Homework 3 - Integer Programming

Adv. Analytics and Metaheuristics

Daniel Carpenter and Iker Zarandona

March 2022

Contents

1 - Problem 1

1.1 Mathematical Formulation

1.1.1 Sets

Set Name	Description
$GENERATORS$	Set of generators i that can be used (`A`,`B`,`C`)
$PERIODS$	2 possible periods p (`1`, `2`) in the production day

1.1.2 Parameters

Parameter Name	Description
S_i	Fixed cost to start a generator ($i \in GENERATORS$) in the entire day
F_i	Fixed cost to operate a generator ($i \in GENERATORS$) in any period
C_i	Variable cost per megawatt to operator a generator ($i \in GENERATORS$) in any period
U_i	Max. megawatts generated for generator ($i \in GENERATORS$) in any period
$demand_p$	Total demanded megawatts for period ($p \in PERIODS$)
M	Large constant to map watts used by each generator ($i \in GENERATORS$)

1.1.3 Decision Variables

Variable Name	Description
$watts_{i,p}$	*Integer variable*: Number of watts to produce per generator ($i \in GENERATORS$) per period ($p \in PERIODS$)
$x_{i,p}$	*Binary variable*: `1` if a generator ($i \in GENERATORS$) is in period p ($p \in PERIODS$), `0` if not turned on at all
y_i	*Binary variable*: `1` if a generator ($i \in GENERATORS$) is used, `0` if not turned on at all

1.1.4 Objective Function

$$minimize\,cost: \sum_{i \in GENERATORS} \Big((\textstyle\sum_{p \in PERIODS} (watts_{i,p}) \times C_i) + (F_i \times \textstyle\sum_{p \in PERIODS} x_{i,p}) + (S_i \times y_i) \Big)$$

1.1.5 Constraints

C1: For each period, meet the demanded megawatts

$$requiredWatts: \sum_{i \in GENERATORS} (watts_{i,p}) = demand_p, \forall\ p \in PERIODS$$

C2: For each generator, don't surpass the allowable megawatts

$$upperBound: \sum_{p \in PERIODS} (watts_{i,p}) \leq U_i, \forall\ i \in GENERATORS$$

C3: For each generator, map decision variables together to account for the fixed costs in a given day S_i

$$mapVars: \sum_{p \in PERIODS} (watts_{i,p}) \leq M_i \times y_i, \forall\ i \in GENERATORS$$

C4: For each generator and period, map decision variables y and $watts$ together to account for the fixed costs in a per period p

$$mapVars2: watts_{i,p} \leq M_i \times x_{i,p}, \forall\ i \in GENERATORS,\ p \in PERIODS$$

C5 Non-negativity or Binary restraints of decision variables

$$watts_{i,p} \geq 0$$

$$x_{i,p},\ y_i \in (0,1)$$

1.2 Code and Output

1.2.1 Code

group23_HW3_p1.mod

```
# Homework 3 - Integer Programming
# Adv. Analytics and Metaheuristics
# Daniel Carpenter and Iker Zarandona
# March 2022
# Problem 1

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

# SETS ============================================================
    set GENERATORS; # Set of generators to use
    set PERIODS;    # Periods in the day

# PARAMETERS ======================================================
    param S {GENERATORS}   >= 0; # Fixed cost to start
    param F {GENERATORS}   >= 0; # Fixed cost to operate
    param C {GENERATORS}   >= 0; # Variable cost per megawatt
    param U {GENERATORS}   >= 0; # Upper bound on megawatts in a day
    param M {GENERATORS}   >= 0; # Map decision variables
    param demand {PERIODS} >= 0; # Megawatts required per period

# DECISION VARIABLES ==============================================
    var watts {GENERATORS, PERIODS} >= 0 integer; # Megawatts to use
    var x {GENERATORS, PERIODS} binary; # Map to watts for fixed daily costs
    var y {GENERATORS} binary; # Map to watts for fixed daily costs

# OBJECTIVE FUNCTION ==============================================
  minimize cost:
      (sum{i in GENERATORS} (sum{p in PERIODS} watts[i,p])*C[i])
    + (sum{i in GENERATORS} F[i]*sum{p in PERIODS}x[i,p])
    + (sum{i in GENERATORS} S[i]*y[i]);

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

    # C1: For each period, meet the demanded megawatta
    subject to requiredWatts {p in PERIODS}:
        (sum{i in GENERATORS} watts[i,p]) = demand[p];

    # C2: For each generator, don't surpass the allowable megawatts
    subject to upperBound {i in GENERATORS}:
        (sum{p in PERIODS} watts[i,p]) <= U[i];

    # C3: For each generator, map decision variables together to account for the fixed costs in a given day Si
    subject to mapVars {i in GENERATORS}:
        (sum{p in PERIODS} watts[i,p]) <= M[i] * y[i];

    # C4: For each generator and period, map decision variables y and watts together to account for the fixed costs in a per period p
    subject to mapVars2 {i in GENERATORS, p in PERIODS}:
        watts[i,p] <= M[i] * x[i,p];

# CONTROLS ========================================================
    data group23_HW3_p1.dat;
    solve;

    print;
    print "Which generators are used?";
    display y;

    print "Which periods were the generators used?";
    display x;

    print "Optimal Amount of Megawatts for each generator and period:";
    display watts;
```

