## 一 问题设置和结果

### 1 问题

Fitness function **适应值函数**是最小化目标函数。可以使用形式为@objfun 函数句柄指定这个函数，其中 objfun.m 是返回一个量的函数文件。

Number of variables **变量个数**是适应值函数中独立变量的个数.

### 2 约束

Linear inequalities **线性不等式**是有矩阵 A 和向量 b 定义的 Ax≤b。

Linear equalities **线性等式**是有矩阵 Aeq 和向量 beq 定义的 Aeq*x=beq。

Bounds **边界**是变量的上界（Upper）和下界（Lower）。

Nonlinear constraint function **非线性约束函数**定义了非线性约束，可把这个函数定义为一个无名函数或者形式为@nonlcon 的函数句柄，其中 nonlcon.m 是返回向量 c 和 ceq 的函数文件。非线性等式的形式是 ceq=0，非线性不等式的形式是 c≤0。

### 3 运行求解器并查看结果

要运行求解器，单击 start 按钮。当算法终止时，状态和结果格中显示该算法终止的原因。

## 二 选项

### 1. Population 种群

- Population options **种群选项**设置遗传算法的种群的选项。
- Population type **种群类型**设置适应值函数的输入类型和约束：

（1）Double vector 条目为 Double 型的向量

（2） Bit string **字节串** 即由 0 和 1 组成的二进制串。对于创建函数和变异函数，选择“均匀”或“自定义”；对于交叉函数，选择“分散”、“单点”、“双点”或“自定义”；不可以使用混合函数或非线性约束函数。

（3）custom **自定义** 对于交叉函数和变异函数，选择“自定义”；对于创建函数，选择自定义或提供一个初始种群；不可以使用混合函数或非线性约束函数。

- Population size **种群规模**指定了每一代有多少个体。如果你把种群规模设置为一个比 1 大的向量，算法会创建多个子种群。向量的每个条目指定一个子种群的规模。
- Creation function **创建函数**指定了创建初始种群的函数：

Use constraint dependent default **使用约束依赖**。如果没有约束，选择“均匀”；否则，选择“可行种群”。

（1）Uniform **均匀** 根据均匀分布创建随机初始种群；

（2）Feasible population **可行种群** 创建满足边界和线性约束的随机初始种群。

Custom **自定义** 可以提供自己的创建函数，必须产生在种群类型中定义的类型的数据。输入函数句柄@CreationFcn 可以定义。

- Initial population **初始种群** 需要为遗传算法指定一个初始种群。如果你不设置初始种群，算法用创建函数创建一个函数。可以设置比种群规模更少的个体；如果这样做，创建函数创建的是最好的。
- Initial score **初始值** 设置初始种群的适应度，如果你不设置初始适应度，算法使用适应度函数计算适应度。
- Initial range **初始范围** 指定了初始种群的向量条目的上下界。可以用一个 2 行和初始长度做列的矩阵指定初始范围，第一行是初始种群的向量条目的下界，第二行是上界。如果指定初始种群为 2 行 1 列的矩阵，两个标量扩大为长度为初始长度的常数向量。

## 2. Fitness scaling 适应值缩放

缩放函数把适应值函数返回的初始适应值转换为适应选择函数的范围的值。

- Rank **排序** 根据每一个个体的排名扩大或缩小原始的适应度。排序之后，个体的排名便是它的位置。最优个体的排名是 1，次优个体的排名为 2，以此类推。排序适应度缩放法消除了原始适应度的扩展效应。
- Proportional **按比例** 使每个个体的期望（比例）与其原始适应度值成比例。当原始适应度值不在一个“好的范围”中，该策略会有缺陷。
- Top 用最大适应度等同的扩大或缩小个体。如果选用了该策略，可以指定 Quantity，即产生后代的最优个体的数目。Quatity 必须是 1 到种群规模之间的整数，或介于 0 到 1 之间的表明占种群规模比例的分数。每一个个体都有相同的复制概率。其余的个体复制概率为 0。
- Shift linear 扩大或缩小初始适应度，使最优个体的期望值等于一个常数，定义此常数为“最大存活率”。
- Custom **自定义** 自己可以定义缩放函数。输入一个形式为@ScaleFcn 的句柄就可以定义。

