

用 MATLAB 实现遗传算法程序 *

刘国华, 包 宏, 李文超
(北京科技大学, 北京 100083)

摘 要: 简要阐述了遗传算法的基本原理, 探讨了在 MATLAB 环境中实现遗传算法各算子的编程方法, 并以一个简单的实例说明所编程序在函数全局寻优中的应用。

关键词: 遗传算法(GA); MATLAB

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1001-3695(2001)08-0080-03

A Genetic Algorithm in MATLAB

LIU Guo-hua, BAO Hong, LI Wen-chao
(Beijing University of Science & Technology, Beijing 100083, China)

Abstract: The principle of genetic algorithm has been presented and its realization in MATLAB has been discussed. A function optimization problem has been given to demonstrate the global optimization functionality of the MATLAB program.

Key words: Genetic algorithm(GA); MATLAB

1 遗传算法概述

遗传算法(Genetic Algorithm, GA)是借鉴生物界自然选择和群体进化机制形成的一种全局寻优算法。与传统的优化算法相比, 遗传算法具有如下优点^[1]: 1) 不是从单个点, 而是从多个点构成的群体开始搜索; 2) 在搜索最优解过程中, 只需要由目标函数值转换得来的适应值信息, 而不需要导数等其它辅助信息; 3) 搜索过程不易陷入局部最优点。目前, 该算法已渗透到许多领域, 并成为解决各领域复杂问题的有力工具^[2]。

在遗传算法中, 将问题空间中的决策变量通过一定编码方法表示成遗传空间的一个个体, 它是一个基因型串结构数据; 同时, 将目标函数值转换成适应值, 它用来评价个体的优劣, 并作为遗传操作的依据。遗传操作包括三个算子: 选择、交叉和变异。选择用来实施适者生存的原则, 即把当前群体中的个体按与适应值成比例的概率复制到新的群体中, 构成交配池(当前代与下一代之间的中间群体)。选择算子的作用效果是提高了群体的平均适应值。由于选择算子没有产生新个体, 所以群体中最好个体的适应值不会因选择操作而有所改进。交叉算子可以产生新的个体, 它首先使从交配池中的个体随机配对, 然后将两两配对的个体按某种方式相互交换部分基因。变异是对个体的某一个或某一些基因值按某一较小概率进行改变。从产生新个体的能力方面来说, 交叉算子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; 而变异算子只是产生新个体的辅助方法, 但也必不可少, 因为它决定了遗传算法的局部搜索能力。交叉和变异相配合, 共同完成对搜索空间的全局和局部搜索。

收稿日期: 2000-12-26

基金项目: 国家自然科学基金资助项目(59874006; 59872002)

遗传算法的基本步骤^[3]如下:

- 1) 在一定编码方案下, 随机产生一个初始种群;
- 2) 用相应的解码方法, 将编码后的个体转换成问题空间的决策变量, 并求得个体的适应值;
- 3) 按照个体适应值的大小, 从种群中选出适应值较大的一些个体构成交配池;
- 4) 由交叉和变异这两个遗传算子对交配池中的个体进行操作, 并形成新一代的种群;
- 5) 反复执行步骤 2~4, 直至满足收敛判据为止。

使用遗传算法需要决定的运行参数有: 编码串长度、种群大小、交叉和变异概率^[4]。编码串长度由优化问题所要求的求解精度决定。种群大小表示种群中所含个体的数量, 种群较小时, 可提高遗传算法的运算速度, 但却降低了群体的多样性, 可能找不出最优解; 种群较大时, 又会增加计算量, 使遗传算法的运行效率降低。一般取种群数目为 20~100。交叉概率控制着交叉操作的频率, 由于交叉操作是遗传算法中产生新个体的主要方法, 所以交叉概率通常应取较大值; 但若过大的话, 又可能破坏群体的优良模式。一般取 0.4~0.99。变异概率也是影响新个体产生的一个因素, 变异概率小, 产生新个体少; 变异概率太大, 又会使遗传算法变成随机搜索。一般取变异概率为 0.0001~0.1。遗传算法常采用的收敛判据有: 规定遗传代数; 连续几次得到的最优个体的适应值没有变化或变化很小等^[5]。

