## 1 数学模型原理描述

对于大规模的空压机群优化操作的问题，在已知用户需求或者能够近似预测用户用气需求的时候采用线性规划求解。

等熵功率求解如下：



式中：表示等熵功率，表示空压机等熵功率系数，表示吸气压力，表示产气流量，表示多变压缩指数。

等熵效率：



，，表示空压机等熵效率的拟合系数。

空压机实际消耗的功率：



采用线性规划对空压机群进行控制优化。

目标函数：

约束条件：

## 2 yalmip使用

求解多变量混合整形线性规划的时候，由于MATLAB没有自带的优化工具箱可以求解混合整形线性规划问题。必须选择其他的求解器。下面对几种常用的求解器做个说明。1）cplex：cplex是IBM公司一款高性能的数学规划问题求解器，可以快速、稳定地求解线性规划、混合整数规划、二次规划等一系列规划问题。cplex的速度非常快，可以解决现实世界中许多大规模的问题，它能够处理有数百万个约束 (constraint) 和变量(variable) 的问题，而且一直刷新数学规划的最高性能记录。但是cplex是一款收费的商业软件并且用cplex解决问题的话，可能还需要学习特定于cplex的建模语言，所以暂时不考虑。2）GLPK:(GNU Linear Programming Kit，GNU线性编程工具)是GNU下的一个项目，用于建立大规模线性规划LP和混合型整数规划MIP问题，并对模型进行最优化求解。由于是GNU下的项目，因此没有商业非商业的版本限制，可以自由使用。GLPK实现了对windows的支持，但是为此，你同样需要学习它的建模语言，并且所有的操作都在 glpsol.exe 提共的命令行下完成，比较不方便，且耗时长。如果要在matlab下使用，还需要下载额外的驱动文件。3）yalmip:可以说，yalmip是一位“集大成者”，它不仅自己包含基本的线性规划求解算法，比如linprog（线性规划）、bintprog（二值线性规划）、bnb（分支界定算法）等，他还提供了对cplex、GLPK、lpsolve等求解工具包更高层次的包装。更为可贵的是，yalmip真正实现了建模和算法二者的分离，它提供了一种统一的、简单的建模语言，针对所有的规划问题，都可以用这种统一的方式建模；至于用哪种求解算法，你只需要通过一次简单的参数配置指定就可以了，甚至不用你指定，yalmip会自动为你选择最适合的算法。

最重要的是，yalmip的建模语法非常简单，简单到只需要记住四个命令就可以求解问题。

### 2.1 yalmip下载安装

1）下载yalmip，网址如下：

<http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Main.Download>

2）解压缩，把MATLAB路径设为yalmap解压缩路径

3）配置yalmip，在命令窗口运行

cd YALMIPfolderShouldbeHere

urlwrite('https://github.com/yalmip/yalmip/archive/master.zip','yalmip.zip');

unzip('yalmip.zip','yalmip')

addpath(genpath([pwd filesep 'yalmip']));

savepat

4）测试，运行yalmiptest.m程序，这是一个测试程序

5）编写自己要求解的问题。

### 2.2yalmip编程

1 变量

1）实型变量—sdpvar。sdpvar是yalmip的一种核心对象，它代表的是优化问题中的实型决策变量，这种变量可以通过如下方法产生：

P=sdpvar(n,m);%表示P为n\*m的矩阵。

P=sdp(n,n);%表示P为n\*n的方阵。

2）整型变量—intvar。Intvar定义整形变量，应用如下：

P=intvar(3,4);

3)二进制变量—binvar。binvar表示0-1型的整型变量，用用如下：

P=binvar(5,5);

2 约束条件

最常用的集合构造方法set,现在set已经不被使用，直接省略就好。

F=(P>0);%表示唯一的不等式约束

F=(2<diag(P)<5);%约束条件也能够是不等式约束

值得注意的是等式约束，表示如下：

F=(x==0);%等式约束”==”来表示。

约束条件之间的连接直接用加号

如

F=(P>0)+(0<diag(P)<5;

F=F+(x==0);

