

基于 Matlab 遗传算法工具箱的优化计算实现

陈秋莲¹, 王成栋²

(1. 广西大学 计算机与电子信息学院 广西 南宁 530004; 2. 南宁利隆消防检测有限公司 广西 南宁 530003)

摘要: 遗传算法是一种基于生物自然选择与遗传机理的随机搜索与优化方法。遗传算法求解复杂优化问题的巨大潜力及其在工业工程领域的成功应用, 而受到了广泛关注。Matlab 的遗传算法工具箱具有使用简单方便, 计算可靠, 快捷高效和图形结果可视化等特点, 具有广阔的应用前景。主要介绍了基于 Matlab 遗传算法工具箱, 深入分析其编码方式、初始群体、选择、交叉、变异操作及适应度函数的实现方法。最后给出应用实例, 验证了 Matlab 遗传算法优化工具的有效性。

关键词: 遗传算法; Matlab; 遗传算法; 工具箱; 演化过程

中图分类号: T P311

文献标识码: B

文章编号: 1004 - 373X(2007) 02 - 124 - 03

Optimization of Genetic Algorithm Toolbox Based on Matlab

CHEN Qiulian¹, WANG Chengdong²

(1. College of Computer and Information Engineering, Guangxi University, Nanning, 530004, China;

2. Nanning Lilong Fire Control Detection Co., Ltd., Nanning, 530003, China)

Abstract: Genetic algorithm is a random search and optimization method based on natural select and genetic mechanism of the living beings. It is successfully used in solving the complex optimization and the industrial engineering problem, and has attracted a lot of attention. GA toolbox is convenient to be used and has the characteristics such as as reliable calculation, high efficiency and visualized graphic results, so it has a broad application prospect. This paper introduces the GA toolbox. Expounds on the Encoding, Initiation, Selection, Xover, Mutation operation and Fitness function. Finally simulation shows the effectiveness and practicability of the GAOT in Matlab.

Keywords genetic; algorithm; Matlab; GAOT; toolbox; evolution process

1 引言

遗传算法(Genetic Algorithm), 是一种基于进化论优胜劣汰、自然选择、适者生存和物种遗传思想的随机优化搜索算法。由于其具有思想简单、易于实现、应用效果明显等优点, 他在解决工程问题中存在的巨大潜力。目前各个领域的学者和专家都对他进行了深入的研究, 并用各种语言, 如 C, C++ 等来实现这个算法。但由于这些语言必须要编写遗传程序才能进行计算, 使得不熟悉编程的使用者望而却步。作为使用者希望找到一个现成的程序直接进行运算, 而 Matlab 的遗传算法工具箱正好满足这一要求。

Matlab 是一种用于面向科学与工程计算的高效率高级语言, Matlab 遗传算法工具箱(GAOT) 提供了对各种优化问题的一个完整的解决方案。其简洁的函数表达、多种优化算法的任意选择、对算法参数的自由设置, 可使用户方便灵活地使用优化函数。遗传算法工具箱(GAOT) 的构造合理, 扩展方便, 具有简单、易学、易用、易修改的特

点, 可实现二进制编码和实数值编码等的模拟进化计算。他的遗传操作非常灵活, 为应用和研究遗传算法提供稳定可靠、结构灵活、可扩展的开发平台。

2 遗传算法的基本思想

遗传算法模拟物种从低级到高级的演化过程, 即从初始群体出发, 采用优胜劣汰, 适者生存的自然法则选择个体, 通过交叉、变异来产生下一代群体, 逐代演化, 直到产生满足条件的个体为止。遗传算法的演化过程可用如下形式描述^[1]:

$$GA = (P(0), N, L, s, g, p, f, t) \tag{1}$$

这里 $P(0) = (p_1(0), p_2(0), \dots, p_n(0)) \in I^N$, 表示初始群体;

$I = B^N = \{0, 1\}^L$ 表示长度为 L 的二进制码位串空间;

N 表示群体中含有个体的个数;

L 表示二进制串的长度;

