

# 基于 MATLAB 的遗传算法优化工具箱的应用

章红兵

同济大学工业工程系, 上海 (201804)

E-mail: [roky951@163.com](mailto:roky951@163.com)

**摘 要:** 本文对遗传算法 (GA) 和基于 MATLAB 的遗传算法优化工具箱 (GAOT) 作了简要的介绍, 详细分析了优化工具函数, 具体探讨了 GAOT 在参数优化和非线性规划中的应用, 实例证明了 GA 在最优化工程中的可行性以及 GAOT 在解决基于 GA 的优化问题上的有效性和实用性。

**关键词:** 遗传算法, Matlab, 优化, 遗传算法工具箱

## 1. 引言

最优化在运筹学和管理科学中起着核心作用。最优化通常是极大或极小某个多变量的函数并满足一些等式和 (或) 不等式约束。最优化技术对社会的影响日益增加, 应用的种类和数量快速增加, 丝毫没有减缓的趋势。然而, 许多工业工程设计问题性质十分复杂, 用传统的优化方法很难解决。近年来, GA 作为一种全新的优化方法, 以其巨大的潜力受到人们的普遍关注<sup>[1][2]</sup>。GA 几乎渗透到从工程到社会科学的诸多领域, 解决这些问题必须要编制 GA 程序进行计算, 作为使用者希望找一个现成的程序, 而基于 MATLAB 的 GAOT 正好满足要求<sup>[3]</sup>。本文将结合 GAOT 讨论在非线性规划和参数优化中的具体应用。

## 2. GA与Matlab语言

遗传算法 (Genetic Algorithm, 缩写为 GA), 最先是由 John Holland 教授于 1975 年提出的。GA 是一种借鉴生物界自然选择和自然遗传机制的高度并行、随机、自适应性强的搜索算法。GA 是将问题的求解表示成“染色体”, 将它们置于问题的“环境”中, 根据适者生存的原则, 从中选择出适应环境的“染色体”进行复制, 即再生 (reproduction, selection), 通过交叉 (crossover)、变异 (mutation) 两种基因操作产生出新一代更适合环境的“染色体”群, 这样一代代不断改进, 最后收敛到一个最适合环境的个体上 (当然也有其他的收敛准则), 求得问题的最佳解<sup>[4]</sup>。

MATLAB (Matrix Laboratory) 是美国 MathWorks 公司推出的一种集数值计算、符号运算、可视化建模、仿真和图形处理等多种功能于一体的图形化语言<sup>[5]</sup>。以 MATLAB 为平台所编制的遗传算法优化工具箱 (Genetic Algorithm Optimization Toolbox, 缩写为 GAOT) 可以利用 MATLAB 强大的仿真功能, 进行 GA 的各种优化运算<sup>[6]</sup>。

## 3. GAOT的函数及其功能

MATLAB 以 m 文件的形式贮存了大量函数, 具有移植性且人机交互性能好, 在解决具体优化问题时, 可通过修改或编制相应的 m 文件, 再灵活地搭配各种函数, 编制相应的 MATLAB 程序就能解决问题。目前 Internet 上有许多的 GAOT 提供下载, 但良莠不齐, 本文所选用的 GAOT: <http://www.ie.ncsu.edu/mirage/GAToolBox/gaot/gaotv5.zip>。其包括许多实用的函数, 各种算子函数, 各种类型的选择方式, 交叉、变异方式。这些函数按照功能可以分成以下几类:

### 3.1 主界面函数

主程序 ga.m 提供了 GAOT 与外部的接口。它的函数格式如下：  
[x endPop bPop traceInfo]=ga(bounds,evalFN,evalOps,startPop,opts,termFN,termOps,  
selectFn,selectOps,xOverFNs,xOverOps,mutFNs,mutOps)  
输出参数及其定义如表 1 所示。输入参数及其定义如表 2 所示。

表 1 ga.m 的输出参数  
Tab.1 Output parameter of ga.m

输出参数	定 义
x	求得的最好的解，包括染色体和适应度
endPop	最后一代染色体（可选择的）
bPop	最好染色体的轨迹（可选择的）
traceInfo	每一代染色体中最好的个体和平均适应度（可选择的）