### 3. Selection 选择

选择函数是基于适应值缩放函数缩放之后的值为下一代选择父代。

可以选择如下的选择函数：

- Stochastic uniform **随机均匀** 产生一条线，每一个父代个体对应于线上的一段，线段的长度所占的比例相应于父代个体的期望值所占的比例。该算法以相同的步长沿线移动，每一步确定一个父代，算法确定其“着陆”的部分为一个父代。该算法的第一步是确定一个比步长小的均匀随机数。
- Remainder 每一个个体的缩放之后的适应度的整数部分赋于父代，然后使用轮盘赌选择方法选择剩余的分数部分。
- Uniform **均匀** 依据均匀分布函数，根据父代的期望和数目随机的选择父代。该方法是无定向搜索。均匀选择不是一个特别有效的搜索策略，但是可以用来测试遗传算法。
- Shift linear 扩大或缩小初始适应度，使最优个体的期望值等于一个常数，定义此常数为“最大存活率”。
- Roulette **轮盘赌** 模拟一个轮盘，使每一个个体所占的面积与其期望成正比。然后产生一个随机数，随机数落在个体的积累面积之内就选择这个个体作为父代。
- Tournament **竞争** 随机的选择个体，这些被选择的个体的数目成为竞争规模，然后选择最优的个体使之成为父代。
- Custom **自定义** 自己可以定义选择函数。输入一个形式为@SelectFcn 的句柄就可以定义。

### 4. Reproduction 复制

复制决定遗传算法在新的一代产生多少后代。

- Elite count **精英数** 下一代最少存活的个体数目。精英数应为小于或等于种群规模的正整数。
- Cross fraction **交叉率** 参与交叉运算的个体所占比例。交叉率为 0 到 1 之间的一个分数。

### 5. mutation 变异

变异函数使个体产生微小变化，扩大遗传算法搜索范围。在 **Mutation function** 中指定变异方程，可从下选择策略：

- Use constraint dependent default **策略**：Gaussian 指无约束；否则选

adaptive feasible ，即自适应可行。

- Gaussian 策略：为每个个体向量提供一个随机数，这个随机数从以零为中心的高斯分布中选出，这个分布的标准差由参数 Scale（决定第一代的标准差）和 Shrink（随着每代的变化控制标准差的收缩）决定。如果 shrink 是 0 则标准差是连续的；如果是 1 则标准差线性递减到 0。

- Uniform 均匀策略：这是一个两步的过程。首先，算法选择个体向量的一段进行变异，每个个体变异的概率相等。然后，算法交换选中的部分。

- Adaptive feasible 自适应可行策略：随机产生适应上一代变异方向。变异长度根据变异方向选择，这样就会满足非线性约束。

- Custom 自定义：可以允许你写下自己的变异函数。格式为@MutateFcn，MutateFcn.m 文件的语法详见帮助。

### 6. Crossover 交叉

结合两个个体（父代）来形成一个新个体，用 Crossover function 来设置交叉函数，可以选择的策略：

- Scattered 分散策略：创造随机二进制向量。从第一个父代中选择向量中为“1”的位置，从第二个父代中选择向量中为“0”的位置，结合组成子代。例如：

p1 = [a b c d e f g h]

p2 = [1 2 3 4 5 6 7 8]

random crossover vector = [1 1 0 0 1 0 0 0]

child = [a b 3 4 e 6 7 8]

- Single point 单点交叉策略：选择一个 1 到变量数的随机整数 n，从第一个父代中选择一个向量，基因位置小于等于 n 片段，从第二个父代选择一个基因位置大于 n 的片段，连接这两个片段形成子代。例如：

p1 = [a b c d e f g h]

p2 = [1 2 3 4 5 6 7 8]

random crossover point = 3

child = [a b c 4 5 6 7]

- Two point 多点交叉策略：选择 1 到变量数间的两个整数 m 和 n。算法从第一个父代中选择基因位置小于等于 m 的片段，从第二个父代中选择基因位置为 m+1 到 n 的片段，从第一个父代中选择基因位置大于 n 的片段，算法连接

这些基因片段形成新基因。例如：

p1 = [a b c d e f g h]

p2 = [1 2 3 4 5 6 7 8]

random crossover points = 3,6

child = [a b c 4 5 6 g h]

- Intermediate 策略：以随机权重由父代产生子代。由参数 Ratio 控制：child1= parent1+…rand***Ratio***(parent2 - parent1)，Ratio 可以是标量或是变量个数长度的向量，如果 Ratio 是[0,1]间的数，子代位于由父代位置定义的空间相反的顶点。
- Heuristic 启发式策略：子代位于父代两点连成的直线，距离父代距离近适应值较好，距离父代远适应值较差。
- Arithmetic 算数策略：子代是父代的差，在父代所在的直线上。
- Custom 自定义策略：允许自定义交叉函数，以@CrossoverFcn 调用，具体语法见帮助。

