Final Exam
Adv. Analytics and Metaheuristics

Daniel Carpenter

March 2022

Contents

1 - *Question 1* (Version 1)

1.1 *Part 1:* Mathematical Formulation

1.1.1 Sets

$NewBuildTypes$: Set of new build types $b \in (Homes,\ Duplex,\ MiniPark)$

1.1.2 Parameters

Parameter	Description	Default Value
$budget$	Federal grant allocation to revitalize neighborhoods	\$15MM total budget
$maxBuildingDemod$	Max amount of buildings that can be demolished	300 total buildings
$demoCost$	Cost of demolishing a building	\$4,000 per building
$freedUpSpace$	Acreage generated from demolishing a building	0.25 per building
$newBuildSpace_b$	Amount of acreage that a new building ($b \in NewBuildTypes$) consumes	$Homes$: 0.2, $Duplex$: 0.4, $MiniPark$: 1.0
$newBuildTax_b$	Amount of tax dollars generated from a new building ($b \in NewBuildTypes$)	$Homes$: 1,500, $Duplex$: 2,750, $MiniPark$: 500
$newBuildCost_b$	Amount of dollars used to create a new building ($b \in NewBuildTypes$)	$Homes$: 150,000, $Duplex$: 190,000, $MiniPark$: 20,000
$newBuildPercShare_b$	Minimum required percentage share of new buildings ($b \in NewBuildTypes$) created	$Homes$: 20%, $Duplex$: 10%, $MiniPark$: 5%

1.1.3 Decision Variables

Variable	Description
$numOldBuildsDemod$	Number of old buildings to demolish
$numNewBuilds_b$	Number of new buildings ($b \in NewBuildTypes$) to produce
$newBuildTotalCost$	Variables to hold total cost of new builds ($b \in NewBuildTypes$). Calculation: $\sum_{b \in NewBuildTypes}(numNewBuilds_b \times newBuildCost_b)$
$oldDemoTotalCost$	Variables to hold total cost of old demolitions. Calculation: $numOldBuildsDemod \times demoCost$
$sumOfNewBuilds$	# Variable to hold the sum of all new build types over all New build types ($b \in NewBuildTypes$). Calculation: $\sum_{b \in NewBuildTypes}(numNewBuilds_b)$

1.1.4 Objective

Maximize the tax revenue from the projects

$$maximize\ taxRevenue : \sum_{b \in NewBuildTypes}(numNewBuilds_b \times newBuildTax_b)$$

1.1.5 Constraints

C1 Spend less than or equal to the federal budget (see variable definitions)

$$meetTheBudget : newBuildTotalCost + oldDemoTotalCost \leq budget$$

C2 Can only produce new builds using the demolished buildings land

$$useAvailLand : \sum_{b \in NewBuildTypes}(numNewBuilds_b \times newBuildSpace_b)$$
$$\leq numOldBuildsDemod \times freedUpSpace$$

C3 Can only clear a certain amount of old buildings

$$maxBuildingsCleared : numOldBuildsDemod \leq maxBuildingDemod$$

C4 For each new build type ($b \in NewBuildTypes$), the percentage share of the new build type must meet the minimum required (see variables)

$$share : numNewBuilds_b \geq newBuildPercShare_b \times sumOfNewBuilds,$$
$$\forall\ b \in Businesses$$

C5 Non-negativity and integer constraints

$$numOldBuildsDemod \in \mathbb{Z}, \geq 0$$
$$numNewBuilds_b \in \mathbb{Z}, \geq 0,\ \forall\ b \in NewBuildTypes$$

1.2 *Part 2*: AMPL Code & Output

1.2.1   AMPL Code

Data *problem1.dat*

```
data;

# SETS ==============================================================

# Set of new build types
set NewBuildTypes := Homes Duplex MiniPark;

# PARAMETERS ========================================================

param budget            := 15000000; # federal budget
param maxBuildingDemod := 300;       # max buildings can be demo'd
param demoCost          := 4000;     # Cost of each demolition
param freedUpSpace     := 0.25;      # Freed up space from demolition

# Amount of acreage that a new building (b in NewBuildTypes) consumes
param: newBuildSpace :=
       Homes     0.2
       Duplex    0.4
       MiniPark 1.0
       ;
# Amount of tax dollars generated from a new building (b in NewBuildTypes)
param: newBuildTax :=
       Homes     1500
       Duplex    2750
       MiniPark 500
       ;

# Amount of dollars used to create a new building (b in NewBuildTypes)
param: newBuildCost :=
       Homes     150000
       Duplex    190000
       MiniPark 20000
       ;

# Minimum required percentage share of new buildings (b in NewBuildTypes) created
param: newBuildPercShare :=
       Homes     0.20
       Duplex    0.10
       MiniPark 0.05
```

