Lince Rumainum

DSA 5113

Advanced Analytics & Metaheuristics

Exam I

Problem 1 – Team Work!

Set:

$P = \{1, 2, 3, 4, 5, 6, 7, 8\}$ – these are the 8 problems on the assignment

Decision Variables:

X_p: whether or not I'm choosing to work on a problem (binary value – 1 indicates work on the problem and 0 indicates otherwise), where $p \in P$

Z: help to activated constraint for disjunctive constraint (binary value)

Constraints:

(a) You **cannot work on all** the problems.

$$\sum_{p}^{P} x_p \leq 7$$

(b) You **must choose at least two** of the problems to work on.

$$\sum_{p}^{P} x_p \geq 2$$

(c) **If** you **choose** problem 2, **then you must choose** problem 7 or 8.

$$X_2 \leq X_7 + X_8$$

(d) Problem 2 **cannot be chosen if** problem 4 **is chosen**.

$$X_2 + X_4 \leq 1$$

(e) Problem 5 **can be chosen only if** problem 3 **is also chosen**.

$$X_5 \leq X_3$$

(f) You must **choose either both problems** 1 and 8 **or neither**.

$$X_1 + X_8 \geq 2 - MZ$$

$$X_1 + X_8 \leq M(1 - Z)$$

where $M = 2$ (sufficient to make both Right Hand Sides to be either zero or two)

When *Z* is zero, I need to choose both problems 1 and 8 but if *Z* is one then I have to choose neither.

(g) You must choose **at least one** of the problems 1, 2, and 5 **or at least two** problems from 3, 4, 6, and 8.

$$X_1 + X_2 + X_5 \geq 1 - MZ$$

$$X_3 + X_4 + X_6 + X_8 \geq 2 - M(1 - Z)$$

where $M = 2$ (sufficient to make both Right Hand Sides to be less than or equal to zero when not activated)

When *Z* is zero, I need to choose at least one of the problems 1, 2, and 5 but if *Z* is one then I have to choose at least two problems from 3, 4, 6, and 8.

Problem 2 – Post COVID Party Planning

Assumptions:

- The amount of ingredients (in liters) to buy does not need to be an integer number.
- The amount of beverages and toxic waste by-product (in liters) being produced does not need to be an integer number.

Set:

INGREDIENTS = {1, 2, 3} – the three types of ingredient that can be mixed
PRODUCTS = {Beverage A, Beverage B, Waste} – consumable products and toxic by-product that can produced from mixing the ingredients

Parameters:

cost{INGREDIENTS} – Costs per liter to buy each ingredients
disposalFee – Cost per liter of disposal fee to dispose the toxic waste = $8.00
maxSpending – Money available to spend to buy the ingredients = $300.00

Decision Variables:

X_i: the amount of ingredients to buy (in liters) - $i \in INGREDIENTS$
Y_p: the amount of products that can be produced from the mixed ingredients (in liters) - $p \in PRODUCTS$

The **objective** is to maximize the consumable beverage that can be produced:

$$y_{BeverageA} + y_{BeverageB}$$

Constraints: (**Note:** $i \in INGREDIENTS$)

(Minimum Amount of Type 1 Ingredient)

$$X_1 \geq 0.45 \sum_{i}^{I} X_i$$

(Minimum Amount of Type 2 Ingredient)

$$X_2 \geq 0.10 \sum_{i}^{I} X_i$$

(Maximum Amount of Type 3 Ingredient)

$$X_3 \leq 0.30 \sum_{i}^{I} X_i$$

(Beverage A Produced)

$$Y_{BeverageA} = 0.40 \sum_{i}^{I} X_i$$

(Beverage B Produced)

$$Y_{BeverageB} = 0.25 \sum_{i}^{I} X_i$$

(Toxic Waste Produced)

$$Y_{Waste} = 0.35 \sum_{i}^{I} X_i$$

(Budget)