$s: I^N \rightarrow I^N$ 表示选择策略。

g : 遗传算子, 通常包括有选择(复制)算子 $Q_r: I \rightarrow I$;

交叉算子 $Q_c: I \times I \rightarrow I \times I$ 和变异算子 $Q_m: I \rightarrow I$;

P : 遗传算子的操作概率, 他包括选择(繁殖)概率 P_r 、

交叉概率 P_c 和变异概率 P_m ;

$f: I \rightarrow R^+$ 表示适应函数;

$t: I^N \rightarrow \{0, 1\}$ 是终止准则。

遗传算法以适应度函数(或目标函数)为依据,通过对群体中的个体施加遗传操作,实现群体内个体结构重组的迭代处理过程,基本的遗传算法框图如图 1 所示。

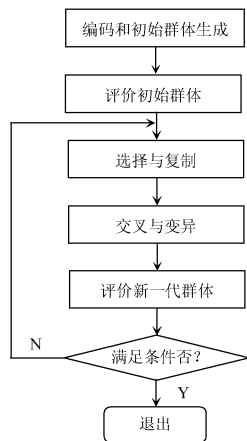


图 1 标准遗传算法流程图

3 Matlab 遗传算法工具箱的使用

3.1 遗传算法工具箱的结构

Matlab 遗传算法工具箱(GAOT)实现遗传算法的步骤和其他编程语言实现遗传算法的流程一样,也需要解决编码方式、初始群体的建立、选择算子、交叉算子、变异算子、评价函数及终止标准等问题。故而 Matlab 的遗传算法工具箱一般由 4 个关键部分组成:编码机制、适应度函数、遗传算子、运行参数。在遗传算法程序中,染色体的数据结构、对应的交叉和变异算子、选择算子与具体问题无关,编写的遗传算法工具包提供了对上述要素的支持。而对与具体问题相关的适应度函数和终止条件,遗传算法工具包给出了对应函数的编写格式和调用方式。

GAOT 通过编写 GA 基本操作的 M 文件和主程序的 M 文件来实现和改进 GA。他首先对优化问题进行分析,建立优化数学模型,定义适应函数(对于约束优化问题,还需要确定约束条件,给出约束函数)。然后利用文件编辑器将这些函数写入到能返回函数值的 M 文件中,这样就把函数表达式写入 Matlab 系统中。最后在命令窗口执行 M 文件,就可以得到优化解。

3.2 遗传算法工具箱的基本 M 文件

(1) 初始化

初始化的程序存在 initialize.m 文件中,初始化函数用法如下:

Function[pop] = initialize(num, bounds, evalFN, evalops, options)

主要输入参数有: Num: 群体数目; Bounds: 变量上限

和下限组成的矩阵; evalFN: 评价函数的文件名,通常是 .m 文件; Options: 一个向量[epsilon float\binary prec], 这里 epsilon 表示两代之间的差距; 第二个参数取 0 表示采用二进制编码,取 1 表示采用实数本身; prec 表示变量的精度。默认值为[1e-6 1](可选择的); 输出参数 pop 中包含有初始群体的变量值和适应值。

(2) 选择操作

选择或复制操作是决定哪些个体可以进入下一代。工具包中提供了赌轮盘选择法(roulette), 锦标赛选择法(tournSelect)和几何规划排序选择(normGeomSelect)。如采用几何规划排序选择可用: function[newPop] = normGeomSelect(oldPop, options)

其中 options 为选择概率; oldpop 为旧群体。

(3) 交叉操作

交叉过程是选取 2 个个体作为父代 parent1, parent2, 产生出 2 个新的子代个体 child1 和 child2。GAOT 中提供了: arithXover, heuristicXover, simpleXover 三种交叉方式。如算术交叉(arithXover)的实现可通过调用 function[c1, c2] = arithXover(p1, p2, bounds, ops) 得到。该函数会产生一个随机数 a; 然后通过父代 p1 和 p2: $p1 * a + p2 * (1 - a)$, $p1 * (1 - a) + p2 * a$ 得到子代个体 c1 和 c2。