表 2 ga.m 的输入参数  
Tab.2 Input parameter of ga.m

输入参数	定 义
Bounds	变量上下界的矩阵，矩阵的行数确定变量的个数
evalFN	适应度函数
evalOps	传递给适应度函数的参数，默认值为(NULL)
startPop	初始染色体
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 表示二进制编码,取 1 表示浮点数编码；display 取 0 表示运行中不输出，取 1 表示运行中显示输出。默认值为[1e-6 1 0]
termFN	终止函数的名称，默认值为[maxGenTerm]
termOps	传递给终止函数的参数，默认值为[100]
selectFn	选择函数的名称，默认值为[normGeomSelect]
selectOps	传递给选择函数的参数，默认值为[0.08]
xOverFNs	交叉函数名称表，二进制编码默认值为[simpleXover]，浮点数编码默认值为[arithXover heuristicXover simpleXover]
xOverOps	传递给交叉函数参数表，二进制编码默认值为[0.6],浮点数编码默认值为[2 0;2 3;2 0]
mutFNs	变异函数名称表，二进制编码默认值为[binaryMutation]，浮点数编码默认值为[boundaryMutation multiNonUnifMutation nonUnifMutation unifMutation]
mutOps	传递给变异函数参数表，二进制编码默认值为[0.05],浮点数编码默认值为[4 0;6 100 3;4 100 3;4 0 0]

3.2 核心函数及其它函数如表3所示

表 3 GAOT 核心函数及其它函数  
Tab.3 Functions of GAOT

核心函数及其它函数		定义及功能
初始化函数	initializega.m	二进制格式和浮点数格式的初始化函数
	initializeoga.m	有序数据的初始化函数
选择函数	roulette.m	常用的轮盘赌法
	normGeomSelect.m	基于归一化的优先选择法
	tourmSelect.m	竞争选择法
演化函数	交 simpleXover.m	二进制格式货浮点数格式的交叉函数

	叉	cyclicXover.m linerXover.m linerorderXover.m	有序数据的交叉函数，可以将演化函数组合使用
	变异	boundaryMutation.m nonUnifMutation.m	浮点数格式的变异函数
终止函数		maxGenTerm.m optMaxGenTerm	主程序 ga.m 用来判断是否满足终止条件
二进制表示函数		calcbits.m	用来计算遗传算法满足精度要求时，染色体所需要的二进制位数
		f2b.m b2f.m	用来完成二进制数和浮点数之间的相互转换

## 4. GAOT在非线形规划中的应用

### 4.1 非线性规划简述<sup>[7]</sup>

非线性规划是在存在等式或不等式约束的前提下最优化某个目标函数的问题。由于许多实际问题不能成功地表达为线性规划模型，因此非线性规划对于工程、数学和运筹学的各个领域都是及其重要的工具。一般非线性规划可描述如下：

$$\begin{aligned}
 &\text{目标函数} && \text{Max } f(x) \\
 &\text{不等式约束} && g_i(x) \leq 0, i=1, \dots, m \\
 &\text{等式约束} && h_i(x)=0, i=m+1, \dots, n \\
 &&& x \in \Omega
 \end{aligned}$$

GA对染色体做遗传操作通常会产生不可行的后代，因而运用GA解非线性规划的核心是如何满足约束的问题。对于约束条件，工程中常用的方法是惩罚策略，通过惩罚不可行解，将有约束问题转化为无约束问题。构造带有惩罚项的适值函数一般有两种。一种是采用加法形式： $\text{val}(x)=f(x)+p(x)$

对于极大化问题，则取：

$$\begin{cases} p(x) = 0, \text{若} x \text{可行} \\ p(x) > 0, \text{其他} \end{cases}$$

对于极小化问题，则取：

$$\begin{cases} p(x) = 0, \text{若} x \text{可行} \\ p(x) < 0, \text{其他} \end{cases}$$

另一种是采用乘法形式： $\text{Val}(x)=f(x)*p(x)$

此时，对于极大化问题，则取：

$$\begin{cases} p(x) = 1, \text{若} x \text{可行} \\ 0 \leq p(x) < 1, \text{其他} \end{cases}$$

对于极小化问题，则取：

$$\begin{cases} p(x) = 0, \text{若} x \text{可行} \\ p(x) > 1, \text{其他} \end{cases}$$

其中： $x$ 代表染色体； $f(x)$ 是问题的目标函数； $p(x)$ 是惩罚项。

## 4.2 实例分析

$$\begin{aligned} \text{Min } f(x) &= (x_1 - 2)^2 + (x_2 - 1)^2 \\ g_1(x) &= x_1 - 2x_2 + 1 \geq 0 \\ \text{s.t. } g_2(x) &= \frac{x_1^2}{4} - x_2^2 + 1 \geq 0 \end{aligned}$$