$$disposalFee \cdot Y_{Waste} + \sum_{i}^{I} X_i \cdot cost_i \leq maxSpending$$

Part b

Below is the ***data file*** for this problem:

```
#Exam 1
#Lince Rumainum
#AMPL data file for Problem 2
#Post COVID Party Planning

#indicate it's a data file-----------------------------------------
data;

#parameters and sets-----------------------------------------------
# Type of ingredients
set INGREDIENTS := 1 2 3;
# Products that will be produced from mixing the ingredients
# BevA  - Beverage A
# BevB  - Beverage B
# Waste - Toxic Waste By-Product
set PRODUCTS := BevA BevB Waste;

# Costs per liter to buy each ingredients (in dollar)
param: cost:=
    1    12
    2    25
    3     9;

# Cost per liter of disposal fee to dispose the toxic waste
param disposalFee = 8;

# Money available to spend to buy the ingredients (in dollar)
param maxSpending = 300;
```

(Continue on the next page)

Below is the ***model file*** for this problem:

```
#Exam 1
#Lince Rumainum
#AMPL model file for Problem 2
#Post COVID Party Planning

#reset ampl-------------------------------------------------------------------
reset;

#options----------------------------------------------------------------------
option solver cplex;
option cplex_options 'sensitivity';

#parameters and sets----------------------------------------------------------
# Type of ingredients
set INGREDIENTS;
# Products that will be produced from mixing the ingredients
# BevA  - Beverage A
# BevB  - Beverage B
# Waste - Toxic Waste By-Product
set PRODUCTS;

# Costs per liter to buy each ingredients (in dollar)
param cost {INGREDIENTS} >= 0;
# Cost per liter of disposal fee to dispose the toxic waste
param disposalFee >= 0;
# Money available to spend to buy the ingredients (in dollar)
param maxSpending >= 0;

#decision variables-----------------------------------------------------------
# the amount of ingredients to buy (in liters)
var x{i in INGREDIENTS} >= 0;
# the amount of products that can be produced from the mixed ingredients (in liters)
var y{p in PRODUCTS} >= 0;

#objective: maximize amount of Beverages to make--------------------------------
maximize totalBeverages: y['BevA'] + y['BevB'];

#constraints:-----------------------------------------------------------------
#amount of type 1 ingredient AT LEAST 45% of the mix
subject to MinAmountOfType1ingredient: x[1] >= 0.45 * sum {i in INGREDIENTS} x[i];
#amount of type 2 ingredient AT LEAST 10% of the mix
subject to MinAmountOfType2ingredient: x[2] >= 0.10 * sum {i in INGREDIENTS} x[i];
#amount of type 3 ingredient NO MORE THAN 30% of the mix
subject to MaxAmountOfType3ingredient: x[3] <= 0.30 * sum {i in INGREDIENTS} x[i];

#amount of Beverage A produced is EXACTLY 40% of the mix
subject to BevAProduced: y['BevA'] = 0.40 * sum {i in INGREDIENTS} x[i];
#amount of Beverage B produced is EXACTLY 25% of the mix
subject to BevBProduced: y['BevB'] = 0.25 * sum {i in INGREDIENTS} x[i];
#amount of Toxic Waste by-product is EXACTLY 35% of the mix
subject to ToxicWasteProduced: y['Waste'] = 0.35 * sum {i in INGREDIENTS} x[i];

#budget
subject to budget: disposalFee * y['Waste'] + sum {i in INGREDIENTS} x[i]*cost[i] <= maxSpending;

#data------------------------------------
data Rumainum-Exam1-p2.dat;

#commands---------------------------------
solve; # solve to maximize total amount of beverages

#spaces
printf "\n\n";

# display the total amount of each ingredients
printf "Total amount of each ingredients \n";
for {i in INGREDIENTS} {
    printf " Ingredient %s: %7.4f liters \n", i, x[i];
}
printf "\n"; #space
# display total amount of ingredients
printf "Total amount of ingredients: %0.4f liters \n",sum {i in INGREDIENTS} x[i];

printf "\n\n";#spaces
#display y;
printf "Total amount of each Beverage: \n";
for {p in PRODUCTS} {
    if p != 'Waste' then printf " %s: %6.4f liters \n", p, y[p];
}
printf "\n"; #space
# display total amount of beverages produced
printf "Total amount of consumable products   : %7.4f liters \n", y['BevA'] + y['BevB'];

# display total amount of toxic waste
printf "Total amount of Toxic Waste by-product: %7.4f liters \n", y['Waste'];

printf "\n\n"; #spaces

# display total costs of ingredients and disposal fee
printf "Total cost of all ingredients: $%6.2f \n",sum {i in INGREDIENTS} x[i]*cost[i];
printf "Total cost of disposal fee   : $%6.2f \n",disposalFee * y['Waste'];

printf "\n\n"; #spaces

#---FOR PART C
#display the shadow price of the budget constraint and its range of feasibility
display budget, budget.up, budget.down;
```