(4) 变异操作

GAOT 提供了几个变异操作的 M 文件: binaryMutation.m, boundaryMutation.m, multiNonUnifMutation.m, nonUnifMutation.m, unifMutation.m, 在各个不同的 M 文件中包含了不同的变异操作, 如 nonUnifMutation.m 文件中包含了非统一变异函数: function[parent] = nonUnifMutate(parent, bounds, ops); 其输入参数有 parent: 父代对应的变量值和适应值, bounds: 变量上限和下限组成的矩阵, ops: nonUnifMutate 的参数[当前代...]

(5) 评价函数

* eval.m 是遗传算法与具体应用问题的惟一接口, 是进行自然选择的惟一依据。改变种群内部结构的遗传操作均要通过评价函数加以控制。许多类型的评价函数都可以用于遗传算法, 但他应至少要满足一个条件: 函数值部分有序地分布。在* eval.m 应包含有这样一行:

function[val, sol] = gaFxeval(sol, options), 其中: sol 为群体中各个个体的值, val 为个体的适应值。

如求函数 $F(x) = x * \sin(10 * \pi * x) + 1$, 在 $x \in [-1, 2]$ 范围内的极大值。 $F(x)$ 为 x 的评价函数。那么编写评价函数 M 文件的代码应有以下 3 行:

```
function[ val, sol] = gaFxeval(sol, options)
% sol: 群体中各个个体的值, val: 个体的适应值
x= sol(:, 1); % 获取各变量的值
val= x * sin(10 * pi * x) + 1; % 计算出各变量的评价值
```

(6) 终止函数

在 GAOT 中提供了这 2 个终止文件: maxGenTerm.

m(到某一最大的代数停止)和 optMaxGenTerm.m(偏差作为终止条件)。如采用最大代数终止,可以在 maxGenTerm.m 文件中添加:

```
function [ done] = maxGenTerm(ops, bPop, endPop)
% ops 一个参数向量
% bPop- 最优解组成的一个数组
% endPop - 当前代的个体和对应的适应值
currentGen = ops(1);
maxGen= ops(2);
done= currentGen >= maxGen;
```

(7) 主程序函数

主程序函数的作用是调用相应的遗传操作函数,完成遗传优化,主程序函数是 ga.m,其用法如下:

```
function[ x, endPop, bPop, traceInfo] = ga( bounds, evalFN,
evalops, startPop, opts, ..., termFN, termops, selectFN, selec-
tops, xoverFNs, xoverops, mutFNs, mutops)。
```

其中主函数的等号左边是输出参数,右边是输入参数,主要数值参数和 GA 的基本操作,而基本操作实际上就是 M 文件。

bounds: 变量上限和下限组成的矩阵;

evalFN: 评价函数的文件名,通常是.m文件;

startPop: 调用 initialize.m 文件得到的初始染色体(可选择的);

opts: 一个向量[epsilon prob- ops display],这里 epsilon 表示两代之间的差距;prob- ops 取 0 表示采用二进制编码,取 1 表示采用实数本身;display 取 1 表示运行中显示当前染色体和最好结果,取 0 则不显示。

termFN: 终止函数名称,默认值为['maxGenTerm'];

selectFN: 选择操作的.m 的文件名,默认值为['norm - GeomSelect'];

xoverFNs: 交叉操作的.m 文件,实数编码默认值为['arithXoverheuristicXoversimplxXover'],二进制编码默认值为['simpleXover'];

xoverops: 交叉函数的输入参数矩阵,实数编码默认值为[20; 23; 20],二进制编码默认值为[0.6];

mutFNs: 变异操作的.m 文件;

mutops: 变异操作的.m 文件的输入矩阵,实数编码默认值为[40; 61003; 41003; 400],二进制编码默认值为[0.05](可选)。

4 算例仿真

用基于 Matlab 的 GA 工具箱 GAOT 来进行函数优化,只需改写相应函数的 M 文件的内容即可。如求多峰值函数 $f(x) = x + 10 * \sin(5 * x) + 7 * \cos(4 * x)$ 在区间 $[0, 9]$ 内的极大值。取初始种群大小等于 10,最大代数为 100 代,直接取 $f(x) = x + 10 * \sin(5 * x) + 7 * \cos(4 * x)$ 作为适应函数。采用几何规划排序选择、算术交叉、多点非

