遗传算法原理及其MATLAB实现

1 遗传算法原理

详见参考书籍。

2 遗传算法基本操作

详见参考书籍。

3 遗传算法的 MATLAB 实现

3.1 编码和种群生成

初始种群的生成函数

```
function [pop] = initializega(num, bounds, eevalFN, eevalOps, options)
```

参数说明

pop	生成的初始种群
num	种群中的个体数目
bounds	代表变量的上下界的矩阵
eevalFN	适应度函数
eevalOps	传递给适应度函数的参数
options	选择编码形式(浮点编码或是二进制编码)[precision, code] **precision**–变量进行二进制编码时指定的精度 **code**为1时选择浮点编码，否则为二进制编码

3.2 终止

```
function done = terminateFunction(options, bestPop, pop)
```

表 1: Matlab Implemented Termination Functions

Name	File	Options
Terminate at Specified Generation	maxGenTerm.m	final generation
Terminate at Optimal or max gen	maxGenOptTerm.m	final generation, optimal value, epsilon

3.3 交叉

```
function [c1, c2] = crossover(p1, p2, bounds, ops)
```

表 2: Matlab Implemented Crossover Functions

Name	File	Options
Arithmetic Crossover	arithXover.m	none
Heuristic Crossover	heuristicXover.m	number of retries
Simple Crossover	simpleXover.m	none

3.4 变异

```
function c1 = Mutation(p1, bounds, genInfo, Ops)
```

表 3: Matlab Implemented Mutation Functions

Name	File	Options
Boundary Mutation	boundary.m	none
Multi-Non-Uniform Mutation	multiNonUnifMut.m	max num of generations
Non-Uniform Mutation	nonUnifMut.m	max num of generations
Uniform Mutation	unifMut.m	none

3.5 选择

```
function [newPop] = Select(oldPop, options)
```

表 4: Matlab Implemented Selection Functions

Name	File	Options
Roulette Wheel	roulette.m	None
Normalized Geometric Select	normGeomSelect.m	Probability of Selection Best
Tournament	tourn.m	Number of individuals in each tournament

3.6 遗传算法

遗传算法函数

```
function [x, endPop, bPop, traceInfo] = ga(bounds, ...
    eevalFN, eevalOps, ...
    startPop, opts, ...
    termFN, termOps, ...
    selectFN, selectOps, ...
    xOverFNs, xOverOps, ...
    mutFNs, mutOps)
```

参数说明

x	求得的最优解
endPop	最终得到的种群
bPop	最优的个体及所处代构成的矩阵
traceInfo	最优种群的一个搜索轨迹
bounds	代表变量上下界的矩阵
evalFN	适应度函数，用m-文件
evalOps	传递给适应度函数的参数，默认为[]
startPop	初始种群
opts	[epsilon prob disp]，默认为[1e-6 1 0] opts(1:2)等同于initializega的options参数 第三个参数控制是否输出
termFN	终止函数的名称，默认为['maxGenTerm']
termOps	传递个终止函数的参数，默认为[100]
selectFN	选择函数的名称，默认为['normGeomSelect']
selectOps	传递个选择函数的参数，默认为[0.08]
xOverFNs	交叉函数名称表，以空格分开 浮点编码下默认为['arithXover heuristicXover simpleXover'] 二进编码下默认为['simpleXover']
xOverOps	传递给交叉函数的参数表 浮点编码下默认为[2 0;2 3;2 0] 二进编码下默认为[0.6]
mutFNs	变异函数表 浮点编码下默认为 ['boundaryMutation multiNonUnifMutation nonUnifMutation unifMutation'] 二进编码下默认为['binaryMutation']
mutOps	传递给交叉函数的参数表 浮点编码下默认为[4 0 0;6 100 3;4 100 3;4 0 0] 二进编码下默认为[0.05]

4 实例一

4.1 问题

利用遗传算法求下面函数的最大值

$$f(x) = x + 10\sin(5x) + 7\cos(4x), \quad x \in [0, 9]$$

4.2 分析

选择二进制编，种群中的个体数目为10，采用浮点编码。

可以用如下MATLAB命令直接绘出f(x)的图形看f(x)的最大值大致范围。

```
fplot('x+10*sin(5*x)+7*cos(4*x)',[0,9])
```

4.3 程序

(1) 编写目标及适应度函数 fitness.m，将文件 fitness.m存放于工作目录。

```
function [sol, eval] = fitness(sol, options)
x = sol(1);
eval = x + 10 * sin(5 * x) + 7 * cos(4 * x);
```

(2) 用遗传算法求最大值

```
%生成初始种群, 大小为10
initPop = initializega(10, [0 9], 'fitness');

%25次遗传迭代
[x, endPop, bPop, trace] = ga([0 9],...
    'fitness',[],...
    initPop, [1e-6 1 1],...
    'maxGenTerm', 25,...
    'normGeomSelect', [0.08],...
    ['arithXover'], [2],...
    'nonUnifMutation', [2 25 3]);
```

(3) 查看迭代过程

```
figure(1)
plot(endPop(:, 1), endPop(:, 2),'y*')

figure(2)
plot(trace(:, 1), trace(:, 2), 'r-')
xlabel('Generation');
ylabel('Fittness');
legend('解的变化','种群平均值的变化');
```

4.4 结果

```
x = 7.8562 24.8553
```

表明当x 为7.8562 时，函数f(x) 取最大值24.8553.

可以使用MATLAB优化函数来验证。

5 实例二

5.1 问题

求下面二元函数

$$f(x_1,x_2) = -20\exp\Big(-0.2\sqrt{0.5(x_1^2+x_2^2)}\Big) - \exp\Big(0.5(\cos(2\pi x_1)+\cos(2\pi x_2))\Big) + 22.71282$$

在区域$[-5,5]\times[-5,5]$ 上的最小值.

5.2 分析

种群大小10，最大代数1000，变异率0.1，交叉率0.3。

5.3 程序

(1) 定义函数f.m

```
function eval = f(sol)
numv = size(sol, 2);
x    = sol(1 : numv);
eval = -20 * exp( - 0.2 * sqrt(sum(x .^ 2)/ numv))) ...
       - exp(sum(cos(2 * pi * x)) / numv) + 22.71282;
```

(2) 定义适应度函数fitness.m

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

(3) 用遗传算法求解

```
bounds = ones(2, 1) * [-5 5];

%生成初始种群，大小为10
initPop = initializega(10, bounds, 'fitness');

[p, endPop, bestSols] = ga(bounds, 'fitness', [], ...
    initPop, [1e-6 0 0],...
    'maxGenTerm', 1000,...
    'normGeomSelect', [0.08],...
    ['simpleXover'], [0.3],...
    'binaryMutation', [0.1]);
```

5.4 运行及结果

将前面两个函数存储为m文件并放在工作目录下，运行结果为

```
p = 0.0000 -0.0000 0.0055
```