group23_HW3_p1.dat

```
# Homework 3 - Integer Programming
# Adv. Analytics and Metaheuristics
# Daniel Carpenter and Iker Zarandona
# March 2022
# Problem 1

# SETS ============================================================
    set GENERATORS  := A B C;   # Set of generators to use
    set PERIODS     := 1 2;     # Periods in the day


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

    # S: Fixed cost to start a generator (i ∈ GENERATORS) in the entire day
    # F: Fixed cost to operate a generator (i ∈ GENERATORS) in any period
    # C: Variable cost per megawatt to operator a generator (i ∈ GENERATORS) in any period
    # U: Max. megawatts generated for generator (i ∈ GENERATORS) in any period
    # M: Value to map watts used by each generator (i ∈ GENERATORS)
    #    Set to be slighlty over the max megawatts per day
    param: S     F    C     U     M :=
        A  3000  700  5.00  2100  2200
        B  2000  500  4.00  1800  1900
        C  1000  900  7.00  3000  3100
        ;

    # Total demanded megawatts for period (p ∈ PERIODS)
    param demand :=
        1   2900
        2   3900
        ;
```

1.2.2 Output

```
ampl: model 'C:\Users\daniel.carpenter\OneDrive - the Chickasaw
CPLEX 20.1.0.0: optimal integer solution; objective 46100
7 MIP simplex iterations
0 branch-and-bound nodes

Which generators are used?
y [*] :=
A  1
B  1
C  1
;

Which periods were the generators used?
x :=
A 1   0
A 2   1
B 1   0
B 2   1
C 1   1
C 2   0
;

Optimal Amount of Megawatts for each generator and period:
watts :=
A 1      0
A 2   2100
B 1      0
B 2   1800
C 1   2900
C 2      0
;
```

1.2.2.1 Analysis of the Output

- The minimized cost is $\$46,100$
- Generator A, B, and C run
- Generator C runs in period 1. Generator A and B run in period 2
- Generator A produces $2,100$ megawatts in total
- Generator B produces $1,800$ megawatts in total
- Generator C produces $2,900$ megawatts in total

2 - Problem 2 (a)

2.1 Mathematical Formulation

2.1.1 Sets

Set Name	Description
$PRODUCTS$	5 types of landscaping and construction products (e.g., cement, sand, etc.) labeled product (p) $A, B, C, D,$ and E
$SILOS$	8 different silos s that each product must be stored in $(1, 2, \ldots, 8)$

2.1.2 Parameters

Parameter Name	Description
$cost_{s,p}$	Cost of storing *one ton* of product $p \in PRODUCTS$ in silo $s \in SILOS$
$supply_p$	Total supply *in tons* available of product $p \in PRODUCTS$
$capacity_s$	Total capacity *in tons* of silo $s \in SILOS$. Can store products.
M	Variable to map *decision variable* $tonsOfProduct_{p,s}$ to $isStored_{p,s}$. Uses big M method.

2.1.3 Decision Variables

Variable Name	Description
$tonsOfProduct_{p,s}$	*Tons* of product $p \in PRODUCTS$ to store in silo $s \in SILOS$. Non-negative.
$isStored_{p,s}$	*Binary variable* indicating if product $p \in PRODUCTS$ is stored in silo $s \in SILOS$.