So, using AMPL to solve this problem, the total amount of each beverage and each ingredient are shown below: (**Note**: *BevA* is Beverage A and *BevB* is Beverage B)

```
Total amount of consumable products   : 12.8289 liters
Total amount of Toxic Waste by-product:  6.9079 liters


Total cost of all ingredients: $244.74
Total cost of disposal fee   : $ 55.26


ampl: model Rumainum-Exam1-p2.mod;
CPLEX 12.9.0.0: optimal solution; objective 12.82894737
3 dual simplex iterations (1 in phase I)


Total amount of each ingredients
 Ingredient 1: 11.8421 liters
 Ingredient 2:  1.9737 liters
 Ingredient 3:  5.9211 liters

Total amount of ingredients: 19.7368 liters


Total amount of each Beverage:
 BevA: 7.8947 liters
 BevB: 4.9342 liters

Total amount of consumable products   : 12.8289 liters
Total amount of Toxic Waste by-product:  6.9079 liters


Total cost of all ingredients: $244.74
Total cost of disposal fee   : $ 55.26
```

Part c

```
budget = 0.0427632
budget.up = 1e+20
budget.down = 0
```

Interpreting the shadow price and the range of feasibility of the $300 budget constraint from the AMPL results above, it shows that:

- For every $1 **increase** (essentially up to infinity) of the budget (parameter: *maxSpending*), the total amount of consumable beverages (Beverage A & Beverage B) will **increase** 0.0427632 liter.
- For every $1 **decrease** down to $0 of the budget (parameter: *maxSpending*), the total amount of consumable beverages (Beverage A & Beverage B) will **decrease** 0.0427632 liter.

Code files:

Lince_Rumainum_Exam1_p2.dat & Lince_Rumainum_Exam1_p2.mod.

Problem 3 – Mysterious Constraints

Decision Variables:

$x_i \ \forall i \in I$

$y_i \ binary \ \forall i \in I$

Z

MIP formulation

The **objective** is max life-happiness-success: …

Constraints:

…

$$0 \le x_i \le \left(\frac{1+\sqrt{5}}{2}\right) \quad \forall i \in I \text{ -------------------- (1)}$$

$$Z \le x_i \quad \forall i \in I \text{ -------------------------------- (2)}$$

$Z \geq x_i - \left(\frac{1+\sqrt{5}}{2}\right)(1 - y_i) \;\; \forall i \in I$ ---- (3)
$\sum_{i \in I} y_i = 1$ -------------------------------- (4)
$y_i \in \{0, 1\} \;\; \forall i \in I$ ------------------------ (5)

…

Although not knowing the whole constraints, logically from constraint #3 and constraint #4, it shows that the problem has either/or behavior, which is one of Mixed Integer Problem constraints. One of the disjunctive constraints is constraint #3 and the "or" part is an unknown constraint. Let's look at the general notation of the disjunctive constraints for any two constraints:

$$a_1^T x \leq b_1 + Mz \text{ ------ (a)}$$

$$a_2^T x \leq b_2 + M(1 - z) \text{ ------ (b)}$$

$$z \in \{0, 1\}$$

Where when z = 0, constraint notation a is activated and when z = 1, constraint notation b is activated.