2 用 MATLAB 实现遗传算法

MATLAB 是 Matwork 公司的产品, 是一个功能强大的数学软件, 其优秀的数值计算能力使其在工业界和学术界的使用率都非常高。MATLAB 还十分便于使用, 它以直观、简洁并符合人们思维习惯的代码给用户提供了一个非常友好的开发环境^[6]。利用 MATLAB 处理矩阵运算的强大功能来编写遗传算法程序有着巨大的优势。

2.1 编码

遗传算法不对优化问题的实际决策变量进行操作,所以应用遗传算法首要的问题是通过编码将决策变量表示成串结构数据。本文中我们采用最常用的二进制编码方案,即用二进制数构成的符号串来表示一个个体,用下面的 encoding 函数来实现编码并产生初始种群:

```
function [bin_gen, bits] = encoding(min_var, max_var, scale_var, popsize)
bits = ceil(log2((max_var-min_var)/scale_var));
bin_gen = randint(popsize, sum(bits));
```

在上面的代码中,首先根据各决策变量的下界(min_var)、上界(max_var)及其搜索精度 scale_var 来确定表示各决策变量的二进制串的长度 bits,然后随机产生一个种群大小为 popsize 的初始种群 bin_gen。编码后的实际搜索精度为 $scale_dec = (max_var - min_var) / (2^{bits} - 1)$,该精度会在解码时用到。

2.2 解码

编码后的个体构成的种群 bin_gen 必须经过解码,以转换成原问题空间的决策变量构成的种群 var_gen。方能计算相应的适应值。我们用下面的代码实现。

```
function [var_gen, fitness] = decoding(funname, bin_gen, bits, min_var, max_var)
num_var = length(bits);
popsize = size(bin_gen, 1);
scale_dec = (max_var - min_var) / (2^bits - 1);
bits = cumsum(bits);
bits = [0 bits];
for i = 1:num_var
    bin_var[i] = bin_gen(:, bits(i) + 1:bits(i + 1));
    var[i] = sum(ones(popsize, 1) * 2.^((size(bin_var{i}, 2) - 1) - 1:0)) * bin_var[i] / 2.^bits(i) * scale_dec(i) + min_var(i);
end
var_gen = [var; 1, :];
for i = 1:popsize
    fitness(i) = eval(funname, var_gen(i, :));
end
```

解码函数的关键在于先由二进制数求得对应的十进制数 D,并根据下式求得实际决策变量值 X:

$$X = D \times scale_dec + min_var$$

2.3 选择

选择过程是利用解码后求得的各个体适应值大小,淘汰一些较差的个体而选出一些比较优良的个体,以进行下一步的交叉和变异操作。选择算子的程序如下:

```
function [evo_gen, best_indiv, max_fitness] = selection(old_gen, fitness)
popsize = length(fitness);
[max_fitness, index1] = max(fitness); [min_fitness, index2] = min(fitness);
best_indiv = old_gen(index1, :);
index = [1; popsize]; index(index1) = 0; index(index2) = 0;
index = nonzeros(index);
evo_gen = old_gen(index, :);
evo_fitness = fitness(index, :);
evo_popsize = popsize - 2;
ps = evo_fitness / sum(evo_fitness);
pcum = cumsum(ps);
r = rand(1, evo_popsize);
selected = sum(pcum * ones(1, evo_popsize) < r);
```

```
popsize, 1) * r) + 1;
evo_gen = evo_gen(selected, :);
```