取加法形式的适值函数:

$$\begin{aligned} \text{Val}(x) &= f(x) + p(x) \\ P(x) &= \begin{cases} 0, & \text{若 } x \text{ 可行} \\ \sum_{i=1}^m r_i g_i(x), & \text{其他} \end{cases} \end{aligned}$$

其中  $r_i$  是约束  $i$  的可变惩罚系数。

## 4.3 采用 GAOT 编程计算

①编制目标函数适值文件 nonlinearMin.m 如下:

```
function[sol,eval]=nonlinearMin(sol,options)
x1=sol(1); x2=sol(2);
r1=0.1; r2=0.8;
g1=x1-2*x2+1; g2=x1.^2/4-x2.^2+1; %约束条件
if(g1>=0)&(g2>=0) %加惩罚项的适值
    eval=(x1-2).^2+(x2-1).^2;
else
    eval=(x1-2).^2+(x2-1).^2+r1*g1+r2*g2;
end
eval=-eval;
```

②编制调用主程序 nonlinear\_ga.m, 其程序如下:

```
bounds=ones(2,1)*[-1 1]; %设置参数边界
initPop=initializega(100,bounds,'nonlinearMin');
[p,endPop,bestSols,trace]=ga(bounds,'nonlinearMin'); %遗传算法优化
p
plot(trace(:,1),trace(:,3),'b-') %性能跟踪
hold on
plot(trace(:,1),trace(:,2),'r-')
xlabel('Generation','fontsize',14); ylabel('Fitness','fontsize',14);
legend('解的变化','种群平均值的变化');
```