统一变异,则要修改的 M 文件主要是适应函数文件。

4.1 适应函数文件编写

用文本编辑窗来编写适应函数文件 gaDemolEval.m。在文件中加入以下 3 行:

```
function [ sol,f]= gaDemolEval( sol,options)
x= sol(:,1);% 获取当前个体的值
f= x+ 10* sin(5* x)+ 7* cos(4* x); % 计算个体的适应值
```

4.2 调用主程序函数

在命令窗口中调用初始化程序和主程序 ga.m,进行遗传优化。代码如下:

```
fplot('x+ 10* sin(5* x)+ 7* cos(4* x)',[0 9])
initPop= initializega(10,[0 9], 'gademoleeval');
% 生成初始种群,大小为 10
hold on
plot(initPop(:,1),initPop(:,2), 'b*')
% 调用遗传函数,一次遗传迭代
[x endPop]= ga([0 9], 'gademoleeval', [], initPop, [1e-5...1],
' maxGenTerm', 1, 'normGeomSelect', [0.08], ['arithXover', [2...0],
' multiNonUnifMutation', [2 1 3] ]);
hold on
plot(endPop(:,1),endPop(:,2), 'bo')
% 100 次遗传迭代
[x endPopbpop trace]= ga([0 9], 'gademoleeval', [], initPop,
[1e-6 1...1], ' maxGenTerm', 100, 'normGeomSelect', [0.08],
['arithXover', [2...0], ' multiNonUnifMutation', [2 100 3] ]);
plot(x(:,1),x(:,2), 'y*')
figure(2)
plot(trace(:,1),trace(:,2), 'g-')
hold on
plot(trace(:,1),trace(:,3), 'r-')
xlabel('Generation');
ylabel('fitness');
legend('解的变化', '种群平均值的变化', 4)
```

仿真实验结果如图 1 所示, x 轴表示代数, y 轴表示适应值。图中有 2 条曲线分别是最优适应值曲线和平均适应值曲线。由图 1 可知,随迭代数的增加,种群的平均值呈现了曲折上升的趋势。最后,解向最优解方向靠拢,种群平均值也不断趋于平稳。这说明 Matlab 遗传算法工具箱在多峰函数寻优中十分有效,并且随着计算代数的增加,计算精度得到改善。

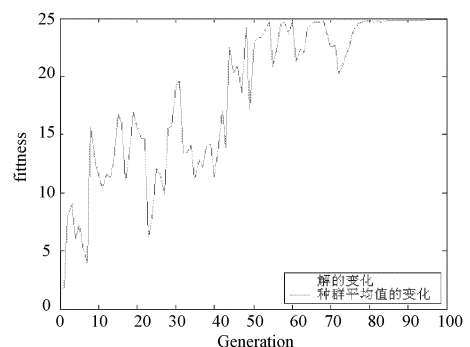


图 2 适应值变化曲线

(下转第 129 页)

每一个 Action 都在 Action 映射列表中定义, Action 映射决定 Action 与其他 Web 组件之间的关联关系。以添加代表队信息为例, 简单叙述一下 Action 映射的过程: 在 struts-config.xml 中配置 R_teamAction 类, 把 parameter 的属性设置为 method, 代码如下:

```
< action path= "/add_team" type= "com.boxmis.struts.R_teamAction" name= "add_teamForm" parameter= "method" scope= "request" input= "/R_team/add_team.jsp">
    < forward name= "toshowteam" path= "/view_team.do? method= View"/>
</action>
```

Struts 接收到“/add_team.do? method= Insert”请求后转到 R_teamAction 中去, 通过表单 Bean add_teamForm 将表单中数据装入模型 R_team 中去, 通过参数 method 调用 R_teamAction 中的 Insert() 方法, 最终将数据插入数据库。