For constraint #3, although the notation is not written exactly the same as the constraint notation *b*, it is that constraint with an unknown constraint notation *a*. Now, rearranging constraint #3 as constraint notation *b*, we have:

$$x_i \leq Z + \left(\frac{1+\sqrt{5}}{2}\right)(1 - y_i)$$

Here the value of $M = \left(\frac{1+\sqrt{5}}{2}\right)$ as "the large constant number" of the disjunctive constraints is sufficient because of constraint #2: $Z \leq x_i \;\; \forall i \in I$ and constraint #1: $0 \leq x_i \leq \left(\frac{1+\sqrt{5}}{2}\right) \;\; \forall i \in I$. So, even when constraint #3 is not chosen, x_i could still be any number and still be less than $\left(\frac{1+\sqrt{5}}{2}\right)$.

When constraint #3 is activated, $y_i = 1$, which means $x_i \leq Z$. To be true and still within constraint #2:

$$Z = x_i$$

When constraint #3 is inactive, $y_i = 0$, the unknown constraint is chosen and constraint #3 becomes the lower bound for decision variable *Z*. The constraint becomes: $Z \geq x_i - \left(\frac{1+\sqrt{5}}{2}\right) \;\; \forall i \in I$. Combining it with constraint #2, it shows the range of value for *Z*, which is $\left(\frac{1+\sqrt{5}}{2}\right)$, for examples:

$$at\ x_i = 0, \qquad -\left(\frac{1+\sqrt{5}}{2}\right) \leq Z \leq 0 \;\; \boldsymbol{OR} \;\; at\ x_i = \left(\frac{1+\sqrt{5}}{2}\right), \qquad 0 \leq Z \leq \left(\frac{1+\sqrt{5}}{2}\right)$$

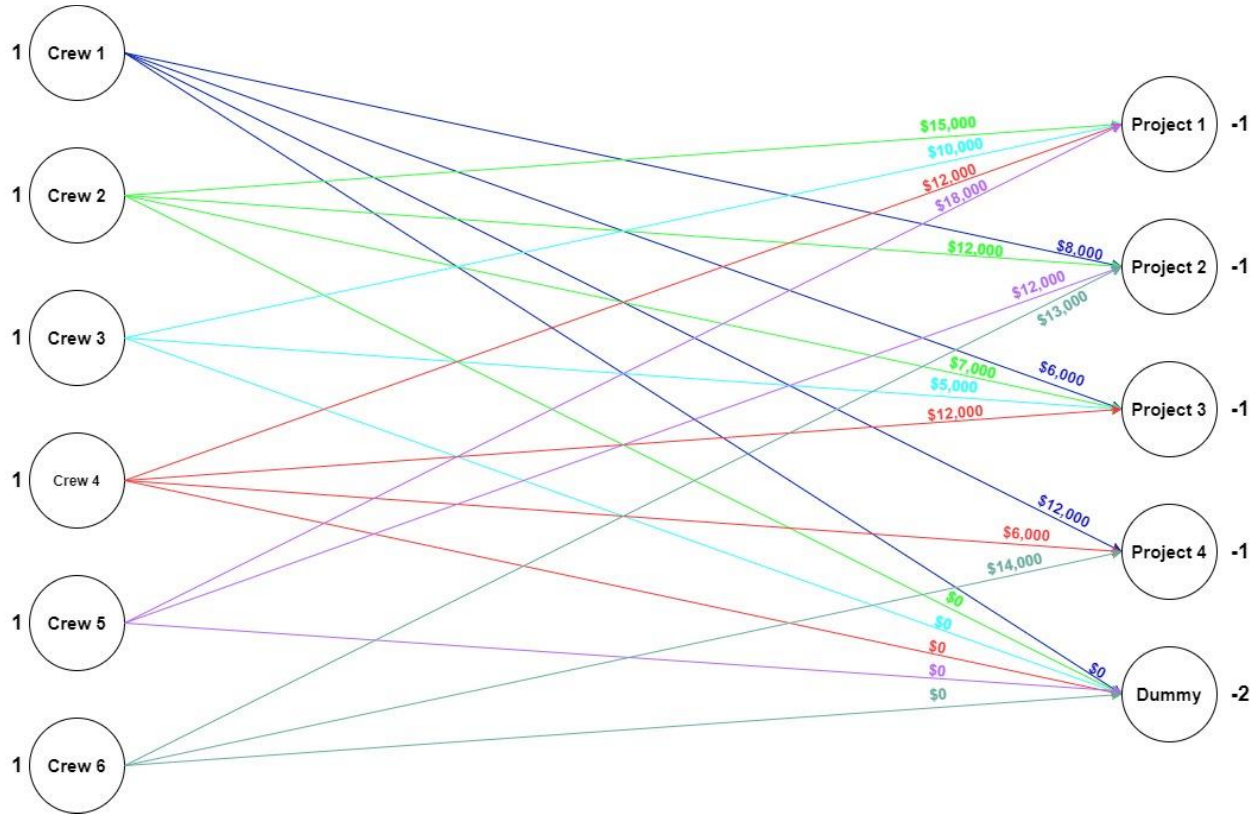
$Combining\ the\ known\ constraints, a\ general\ notation\ for\ the\ range\ of\ value\ for\ Z$:

