

# BAS-toolbox manual

2019/04/09

## 安装

1. 下载 BASmatlab 工具箱，并解压；
2. 运行 `install.m` 文件，完成安装。

第二步中，出现如下提示：

```
> Save path for future Matlab sessions?[y/n]
```

输入 `y`，则工具箱被加入 `matlab` 的路径及子路径，此后使用无需再次安装；反之，每次运行需要重新安装。

## 算法

在rBAS 文档中，记录了 `BAS` 及各种改进算法的原理，以及如何在 `R` 和 `matlab` 下使用对应的工具箱。当然，还会有一些调参经验。

## 使用

### 函数形式

目前工具箱提供了 `BAS` `BSAS` 以及 `BSO` 三种算法，都遵循下面的调用形式。

```
options = BASoptimset;  
fit1 = BASoptim(objective,constraint,lower,upper,init,options);  
fit2 = BSASoptim(objective,constraint,lower,upper,init,options);  
fit3 = BSOoptim(objective,constraint,lower,upper,init,options);
```

- `objective`; 目标函数的句柄；
- `constraint`; 约束函数的句柄；可以取空，即 `[]`
- `lower/upper`; 上下限；
- `init`; 初始值；可以为 `[]`
- `options`; 关于算法的各种参数设定；可以为 `[]`

### 案例 1

考虑简单的案例，如式 1 所示。

$$f(x) = \sum_{i=1}^n x_i^2, \quad n = 1, 2, 3, 4 \quad (1)$$

## 参数设置

通过 `BASoptimset` 函数，来得到一个关于算法参数的结构体。（所有的算法都是通过 `BASoptimset`）来提供参数设置的。

```
options = BASoptimset;
options
```

该结构体包含有算法的各种参数，如步长，迭代数，衰减系数等等：

```
options =

struct with fields:

    step0: 2.2204e-16 % 步长最小分辨率
    step1: 0.8000     % 步长初始值
    eta_step: 0.9500  % 步长衰减系数
    d0: 2.2204e-16   % 触须长度最小分辨率
    d1: 2            % 触须长度初始值
    eta_d: 0.9500    % 衰减系数
    n: 100           % 迭代次数
    seed: []         % 随机种子
    trace: 0         % 信息可视化
    steptol: 2.2204e-16 % 迭代终止的步长条件
    penalty: 100000   % 惩罚因子
    k: 2             %BSAS: 每回合天牛数/触须对数
    Pgreedy: 0.8000   %BSAS: 贪婪概率
    PstepUpdate: 0.8000 %BSAS: 更新步长概率（找不到更优解时）
    nflag: 3         %BSAS: 超过该回合数，强制更新步长等参数
    vmax: 5          %BSO: 粒子速度上限
    vmin: -5         %BSO: 粒子速度下限
    wstepmax: 0.9000  %BSO: 步长更新权重上限
    wstepmin: 0.4000  %BSO: 步长更新权重下限
    wvmax: 0.9000     %BSO: 速度更新权重上限
    wvmin: 0.4000     %BSO: 速度更新权重下限
    lambda: 0.4000    %BSO: 位置更新权重
    s: []            %BSO: 粒子数量
```

`options` 结构体，包含目前提供的算法所需的所有参数。

## 优化

案例函数在工具箱中提供，句柄为 `SquareSums`。由于没有约束，第二项参数为空。初始值和 `options` 也可以为空。

```
fit = BASoptim(@SquareSums,[],[-2,-2,-2,-2],[2,2,2,2]);  
fit.par  
fit.fitness
```

```
ans =
```

```
0.0001    -0.0041    0.0042    -0.0032
```

```
ans =
```

```
4.5252e-05
```

## 案例 2-压力容器

考虑更为复杂的混合整形规划案例，如式 2 所示。

$$\begin{aligned} \text{minimize } f(\mathbf{x}) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\ &\quad + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\ s.t. \quad g_1(\mathbf{x}) &= -x_1 + 0.0193x_3 \leq 0 \\ g_2(\mathbf{x}) &= -x_2 + 0.00954x_3 \leq 0 \\ g_3(\mathbf{x}) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\ g_4(\mathbf{x}) &= x_4 - 240 \leq 0 \\ x_1 &\in \{1, 2, 3, \dots, 99\} \times 0.0625 \\ x_2 &\in \{1, 2, 3, \dots, 99\} \times 0.0625 \\ x_3 &\in [10, 200] \\ x_4 &\in [10, 200] \end{aligned} \tag{2}$$

## 约束形式

问题可以写为如下的形式，我们通过结构体来包含目标函数和约束。

```
function f = Pressure_Vessel  
    %obj  
    f.obj = @obj;  
    %con
```

```

    f.con = @con;
end

function fobj = obj(x)
    x1 = floor(x(1)) * 0.0625;
    x2 = floor(x(2)) * 0.0625;
    x3 = x(3);
    x4 = x(4);
    fobj = 0.6224*x1*x3*x4 + 1.7781*x2*x3^2 + ...
        3.1611*x1^2*x4 + 19.84*x1^2*x3;
end

function fcon = con(x)
    x1 = floor(x(1)) * 0.0625;
    x2 = floor(x(2)) * 0.0625;
    x3 = x(3);
    x4 = x(4);
    fcon = [0.0193*x3 - x1, 0.00954*x3 - x2, ...
        750.0*1728.0 - pi*x3^2*x4 - 4/3*pi*x3^3];
end

```

具体函数见github或工具箱 /BAS/funcs/Pressure\_Vessel.m 文件。

## 参数设置与优化

参数设置如下：

```

PV = Pressure_Vessel;
objective = PV.obj;
constraint = PV.con;

options = BASoptimset;
options.k = 2;
options.step1 = 100;
options.d1 = 5;
options.n = 200;
options.seed = 5;

```

调用 BSAS 算法来进行优化，如下：

```
fit = BSASoptim(objective,constraint,[1,1,10,10],[100,100,200,200],[],options);  
fit.par  
fit.fitness
```

结果如下:

```
ans =  
  
    14.2773    7.5271   44.9014  144.7474  
  
ans =  
  
    6.1403e+03
```

## 后续工作

- 待完成: 目前还有部分的算法没有迁移到 matlab 工具箱。后续我会进行这部分地工作, 当然, 欢迎大家复现, 并 pull requests。
- 问题: 复现的过程中, 我可能会犯一些错误。尽管程序可能不会出错, 但是结果会不在预期范围之内。希望大家能够在issues下指出, 当然, 能给出可能的原因就更好了。
- 新的算法: 十分欢迎大家, 列出自己的文章和代码, 来丰富工具箱, 不管是 pull requests 或是在issues里面提。