Linear Programming

Chongnan Li
chongnanli1997@hotmail.com

Additional content 1: Dantzig-Wolfe Decomposition

● Introduction

In this file, we would give the coding fulfillment of D-W Decomposition.

● Dantzig-Wolfe Decomposition MATLAB Code

A MATLAB function DantzigWolfeDecomp

function [**x, fval, bound, exit_flag**] = DantzigWolfeDecomp(**mast, sub, K**)

is given below that implements the Dantzig-Wolfe decomposition. The argument **mast** is a struct that contains the coefficients of the linking constraints L_i and **sub** is a struct that contains the coefficient s of subproblems, i.e., $\boldsymbol{c}^i$, A_i and $\boldsymbol{b}^i$ as well as the initial extreme points for each subproblem. **K** is the number of subproblems.

Consider the linear program in Example 5.1:

$$
\begin{array}{ll}
\text{Minimize} & -2x_1 - 3x_2 - 5x_3 - 4x_4 \\
\text{subject to} & \\
& \begin{array}{rrrrcl}
x_1+ & x_2+ & 2x_3 & & \leq & 4 \\
 & x_2+ & x_3+ & x_4 & \leq & 3 \\
2x_1+ & x_2 & & & \leq & 4 \\
x_1+ & x_2 & & & \leq & 2 \\
 & & x_3+ & x_4 & \leq & 2 \\
 & & 3x_3+ & 2x_4 & \leq & 5
\end{array} \\
& x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0
\end{array}
$$

The LP is in the form

$$
\begin{array}{ll}
\text{Minimize} & \boldsymbol{c}^T \boldsymbol{x} \\
\text{subject to} & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\
& \boldsymbol{x} \geq \boldsymbol{0}
\end{array}
$$

where that data $\boldsymbol{c}$, $\boldsymbol{A}$, and $\boldsymbol{b}$ are entered in MATLAB as

```
c=[-2;-3;-5;-4];
A=[1,1,2,0;
   0,1,1,1;
   2,1,0,0;
   1,1,0,0;
   0,0,1,1;
   0,0,3,2];
b=[4;3;4;2;2;5];
```

There are two subproblems, and so **K** is set equal to 2, in MATLAB we have

```
K=2;  %number of subproblems
```

Now, the first two constraints of the LP are the linking constraints and the submatrices and corresponding right-hand-side values are

$$L_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\ L_2 = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}, \boldsymbol{b}^0 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

and are written in MATLAB as

```
mast.L{1}=A(1:2,1:2);
mast.L{2}=A(1:2,3:4);
mast.b=b(1:2);
```

The cost coefficient vector, constraint matrix, and right-hand coefficient vector of the first subproblem are given as

$$\boldsymbol{c}^1 = \begin{bmatrix} -2 \\ -3 \end{bmatrix},\ A_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix},\ \boldsymbol{b}^1 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

and the corresponding MATLAB statements to represent these matrices are

```
sub.c{1}=c(1:2);
sub.A{1}=A(3:4,1:2);
sub.b{1}=b(3:4);
```

The cost coefficient vector, constraint matrix and right hand coefficient vector of the second subproblem are given as

$$\boldsymbol{c}^2 = \begin{bmatrix} -5 \\ -4 \end{bmatrix},\ A_2 = \begin{bmatrix} 1 & 1 \\ 3 & 2 \end{bmatrix},\ \boldsymbol{b}^1 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

and the correponding MATLAB statements to represent these matrices are

```
sub.c{2}=c(3:4);
sub.A{2}=A(5:6,3:4);
sub.b{2}=b(5:6);
```

The initial extreme points for each subproblem are set to the origin in the feasible set for each subproblem and in MATLAB are written as

```
sub.v{1}=zeros(length(sub.c{1}),1);
sub.v{2}=zeros(length(sub.c{2}),1);
```

Finally, once the data is entered as above, the function DantzigWolfeDecomp can be called by the following MATLAB statement

```
[x_DanWof, fval_DanWof, iter, exitflag_DanWof] =
DantzigWolfeDecomp(mast,sub,K)
```

The optimal solution and the optimal objective function can be accessed in MATLAB by entering x_DanWof at the prompt.