;

Model *problem1.mod*

```
reset;                    # Reset globals
options solver cplex;    # Using cplex for simplex alg

# SETS ================================================================
set NewBuildTypes; # Set of new build types

# PARAMETERS ================================================================
param budget            >= 0; # federal budget
param maxBuildingDemod  >= 0; # max buildings can be demo'd
param demoCost          >= 0; # Cost of each demolition
param freedUpSpace      >= 0; # Freed up space from demolition

param newBuildSpace    {NewBuildTypes} >= 0; # new build acreage
param newBuildTax      {NewBuildTypes} >= 0; # n.b. tax generation
param newBuildCost     {NewBuildTypes} >= 0; # n.b. cost
param newBuildPercShare{NewBuildTypes} >= 0; # n.b. min % share

# DECISION VARIABLES ================================================
var numOldBuildsDemod                    >= 0 integer; # Num old builds to demo
var numNewBuilds      {NewBuildTypes} >= 0 integer; # Num new builds to create

# Variables to hold total cost of new builds over all types
var newBuildTotalCost = sum{b in NewBuildTypes} ( (numNewBuilds[b] * newBuildCost[b]));

# Variables to hold total cost of old demolitions
var oldDemoTotalCost = (numOldBuildsDemod * demoCost) ;

# Variable to hold the sum of all new build types over all New build types
var sumOfNewBuilds =  sum{b in NewBuildTypes}( numNewBuilds[b] );

# OBJECTIVE FUNCTION ================================================
maximize taxRevenue: sum{b in NewBuildTypes}( numNewBuilds[b] * newBuildTax[b] );

# CONSTRAINTS ================================================

# C1  Spend less than or equal to the federal budget
s.t. meetTheBudget:
    newBuildTotalCost + oldDemoTotalCost <= budget ;

# C2 Can only produce new builds using the demolished buildings land
```

5

```
s.t. useAvailLand:
    sum{b in NewBuildTypes}( numNewBuilds[b] * newBuildSpace[b] )
    <= numOldBuildsDemod * freedUpSpace ;

# C3 Can only clear a certain amount of old buildings
s.t. maxBuildingsCleared: numOldBuildsDemod <= maxBuildingDemod ;

# C4 For each new build type (b in NewBuildTypes),
#    the percentage share of the new build type must meet the minimum required
s.t. share {b in NewBuildTypes}:
    numNewBuilds[b] >= newBuildPercShare[b] * sumOfNewBuilds ;

# CONTROLS ============================================================
    data problem1.dat; # retreive data file with sets/param. values
    solve;

    print;

    print "Number of old buildings to demolish and cost (dollars):";
    display numOldBuildsDemod, oldDemoTotalCost ;

    print "Number of new buildings produced and cost (dollars):";
    display numNewBuilds , newBuildTotalCost ;

    print "Total Budget Used (dollars):";
    display newBuildTotalCost + oldDemoTotalCost ;

    print "Part 3: Max Tax Revenue generated (dollars):";
    display taxRevenue ;
```

1.2.2 *Part 2/3:* Solve AMPL Model and Display Solution

```
CPLEX 20.1.0.0: optimal integer solution; objective 199000
6 MIP simplex iterations
0 branch-and-bound nodes

Number of old buildings to demolish and cost (dollars):
numOldBuildsDemod = 137
oldDemoTotalCost = 548000

Number of new buildings produced and cost (dollars):
numNewBuilds [*] :=
   Duplex  62
    Homes  17
MiniPark   6
;

newBuildTotalCost = 14450000

Total Budget Used (dollars):
newBuildTotalCost + oldDemoTotalCost = 14998000

Part 3: Max Tax Revenue generated (dollars):
taxRevenue = 199000
```

6

2 - *Question 2* (Version 6)

3 - *Question 3* (Version 2)

4 - *Question 4* (Version 3)

5 - *Question 5* (Version 2)