2.1.4 Objective Function

$$minimize\ costOfStorage : \sum_{p \in PRODUCTS} \sum_{s \in SILOS} tonsOfProduct_{p,s} \times cost_{p,s}$$

2.1.5 Constraints

C1: For each silo s, the *tons* of the supplied product p must be less than or equal to the capacity limit of silo s

$$meetCapacity : \sum_{p \in PRODUCTS} tonsOfProduct_{p,s} \leq capacity_s, \; \forall \; s \in SILOS$$

C2: For each product p, must use all of the total product that is available

$$useAllProduct : \sum_{s \in SILOS} tonsOfProduct_{p,s} = supply_p, \; \forall \; p \in PRODUCTS$$

C3: For each silo s and product p,

$$oneProductInSilo : \sum_{pinPRODUCTS} isStored_{p,s} = 1, \; \forall \; s \in SILOS$$

C4: Map the decision variables together using the Big M method

$$mapVars : tonsOfProduct_{p,s} \leq M \times isStored_{p,s}, \; \forall \; p \in PRODUCTS, \; \forall \; s \in SILOS$$

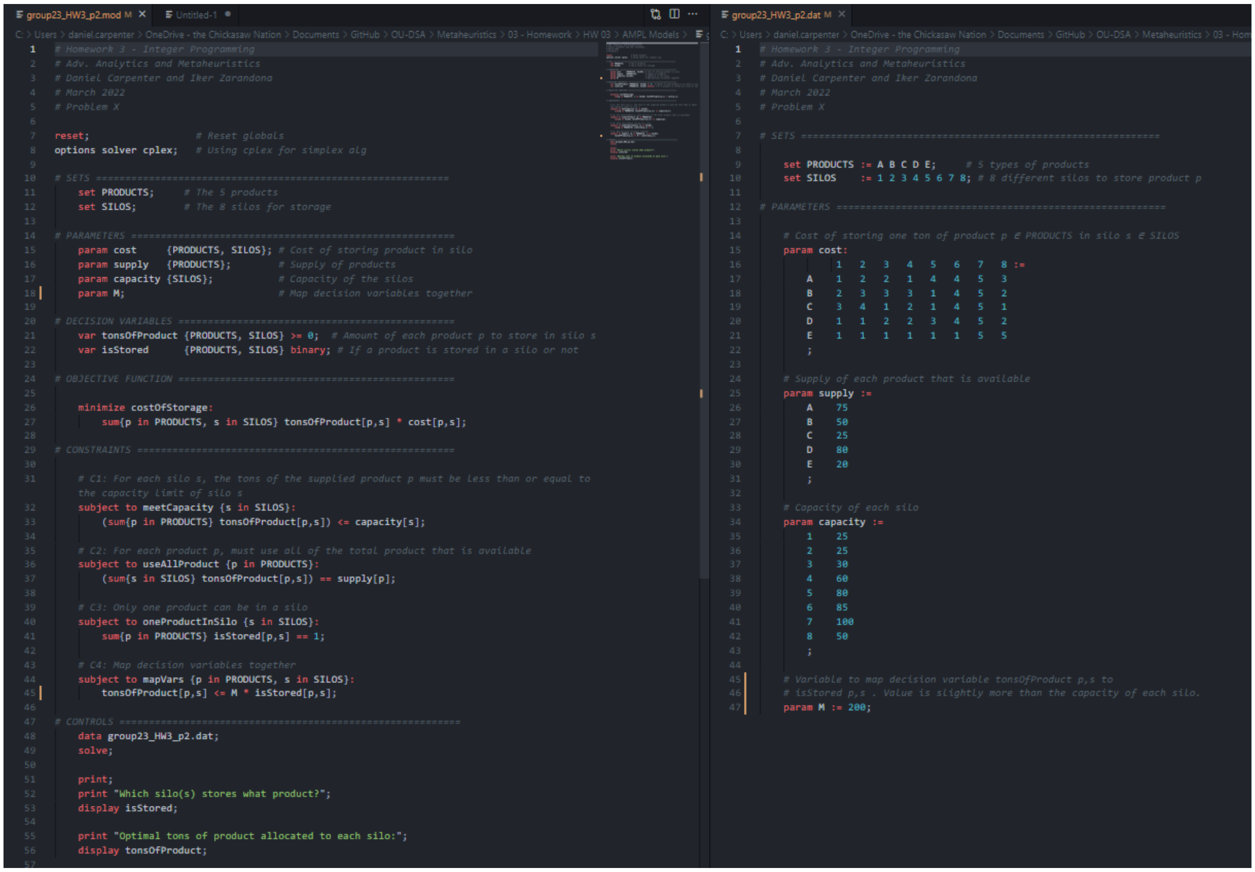
C5 Non-negativity or Binary restraints of decision variables