### 7. Migration 迁移

是不同子代间个体的移动，如果种群规模（Population size）设定为长度大于 1 的向量则有算法产生。偶尔，子代中最优个体代替另一子代中的最差个体。可由以下三个参数控制迁移产生的频率：

- Direction 指定迁移发生的方向，如果设为 Forward，迁移在最后一代发生，就是第 n 个子代迁移到第 n+1 个子代；如果设为 Both，则第 n 个子代既迁移到第 n-1 个子代也迁移到第 n+1 个子代。迁移可能陷入局部解，也就是最后一个子代迁移到第一个子代，第一个迁移到最后一个，为了防止这种情况要指定一个子代的规模为 0。
- Fraction 控制子代间移动的个体数。移动个体数要小于较小规模子代的个体数。如果从一个个体数为 50 的子代移动到个体数位 100 的子代，Fraction 为 0.1，则要移动 0.1×50=5 个个体。从一个子代移动到另一个的个体是要复制过去，移动源的子代不变。
- Interval 控制移动间隔的代数。如果设置为 20，则每 20 代移动一次。

### 8. Algorithm settings 算法设置（针对非线性约束）

- 初始惩罚（Initial penal ty）算法的初始值，必须大于等于 1。
- 惩罚因子（Penalty factor）当问题不满足要求的精度或者不满足约束

时，就用惩罚因子加大惩罚参数。惩罚因子要大于 1。

### 9. Hybrid function 杂交函数

允许在遗传算法终止后，再指定一个最小化函数。可选择项为：

- 无
- fminsearch（无约束）
- patternsearch（有约束或无约束）
- fminunc（无约束）
- fmincon（有约束）

设置此选项的命令行为“optmset (psoptimset for *patternsearch*)”

### 10. stopping criteria 停止准则

- Generations 指定遗传算法的迭代次数。
- Time limit 指定遗传算法运行的最长时间，以秒为单位。
- Fitness limit 最优适应值小于等于 fitness limit 则停止算法。
- Stall generations 平均加权适应值函数小于 function tolerance。
- Stall time limit 在一定间隔时间秒数内（stall time limit 设定）适应值函数值没有提高，算法终止。
- Function tolerance 如果适应值函数在 stall generations 累计的变动小于 function tolerance，算法终止。
- Nonlinear constraint tolerance 超出非线性约束的最大终止容忍度。

### 11. plot functions 散点图绘制函数

可以用散点图标示遗传算法执行时的不同方面。每方面在屏幕上用一个单独的轴表示。用“stop”键停止运行。

- Plot interval 指定区域中更新的代数。
- Best fitness 根据迭代次数标示出每代的最优函数值。
- Best individual 标示出每代种群中，有最优适应值的个体的向量。
- Distance 标示出每代种群中个体的平均距离。
- Expectation 根据每代初始适应值标示出期望的子代的个数。
- Genealogy 即家谱，标示出个体的谱系。代数间的颜色线表示：红色表示变异子代；蓝色表示交叉子代；黑色表示优秀个体。
- Range 标示出每代最小、最大和平均适应值。
- Score diversity 标示出每代适应值的柱状图。

- Scores 表示出每代个体的适应值。
- Selection 绘制出父代的柱状图。可以显示父代中哪个个体对子代贡献最大。
- Stopping 标示出停止准则等级。
- Max constraint 标示出可以超出非线性约束的最大值。
- Custom 输入一个函数，形式为*@plotfun*，plotfun.m 文件句法是 function state = plotfun(options, state, flag)。

### 12. output function 输出函数

- History to new window 输出算法的迭代历史。
- Interval 设置迭代多少代个体输出一次结果。
- Custom 自定义设置输出函数，输入@outputfun，句法详见帮助。

### 13. display to command window 输出到命令窗口

Level of display 指定展示在命令窗口的信息量。可选项为：

- Off 不输出。
- Iterative 每次迭代都输出。
- Diagnose 每次迭代都输出，还包括一些问题诊断信息和改变缺省值的options。
- Final 只展示停止运行的原因。

### 14. user function evalution 用户函数评价

评价适应函数和约束函数。指定如何评价你提供的函数。

- In serial 意思是适应值函数和约束函数在种群中分开评价。
- Vectorized 意思是适应值函数和约束函数在函数调用时分开评价。
- In parallel 意思是适应值函数和约束函数评价在批处理时完成。必须先设置并行环境。