3 求解函数

用函数solvesdp来求解优化问题

s=solvesdp(F);%求解可行解问题

s=solvesdp(F,f);%求解一般优化问题，其中f为目标函数

s=solvesdp(F,f,options);%设定选项，如选择算法

4 提取结果

求解完成后，用P=double(P)；

## 3 yalmip编程求解

x=sdpvar(1,4);%定义每一台产出体积流量

m=sdpvar(1,4);%定义每一台产出质量流量

Y=binvar(1,4);%定义是否启动/加卸载

a1=sdpvar(1,4);%定义参数a1

a1=[8.35\*10^(-1),1.66,1.72\*10^(-3),5.47\*10^(-1)];%计算a1

a2=sdpvar(1,4);

a2=[1.007\*10^(-5),3.3\*10^(-4),8.49\*10^(-4),2.62\*10^(-5)];

a3=sdpvar(1,4);

a3=[6.287\*10^(-8),9.57\*10^(-8),9.395\*10^(-11),3.518\*10^(-8)];

a4=sdpvar(1,4);

a4=[2.26\*0.1,3.42,7.78\*10^(-4),-1.77\*10^(-2)];

a5=sdpvar(1,4);

a5=[6.44\*10^(-4),7.034\*10^(-4),1.675\*10^(-3),9.9\*10^(-4)];

a6=sdpvar(1,4);

a6=[5.087\*10^(-8),1.94\*10^(-7),4.175\*10^(-11),2.56\*10^(-8)];

b4=sdpvar(1,4);

b4=[0.607,0.636,-3.06\*10^(-4),0.405];

b5=sdpvar(1,4);

b5=[877,751,1521,1252];

b6=sdpvar(1,4);

b6=[-7\*10^5,-6.14\*10^5,-6.5\*10^5,-8.44\*10^5];

S=intvar(1,4);%定义每一台的转速

Ps=sdpvar(1);%出气压力

Ps=6.4;

Pd=sdpvar(1);%吸气压力

Pd=9.0;

%目标函数1.35表示多变指数的值

f=7\*Ps\*x(1)\*(1.35/(1.35-1))\*((Pd/Ps)^0.26-1)/(b4(1)+b5(1)\*(x(1)/S(1))+b6(1)\*(x(1)/S(1)^2))

+Ps\*x(2)\*(1.35/(1.35-1))\*((Pd/Ps)^0.26-1)/(b4(2)+b5(2)\*(x(2)/S(2))+b6(2)\*(x(2)/S(2)^2))

+Ps\*x(3)\*(1.35/(1.35-1))\*((Pd/Ps)^0.26-1)/(b4(3)+b5(3)\*(x(3)/S(3))+b6(3)\*(x(3)/S(3)^2))

+Ps\*x(4)\*(1.35/(1.35-1))\*((Pd/Ps)^0.26-1)/(b4(4)+b5(4)\*(x(4)/S(4))+b6(4)\*(x(4)/S(4)^2));

%约束条件

F=(x==m\*8.3\*293/8);

F=F+(x(1)+x(2)+x(3)+x(4)==25);

F=F+(Y(1)\*(2+a1(1)+a2(1)\*S(1)+a3(1)\*S(1)^2)<=x(1)<=Y(1)\*(2+a4(1)+a5(1)\*S(1)+a6(1)\*S(1)^2));

F=F+(Y(2)\*(a1(2)+a2(2)\*S(2)+a3(2)\*S(2)^2)<=x(2)<=Y(2)\*(a4(2)+a5(2)\*S(2)+a6(2)\*S(2)^2));

F=F+(Y(3)\*(a1(3)+a2(3)\*S(3)+a3(3)\*S(3)^2)<=x(3)<=Y(3)\*(a4(3)+a5(3)\*S(3)+a6(3)\*S(3)^2));

F=F+(Y(4)\*(a1(4)+a2(4)\*S(4)+a3(4)\*S(4)^2)<=x(4)<=Y(4)\*(a4(4)+a5(4)\*S(4)+a6(4)\*S(4)^2));

F=F+(3960<=S(1)<=6405);

F=F+(3960<=S(2)<=3960);

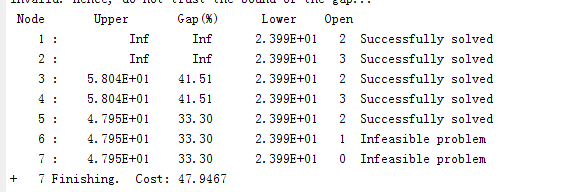
F=F+(3120<=S(3)<=5040);

F=F+(3380<=S(4)<=5460);

%求解

solvesdp(F,f);

求解结果：



我们可以看到对于只有四台空压机的小型模型而言，yalmip能很快的找到最优解，并且我们可以提取出想要的结果。

每台空压机的产气质量流量Kg/min



每台空压机的转速rad/min