$$x_i - \left(\frac{1+\sqrt{5}}{2}\right)(1 - y_i) \leq Z \leq x_i$$

Problem 4 – Work Crews

Part a

Formulating the problem as a minimum cost network flow problem, we have:



The left-side of the network is the supply nodes and the right-side of the network is the demand nodes. Since we know that each crew can do at most one project, each of the crews has a supply node value of 1 and no project can use more than one crew, each of the projects has a demand node value of -1. Also, since not all crews have to be assigned, to balance the network where supply is equal to demand, a "dummy" node is needed with demand node value of -2. The costs of each crew-project assignment are noted on the arcs of the crew-project assignment network. The objective of this problem is to find the minimum cost of crew-project assignment problem and its flow paths.

(Continue on the next page)

Part b

Using the ***data file*** below and the mcnfp.txt model file to solve it,

```
#Exam 1
#Lince Rumainum
#AMPL data file for Problem 4
#Work Crews

#indicate it's a data file----------------------------------------
data;

#parameters and sets----------------------------------------------
#all the nodes needed for the Work Crews Network
# DUMMY node is used to balanced the network
set NODES := CREW1 CREW2 CREW3 CREW4 CREW5 CREW6 PROJECT1 PROJECT2 PROJECT3 PROJECT4 DUMMY;

#Arcs flow from one node to another in the network
# what projects can each crew possibly do
set ARCS := (CREW1, *) PROJECT2 PROJECT3 PROJECT4 DUMMY
            (CREW2, *) PROJECT1 PROJECT2 PROJECT3 DUMMY
            (CREW3, *) PROJECT1 PROJECT3 DUMMY
            (CREW4, *) PROJECT1 PROJECT3 PROJECT4 DUMMY
            (CREW5, *) PROJECT1 PROJECT2 DUMMY
            (CREW6, *) PROJECT2 PROJECT4 DUMMY;

# positive values indicate they are the "supply" nodes
# negative values indicate they are the "demand" nodes
param b :=
    CREW1 1
    CREW2 1
    CREW3 1
    CREW4 1
    CREW5 1
    CREW6 1
    PROJECT1 -1
    PROJECT2 -1
    PROJECT3 -1
    PROJECT4 -1
    DUMMY    -2;

#the cost of one unit flow on (in thousands of dollar) (i in CREWS,j in PROJECTS)
param c: PROJECT1 PROJECT2 PROJECT3 PROJECT4 :=
    CREW1   .          8        6       12
    CREW2   15        12        7        .
    CREW3   10         .        5        .
    CREW4   12         .       12       16
    CREW5   18        17        .        .
    CREW6   .         13        .       14;
```

We find that the minimum cost for the crew-project assignment is $39,000. This can be done by assigning Project 1 to Crew 3 for $10,000, Project 2 to Crew 1 for $8,000, Project 3 to Crew 2 for $7,000, and Project 4 to Crew 6 for $14,000 while Crew 4 and Crew 5 are not assigned to any of the projects.