③结果输出

```
p=[1 1]
eval(p)=1, g1(p)=0, g2(p)=0.25
```

显然最优解满足约束条件, 图 1 是 GA 寻优性能跟踪图。

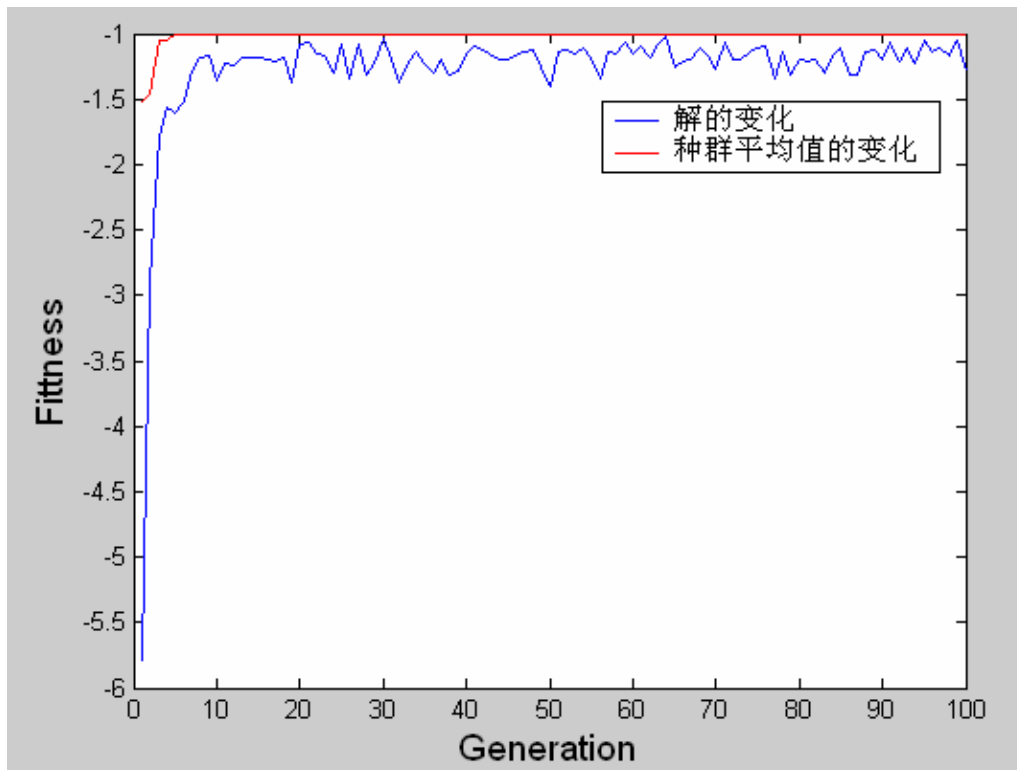


图 1 GA 寻优性能跟踪图  
Fig1 Track figure of searching optimization

## 5. GAOT在参数优化中的应用

De Jong 函数是一个连续的、凸起的单峰函数，表达式为  $f(x) = \sum_{i=1}^n x_i^2$ ，其二维函数图形

如图 2 所示。

下面用 GA 解 De Jong 函数的参数优化问题，求解： $\min f(x)$ ， $-512 \leq x_i \leq 512$

采用 GAOT 的步骤如下：

①编制 De Jong 函数文件 DeJong.m 如下：

```
function[eval]=DeJong(sol)
numv=size(sol,2);
x=sol(1:numv);
eval=sum(x.^2);
```

②编制目标函数适值文件 DeJongMin.m 如下：

```
function[sol,eval]=DeJongMin(sol,options)
numv=size(sol,2)-1;
x=sol(1:numv);
eval=DeJong(x);
eval=-eval;
```

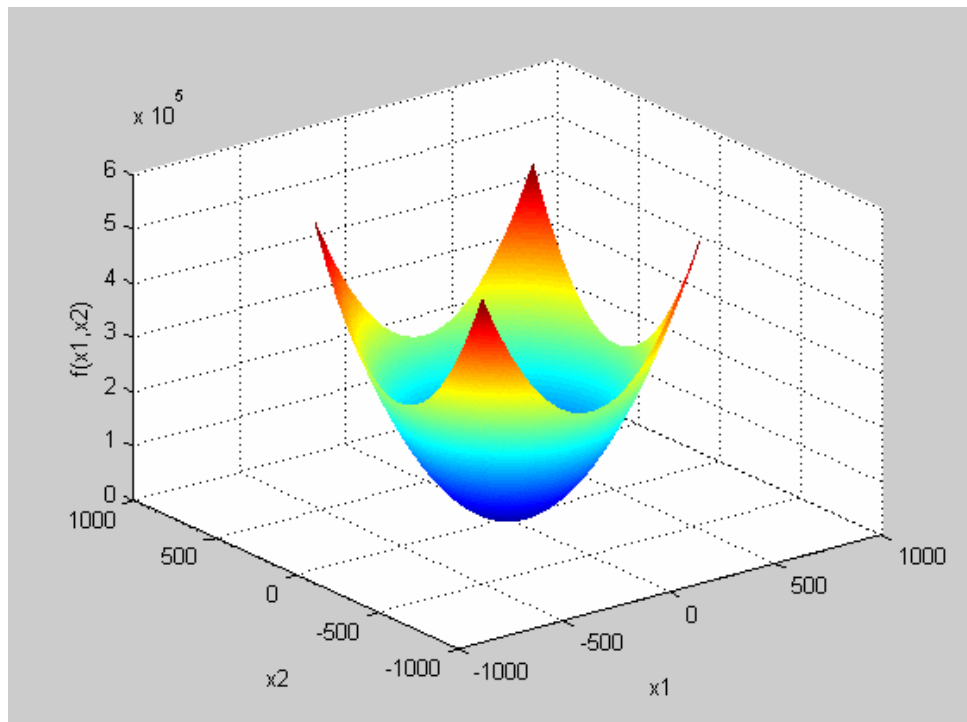


图 2 De Jong 函数图  
Fig2 Function figure of De Jong

③编程调用主程序 DeJong\_ga.m, 其程序如下:

```
bounds=ones(3,1)*[-512 512]; %设置参数边界
[p,endPop,bestSols,trace]=ga(bounds,'DeJongMin'); %遗传算法优化
p
plot(trace(:,1),trace(:,3),'b-') %性能跟踪
hold on
plot(trace(:,1),trace(:,2),'r-')
xlabel('Generation','fontsize',14); ylabel('Fitness','fontsize',14);
legend('解的变化','种群平均值的变化')
```

④结果输出

```
p=1.0e-003 * [0.0112    0.1450    0.1183]
eval(p)=0.0000
```

理论上最优解为:  $p=[0 \ 0 \ 0]$ , 极小值为 0。显然 GA 有效地解决了 De Jong 函数的极小化问题。图 3 是 De Jong 函数的 GA 的寻优性能图。

