1000 次时,求得的最好解是 (0.5767,0.5768,0.5786),函数值是 2.2795。当设定交叉概率为 0.3,变异概率为 0.05,迭代 3000 次时,求得的最好解为(0.5817,0.5732,0.5771),函数值是 2.2795。由此,可以看出,最优解并不唯一。

为了验证结果是否为最好,又将上述算法在 VB6.0 下编程运行,同样设定交叉概率为 0.3,变异概率为 0.2,迭代次数为 1000,经过很长时间的运行后,显示的结果为(0.577,0.507,0.635),函数值为 2.269。而且在用 VB 运行时,若在用生成随机数 Rnd()函数前加上初始化种子数语句 Randomize,则每次运行的结果都有一点差别。所以在编写涉及大量计算的程序时,还是选用执行效率高且结果更优的 C 语言为好。

3 结论

本文给出了遗传算法的基本原理与步骤。将其应用于最优化问题,结合实例给出了详尽求解过程,并同时采用 VC 和 VB 编程。结果表明,在应用遗传算法求解最优化问题时,选用 C 语言编程求解,结果更加接近最优解,其求解效率也比 VB 语言要高。

参考文献:

[1] 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.
[2] 唐焕文,秦学志.实用最优化方法[M].大连:大连理工大学出版社,2004.
[3] 王凌.智能优化算法及其应用[M].北京:清华大学出版社,2001.
[4] 黄少荣.遗传算法及其应用[J].电脑知识与技术,2008,4(7):1874-1876.
[5] 岳嵚,冯珊.遗传算法的计算性能的统计分析[J].计算机学报,2009,32(12):2389-2392.
[6] 刘宝碇,赵瑞清,王纲.不确定规划及应用[M].北京:清华大学出版社,2003.
[7] 周丽,张智顺.遗传算法求解函数极值的应用[J].电脑知识与技术,2007,4(21): 802-803.
[8] 黄竞伟,朱福喜,康立山.计算机智能[M].北京:科学出版社,2010.