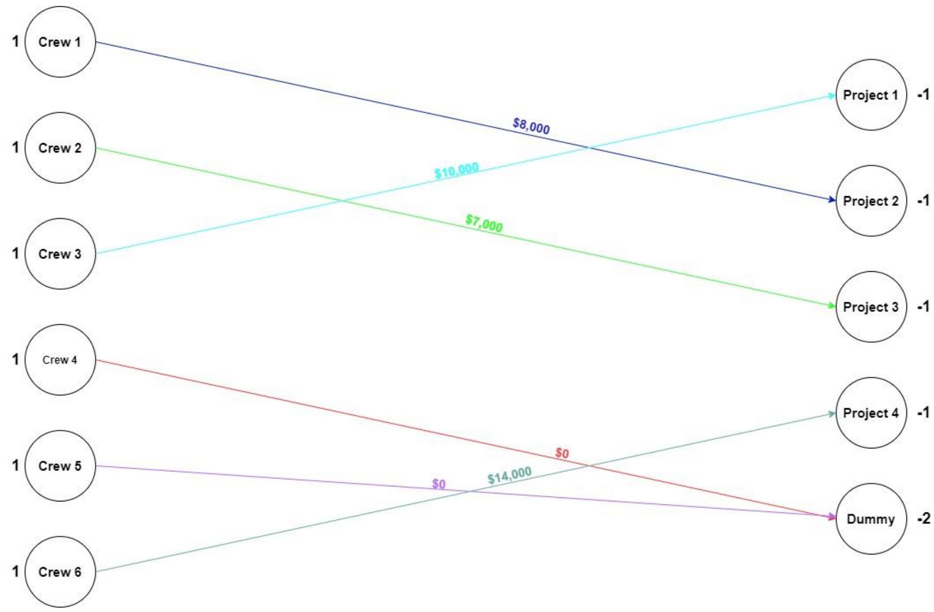
```
The minimum cost is $39000.00

The flow path taken to minimize the cost:
x [*,*]
:     DUMMY PROJECT1 PROJECT2 PROJECT3 PROJECT4    :=
CREW1    0      .        1        0        0
CREW2    0      0        0        1        .
CREW3    0      1        .        0        .
CREW4    1      0        .        0        0
CREW5    1      0        0        .        .
CREW6    0      .        0        .        1
;


The above data mean, the flow path taken to  minimize the cost are:
CREW1 -----> PROJECT2
CREW2 -----> PROJECT3
CREW3 -----> PROJECT1
CREW4 -----> DUMMY
CREW5 -----> DUMMY
CREW6 -----> PROJECT4


The cost for the selected crews doing each project:
CREW1 will work on PROJECT2 for $ 8000.00
CREW2 will work on PROJECT3 for $ 7000.00
CREW3 will work on PROJECT1 for $10000.00
CREW6 will work on PROJECT4 for $14000.00
```

Below is the optimal path solution for the crew-project assignment problem:



Code files:
Lince_Rumainum_Exam1_p4.dat & Lince_Rumainum_Exam1_p4.mod.

Problem 5 – Mix Tape
Set:
SIDES = {A, B} – the two sides of the mix tape
SONGS = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11} – the eleven choices of songs to put on the mix tape

Parameters:
length{SONGS} – tape lengths for each songs (in cm)

Decision Variables:
$X_{side,song}$: whether or not the song is chosen and for which side of the mix tape (binary value – 1 indicates chosen and 0 indicates otherwise), where $side \in SIDES$ and $song \in SONGS$

The **objective** is to **minimize the total length of the mix tape by**

$$\sum_{s}^{S} x_{A,s} length_s$$

where $s \in SONGS$ = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}, list of songs available

Constraints:
(total length of the mix tape), where $s \in SONGS$

$$\sum_{s}^{S} x_{A,s} length_s + \sum_{s}^{S} x_{B,s} length_s = \sum_{s}^{S} length_s$$