在该算子中,采用了最优保存策略和比例选择法相结合的思路,即首先找出当前群体中适应值最高和最低的个体,将最佳个体 best_indiv 保留并用其替换掉最差个体。为保证当前最佳个体不被交叉、变异操作所破坏,允许其不参与交叉和变异而直接进入下一代。然后将剩下的个体 evo_gen 按比例选择法进行操作。所谓比例选择法,也叫赌轮算法^[4],是指个体被选中的概率与该个体的适应值大小成正比。将这两种方法相结合的目的是:在遗传操作中,不仅能不断提高群体的平均适应值,而且能保证最佳个体的适应值不减小。

2.4 交叉

下面采用单点交叉的方法来实现交叉算子,即按选择概率 PC 在两两配对的个体编码串 cpairs 中随机设置一个交叉点 cpoints,然后在该点相互交换两个配对个体的部分基因,从而形成两个新的个体。交叉算子的程序如下:

```
function new_gen = crossover(old_gen, pc)
[nouse, mating] = sort(rand(size(old_gen, 1), 1));
mat_gen = old_gen(mating, :);
pairs = size(mat_gen, 1) / 2;
bits = size(mat_gen, 2);
cpairs = rand(pairs, 1) < pc;
cpoints = randint(pairs, 1, [1, bits]);
cpoints = cpoints.' * cpoints;
for i = 1:pairs
    new_gen([2*i-1 2*i], :) = [mat_gen([2*i-1 2*i], :);
    cpoints(i) mat_gen([2*i 2*i-1], cpoints(i) + 1:bits)];
end
```

2.5 变异

对于二进制的基因串而言,变异操作就是按照变异概率 pm 随机选择变异点 mpoints,在变异点处将其位取反即可。变异算子的实现过程如下:

```
function new_gen = mutation(old_gen, pm)
mpoints = find(rand(size(old_gen)) < pm);
new_gen = old_gen;
new_gen(mpoints) = 1 - old_gen(mpoints);
```

3 应用实例

上述程序已经考虑了多参数编码问题,可以用于搜索多变量函数的最优解。为简单起见,下面仅以一个单变量函数为例,来验证所编遗传算法程序的全局寻优能力。设函数为: $y = \cos(5x) - \sin(3x) + 10$, $x \in [1, 7]$, 函数特性如图 1 所示。

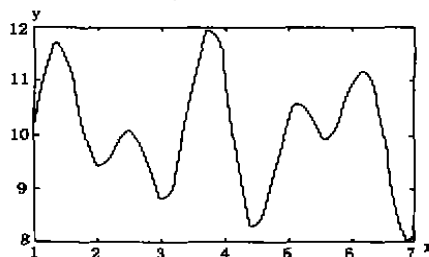


图1 函数特性示意图

取种群大小 popsize = 20, 搜索精度 scale_var = 0.0001, 交叉概率 pc = 0.6, 变异概率 pm = 0.1。图 2 和图 3 是