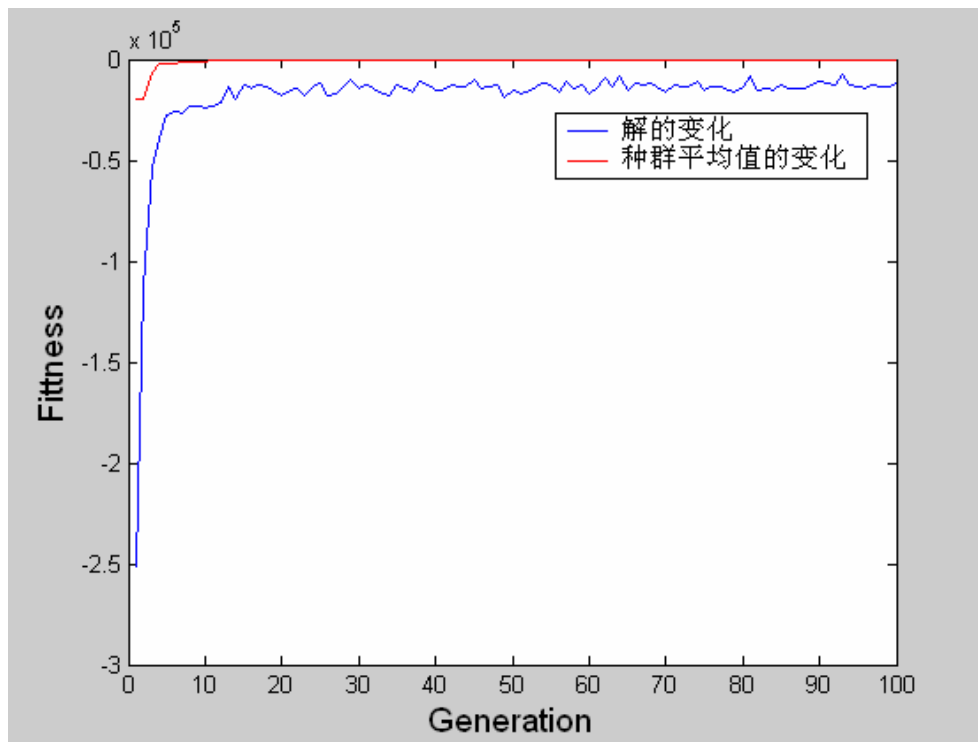


图 3 GA 的寻优性能图

Fig3 Track figure of searching optimization

## 6. 总结

GA 作为一种求解复杂系统优化问题的通用算法,正在得到广泛的应用。基于 MATLAB 的 GAOT 提供了一个标准的、可扩展的、简单的算法。使用者可以节省大量编程时间和精力,且使用灵活、方便,易于学习和掌握。只要对其相关模块作适当修改,即可解决许多实际问题。实例表明,GAOT 对于实际工程优化问题具有较好的应用前景。

## 参考文献

- [1] 玄光男,程润伟著.遗传算法与工程优化[M].北京:清华大学出版社,2004.
- [2] 王春水,肖学柱,陈汉明.遗传算法的应用举例[J].计算机仿真.第22卷,第6期.
- [3] 刘勇,刘宝坤,李光泉.基于MATLAB平台的遗传算法工具包[J].天津大学学报.第34卷,第4期.
- [4] 周明,孙树栋.遗传算法原理及应用[M].北京:国防工业出版社,1991.
- [5] 张志涌.精通Matlab6.5版[M].北京:北京航空航天大学出版社,2003.
- [6] A. J. Chipperfield, P. J. Fleming. The MATLAB genetic algorithm toolbox[C]. IEE colloquium on Applied Control Techniques Using MATLAB, Digest No. 1995/014, 26/01/95.
- [7] 飞思科技产品研发中心编著.MATLAB6.5 辅助优化计算与设计[M].北京:电子工业出版社,2003.

## Application of Genetic Algorithm Optimization Toolbox Based on Matlab

Zhang Hongbing

Dept. of Industrial Engineering, Tongji University, Shanghai 201804, China

### Abstract

In this paper, genetic algorithm (GA) and genetic algorithm optimization toolbox (GAOT) based on MATLAB have been introduced simply. The optimization toolbox function is analyzed particularly. It has discussed specifically the application of GAOT in the parameter optimization and the nonlinear programming. The two examples have proven GA in the optimized project feasibility as well as GAOT in the solution based on GA validity and usability.

**Keywords:** genetic algorithm (GA); Matlab; optimization; genetic algorithm toolbox (GAT)

**作者简介:** 章红兵, 男, 1982 年生, 硕士研究生, 主要研究方向为先进生产系统组织与优化。