(total length of Side A of the mix tape), where $s \in SONGS$

$$\sum_{s}^{S} x_{A,s} length_s \geq 1/2 \sum_{s}^{S} length_s$$

(if song is on Side A then it can't be on Side B)

$$x_{A,s} + x_{B,s} \leq 1, \text{ where } s \in \ SONGS$$

(non-zero tape length) $length_s \geq \ 0$, where $s \in \ SONGS$

Part b

Below is the ***data file*** for the Mix Tape problem:

```
#Exam 1
#Lince Rumainum
#AMPL data file for Problem 5
#Mix Tape

#indicate it's a data file------------------
data;

#parameters and sets-------------------------
#Sides of the mix tape
set SIDES := A B;
#Songs available to put on the mix tape
set SONGS := 1 2 3 4 5 6 7 8 9 10 11;

#tape length for each song (in cm)
param: length :=
 1      44
 2      67
 3      37
 4      54
 5      79
 6      56
 7      35
 8      37
 9      53
10      70
11      37 ;
```

Below is ***model file*** for the Mix Tape problem:

```
#Exam 1
#Lince Rumainum
#AMPL model file for Problem 5
#Mix Tape

#reset ampl-------------------------------------------------------------------
reset;

#options----------------------------------------------------------------------
option solver cplex;

#parameters and sets----------------------------------------------------------
set SIDES; # Side of the mix tape
set SONGS; # Songs available to put on the mix tape

param length {SONGS} >= 0; # tape length for each song (in cm)

#decision variables-------------------------------------------------------------
# whether or not the song is chosen and for which side of the mix tape
var x{side in SIDES, song in SONGS} binary;

#objective: minimize the length of the mix tape
#       by minimize the length of side A-------------------------------------
minimize totalLengthOfTape: sum {song in SONGS} x['A',song]*length[song];

#constraints:-------------------------------------------------------------------
# the total length of side A added to total length of side B HAVE TO BE EQUAL TO the total length of all songs available
subject to totalLength: sum {song in SONGS} x['A',song]*length[song] + sum {song in SONGS} x['B',song]*length[song] = sum {song in SONGS} length[song]
# the side A of the mix tape need to be GREATER THAN the total length of all songs available
subject to sideAsLength: sum {song in SONGS} x['A',song]*length[song] >= (1/2) * sum {song in SONGS} length[song];
# if the song is on side A then it can't be on side B
subject to pickAllSong{song in SONGS}: x['A',song] + x['B',song] <= 1;

#data------------------------------------
data Rumainum-Exam1-p5.dat;

#commands--------------------------------
solve; # solve to minimize the total length of the tape

#spaces
printf "\n\n";

#Display the optimal objective solution
printf "The shortest total tape length for the mix tape: %d cm \n", totalLengthOfTape;

#spaces
printf "\n\n";

# display
for {side in SIDES} {
    printf "Songs on Side %s: \n", side;
    for {song in SONGS} {
        #display the songs that chosen for this side of the tape and its length
        if x[side, song] > 0 then printf "Song %2.0d with tape length: %2.0d cm \n", song, length[song];
    }
    printf "\n"; #space
    #Display the total length of each side of the tape
    printf "Total length for Side %s: %d cm \n", side, sum {song in SONGS} x[side,song]*length[song];
    printf "\n\n"; #spaces
}
```

Below are the results for solving the Mix Tape problem using AMPL:

```
ampl: model Rumainum-Exam1-p5.mod;
CPLEX 12.9.0.0: optimal integer solution; objective 285
13 MIP simplex iterations
0 branch-and-bound nodes


The shortest total tape length for the mix tape: 285 cm


Songs on Side A:
Song  2 with tape length: 67 cm
Song  6 with tape length: 56 cm
Song  7 with tape length: 35 cm
Song  8 with tape length: 37 cm
Song  9 with tape length: 53 cm
Song 11 with tape length: 37 cm

Total length for Side A: 285 cm


Songs on Side B:
Song  1 with tape length: 44 cm
Song  3 with tape length: 37 cm
Song  4 with tape length: 54 cm
Song  5 with tape length: 79 cm
Song 10 with tape length: 70 cm

Total length for Side B: 284 cm
```

The optimal objective is 285 cm of tape length to be used for the eleven song choices for the mix tape.
Side A will have song 2, 6, 7, 8, 9, and 11 with a total tape length of 285 cm, while
Side B will have song 1, 3, 4, 5, and 10 with a total tape length of 284 cm.

Code files:

Lince_Rumainum_Exam1_p5.dat & Lince_Rumainum_Exam1_p5.mod.