某一次运算遗传 20 代后最佳个体的适应值增长情况和最佳个体的变化情况。

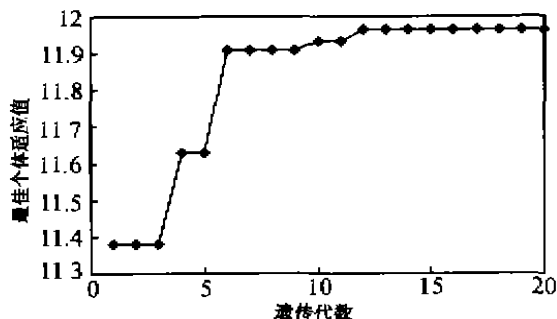


图 2 最佳个体适应值的增长情况

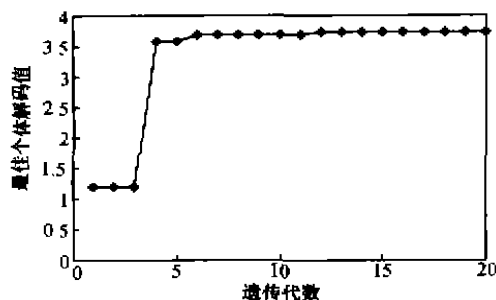


图 3 最佳个体的变化情况

由于采用了最优保存策略,所以在图 2 中未看到最佳个体适应值减少的现象。由图 3 可见:在前三代

种群中适应值最大的个体解码后的值为 1.1944,落在函数的一个局部极值处。但是搜索并没有在此处停滞,很快就跃到了另一个更大的极值点 3.7 附近。在该点附近搜索多次后,我们发现连续几次得到的最优个体的适应值变化很小,可以认为找到全局最大值,全局最大值为 3.7414,最大值为 11.9638。

4 结束语

我们用 MATLAB 编写了遗传算法程序,并给出完整代码,程序在 MATLAB 5.3 中调试通过。最后,通过一个实例说明其在函数优化中的应用。

参考文献:

- [1] 刘勇,康立山,陈毓屏.非数值并行算法(第二册)——遗传算法[M].北京:科学出版社,1997.
- [2] 席裕庚,柴天佑,恽为民.遗传算法综述[J].控制理论与应用,1996,13(6):697-708.
- [3] 韩颖祥,文福拴.模拟进化优化方法及其应用——遗传算法[J].计算机科学,1995,22(2):47-56.
- [4] 周明,孙树栋.遗传算法原理及其应用[M].北京:国防工业出版社,1999.
- [5] 梁吉业.遗传算法应用中的一些共性问题研究[J].计算机应用研究,1999,16(7):20-21.
- [6] 张宜华.精通 MATLAB 5[M].北京:清华大学出版社,1999.

作者简介:

刘国华(1974-),男,博士生,研究方向为人工智能及其在计算机辅助材料设计中的应用;包宏,副教授;李文超,教授,博导。

(上接第 79 页)

```
END IF
Ids_datastore = CREATE datastore
Ids_datastore.dataobject = "d_emplst"
Ids_datastore.setTransobject(SQLCA)
RetrieveData 函数实现检索功能。其函数原形为:
RetrieveData(reference blob able_data) return Long
Long ll_rr
Ids_datastore.Retrieve()
ll_rr = ids_datastore.GetFullState(ablb_data)
Transrv.Setcomplete()
Return ll_rr
UpdateDate 函数实现代码如下:
Long ll_rr
If ids_datastore.SetChanges(ablb_data) = 1 then
ll_rr = ids_datastore.Update()
end if
if ll_rr = 1 then
transrv.Setcomplete()
else
transrv.SetAbort()
end if
return ll_rr
```

在组件关闭事件中必须回收存储过程资源并断开连接。

```
DESTROY ids_datastore
DISCONNECT USING SQLCA;
```

5 结束语

本例主要讲述的是基于组件事务服务器中数据库

的操作。由于组件三层/多层模式的结构特点对数据的可保护性和维护性在性能上有很大的提高,但在应用程序中也必要加上状态标志和缓冲,否则将使得其性能适得其反。在应用中当大量客户端同时访问数据库的情况下,可以通过组件事务服务器定义一个数据库连接缓冲池,各客户端可以共享这个缓冲池,使得数据库服务器所需管理和维护的连接数大大减少,从而提高系统的性能。

参考文献:

- [1] 恽为民. PowerBuilder 7.0 实例应用进阶[M].北京:机械工业出版社,222-253.
- [2] PowerBuilder 6.0 用户参考手册[M].北京:希望电子出版社,344-500.
- [3] 郭迅华. Sybase 组件事务服务器 Jaguar CTS[M].北京:电子工业出版社,200-245.
- [4] McFall C. An Object Infrastructure for Internet Middleware IBM on Component Broker[J]. IEEE Internet Computing, 1998,42-50.
- [5] Anne I. Jaguar CTS, Jaguar CTS white paper [M]. Emery ville Sybase cop 1999.

作者简介:

杨书凡(1975-),女,湖南大学计算机系硕士生,助教,研究方向为分布式计算、网络技术。