图 3 给出了代表队信息模块中 Action 组件的映射图。

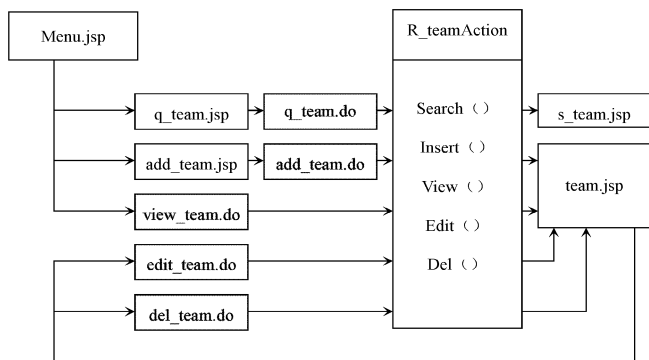


图 3 代表队信息模块中 Action 组件映射图

作者简介 余剑波 男, 1981 年出生, 硕士研究生。研究方向为计算机网络与管理信息系统。

(上接第 126 页)

5 结 语

基于 Matlab 的遗传算法工具箱提供了一个标准的、可扩展的、简单的算法。利用 Matlab 的强大的矩阵运算能力, 使用者可以避免维护遗传算法种群和染色体的数据结构繁重编程工作, 节省大量编程时间和精力, 将精力集中在遗传算法的改进和具体问题的应用中去。

作者简介 陈秋莲 女, 1974 年出生, 讲师, 硕士。主要从事遗传优化设计、数据库方面的研究。

王成栋 男, 1974 年出生, 工程师。主要从事 Matlab 应用、电气工程方面的研究。

图 4 给出了代表队信息模块的运行效果图。

5 结 语

本系统利用当前主流的 Struts 框架, 实现了以浏览器为客户端, 网络服务器、数据库服务器的三层体系结构, 满足拳击比赛流动性大的要求, 同时系统又充分体现了拳击比赛的特点。目前本系统正处在试运行阶段, 随着系统的不断完善, 必将对我国的拳击比赛项目的发展起到积极的作用。

序号	代表队名称	国家名称	国家代码	比赛日期	比赛时间	比赛地点	比赛结果
00000	中国国家队男子拳击队	中国	CHN	2004-11-14	2004-11-14	2004-11-14	胜利
00001	中国队	中国	CHN	2004-11-25	2004-11-25	2004-11-25	胜利
00002	中国队	中国	CHN	2004-11-25	2004-11-25	2004-11-25	胜利
00003	中国队	中国	CHN	2004-11-25	2004-11-25	2004-11-25	胜利
00004	中国队	中国	CHN	2004-11-25	2004-11-25	2004-11-25	胜利
00005	中国队	中国	CHN	2004-11-25	2004-11-25	2004-11-25	胜利
00006	中国队	中国	CHN	2004-11-25	2004-11-25	2004-11-25	胜利

图 4 代表队信息模块的运行效果图

参 考 文 献

- [1] 黄庄庄. 基于 C/S 结构的拳击比赛信息系统总体设计[D]. 北京: 北京工业大学, 2005.
- [2] Vivek Chopra. JSP 程序设计[M]. 北京: 人民邮电出版社, 2006.
- [3] 张桂元, 贾燕枫. Struts 开发入门与项目实践[M]. 北京: 人民邮电出版社, 2005.

参 考 文 献

- [1] 潘正君, 康立山, 陈毓屏. 演化计算[M]. 北京: 清华大学出版社, 1994.
- [2] 尹泽明, 丁春利. 精通 Matlab 6[M]. 北京: 清华大学出版社, 2002.
- [3] 陈永春. Matlab M 语言高级编程[M]. 北京: 清华大学出版社, 2004.

《现代电子技术》(半月刊) 2006 年电子版开始征订
电话: 029-85393376 邮发代号: 52-126 定价: 158 元/套