

Yalmip使用学习

RavenX 关注

2018.01.23 21:44* 字数 925 阅读 2497 评论 0 喜欢 1

yalmip学习

0. yalmip简介

0.1 什么是yalmip

yalmip是由Lofberg开发的一种免费的优化求解工具，其最大特色在于集成许多外部的最优化求解器，形成一种统一的建模求解语言，提供了Matlab的调用API，减少学习者学习成本。

0.2 yalmip安装方式

这里以MATLAB的安装方式为例，在[官网](#)上下载最新包，将其解压至 `matlab` 的 `toolbox` 文件夹下（当然也放置在其他文件夹），打开matlab软件添加Path路径即可。最后键入 `which sdpvar` 命令，显示 `sdpvar` 路径则安装成功。

1.yalmip求解优化问题的四部曲

1.1 创建决策变量

yalmip一共有三种方式创建决策变量，分别为：

1. `sdpvar`-创建实数型决策变量
2. `intbar`-创建整数型决策变量
3. `binvar`-创建0/1型决策变量

不过值得注意的是，在创建 $n \times n$ 的决策变量时，yalmip默认是对称方阵，所以要创建非对称方阵时，需要这样写：

```
xxxvar(n,n,'full')
```

1.2 添加约束条件

比起matlab自带的各种优化函数所要写明的约束条件，yalmip的约束条件写起来是非常舒适直观的。

比如要写入 $0 \leq x_1 + x_2 + x_3 \leq 1$ 。

那么可以这样写：

```
% 创建决策变量
x = sdpvar(1,3);
% 添加约束条件
C = [0<=x(1)+x(2)+x(3)<=1];
```

是不是非常爽呢。这才是人类语言（和我初见python的感觉差不多）

1.3 参数配置

关于参数设置，我们大多数是用来设置求解器solver的，当然还有其它的选项，可以通过doc sdpsettings查看。

1.4 求解问题

最后就是求解问题了。

首先要明确求解目标 z ，yalmip默认是求解最小值问题，所以遇到求解最大值的问题，只需要在原问题的基础上添加一个负号即可。

求解调用格式：

```
optimize(target,constraints,options)
```

1.5 几个常用的其它指令

1. check：可以检查约束条件是否被满足（检查约束条件的余值）
2. value：可以查看变量或表达式的值
3. assign: 可以给变量赋值，这个命令调试时很重要

2. 举两个栗子

2.1 简单例子

简书的markdown没办法用latex，就直接贴图啦（百度随便找了个题目）

$$z = \max\left(\frac{x_1 + 2x_2}{2x_1 + x_2}\right)$$

$$\begin{cases} x_1 + x_2 \geq 2 \\ x_2 - x_1 \leq 1 \\ x_1 \leq 1 \end{cases}$$

image.png

代码如下，附有详细解释，就不说明了：

```
% 清除工作区
clear;clc;close all;
% 创建决策变量
x = sdpvar(1,2);
% 添加约束条件
C = [
    x(1) + x(2)  >= 2
    x(2)-x(1) <=1
    x(1)<=1
];
% 配置
ops = sdpsettings('verbose',0,'solver','lpsolve');
% 目标函数
z = -(x(1)+2*x(2))/(2*x(1)+x(2)); % 注意这是求解最大值
% 求解
result = optimize(C,z);
if result.problem == 0 % problem =0 代表求解成功
    value(x)
    -value(z)    % 反转
else
    disp('求解出错');
end
```

求解结果:

```
ans =

    0.500000999998279    1.500000333331623

ans =

    1.399999360001426
```

2.2 解决经典的TSP问题

关于TSP的理论，这里我就不详细介绍了，百度有很多。在遇到yalmip之前，我学习的求解TSP的第一解法就是利用lingo来求解，后来学习了几种智能算法，如遗传算法，模拟退火，蚁群算法等等都可以解决这个问题。现在，学习了yalmip之后，我们可以完全抛弃lingo那种简陋的ide。废话不多说，先贴上约束条件：

$$\begin{aligned} \min Z &= \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \\ \text{s.t.} \quad &\begin{cases} \sum_{\substack{i=1 \\ i \neq j}}^n x_{ij} = 1, & j = 1, \dots, n \\ \sum_{\substack{j=1 \\ j \neq i}}^n x_{ij} = 1, & i = 1, \dots, n \\ i - u_j + nx_{ij} \leq n - 1, & 1 < i \neq j \leq n \\ x_{ij} = 0 \text{或} 1, & u, j = 1, \dots, n \\ u_i \text{为实数}, & i = 1, \dots, n \end{cases} \end{aligned}$$

TSP约束条件

再来看看代码：

```
% 利用yalmip求解TSP问题
clear;clc;close all;
d = load('tsp_dist_matrix.txt');
n = size(d,1);
% 决策变量
x = binvar(n,n,'full');
u = sdpvar(1,n);
% 目标
z = sum(sum(d.*x));
% 约束添加
C = [];
for j = 1:n
    s = sum(x(:,j))-x(j,j);
    C = [C, s == 1];
end
for i = 1:n
    s = sum(x(i,:)) - x(i,i);
    C = [C, s == 1];
end
for i = 2:n
    for j = 2:n
        if i~=j
            C = [C,u(i)-u(j) + n*x(i,j)<=n-1];
        end
    end
end
% 参数设置
```

```
ops = sdpsettings('verbose',0);
% 求解
result = optimize(C,z);
if result.problem== 0
    value(x)
    value(z)
else
    disp('求解过程中出错');
end
```

这里用到的 `tsp_dist_matrix.txt` 如下：

```
0 7 4 5 8 6 12 13 11 18
7 0 3 10 9 14 5 14 17 17
4 3 0 5 9 10 21 8 27 12
5 10 5 0 14 9 10 9 23 16
8 9 9 14 0 7 8 7 20 19
6 14 10 9 7 0 13 5 25 13
12 5 21 10 8 13 0 23 21 18
13 14 8 9 7 5 23 0 18 12
11 17 27 23 20 25 21 18 0 16
18 17 12 16 19 13 18 12 16 0
```

最后来看看结果吧：

```
>> value(x)

ans =

    NaN     0     0     0     0     0     0     0     1     0
     0    NaN     1     0     0     0     0     0     0     0
     0     0    NaN     1     0     0     0     0     0     0
     1     0     0    NaN     0     0     0     0     0     0
     0     0     0     0    NaN     0     1     0     0     0
     0     0     0     0     1    NaN     0     0     0     0
     0     1     0     0     0     0    NaN     0     0     0
     0     0     0     0     0     1     0    NaN     0     0
     0     0     0     0     0     0     0     0    NaN     1
     0     0     0     0     0     0     0     1     0    NaN

>> value(z)

ans =

    77
```

最后，发现了yalmip的一个bug，在书写yalmip的约束条件时，如下：

```
x(1) + x(2)-2 >= 0
```

注意 $x(2)-2$ 这个-2是紧贴 $x(2)$ 的。这样求解释正确的，如果换成

$$x(1) + x(2) - 2 \geq 0$$

$x(2)-2$ 这个-2不是紧贴 $x(2)$ 的。这样求解则会出现问题。

建议把所有常数写在等式的右边