$$tonsOfProduct_{p,s} \geq 0$$

$$isStored_{p,s} \in (0, 1)$$

2.2 Code and Output

2.2.1 Code



group23_HW3_p2.mod

```
# Homework 3 - Integer Programming
# Adv. Analytics and Metaheuristics
# Daniel Carpenter and Iker Zarandona
# March 2022
# Problem X

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

# SETS ============================================================
    set PRODUCTS;     # The 5 products
    set SILOS;        # The 8 silos for storage

# PARAMETERS ========================================================
    param cost     {PRODUCTS, SILOS}; # Cost of storing product in silo
    param supply   {PRODUCTS};        # Supply of products
    param capacity {SILOS};           # Capacity of the silos
    param M;                          # Map decision variables together

# DECISION VARIABLES ===============================================
    var tonsOfProduct {PRODUCTS, SILOS} >= 0;  # Amount of each product p to store in silo s
    var isStored      {PRODUCTS, SILOS} binary; # If a product is stored in a silo or not

# OBJECTIVE FUNCTION ===============================================

    minimize costOfStorage:
        sum{p in PRODUCTS, s in SILOS} tonsOfProduct[p,s] * cost[p,s];

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

    # C1: For each silo s, the tons of the supplied product p must be less than or equal to
    the capacity limit of silo s
    subject to meetCapacity {s in SILOS}:
        (sum{p in PRODUCTS} tonsOfProduct[p,s]) <= capacity[s];

    # C2: For each product p, must use all of the total product that is available
    subject to useAllProduct {p in PRODUCTS}:
        (sum{s in SILOS} tonsOfProduct[p,s]) == supply[p];

    # C3: Only one product can be in a silo
    subject to oneProductInSilo {s in SILOS}:
        sum{p in PRODUCTS} isStored[p,s] == 1;

    # C4: Map decision variables together
    subject to mapVars {p in PRODUCTS, s in SILOS}:
        tonsOfProduct[p,s] <= M * isStored[p,s];

# CONTROLS =========================================================
    data group23_HW3_p2.dat;
    solve;

    print;
    print "Which silo(s) stores what product?";
    display isStored;

    print "Optimal tons of product allocated to each silo:";
    display tonsOfProduct;
```

group23_HW3_p2.dat

```
# Homework 3 - Integer Programming
# Adv. Analytics and Metaheuristics
# Daniel Carpenter and Iker Zarandona
# March 2022
# Problem X

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

    set PRODUCTS := A B C D E;      # 5 types of products
    set SILOS    := 1 2 3 4 5 6 7 8; # 8 different silos to store product p

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

    # Cost of storing one ton of product p ∈ PRODUCTS in silo s ∈ SILOS
    param cost:
            1   2   3   4   5   6   7   8 :=
        A   1   2   2   1   4   4   5   3
        B   2   3   3   3   1   4   5   2
        C   3   4   1   2   1   4   5   1
        D   1   1   2   2   3   4   5   2
        E   1   1   1   1   1   1   5   5
        ;

    # Supply of each product that is available
    param supply :=
        A   75
        B   50
        C   25
        D   80
        E   20
        ;

    # Capacity of each silo
    param capacity :=
        1   25
        2   25
        3   30
        4   60
        5   80
        6   85
        7   100
        8   50
        ;

    # Variable to map decision variable tonsOfProduct p,s to
    # isStored p,s . Value is slightly more than the capacity of each silo.
    param M := 200;
```

2.2.2 Output

```
ampl: model 'C:\Users\daniel.carpenter\OneDrive - the Chickasaw Nation\D
CPLEX 20.1.0.0: optimal integer solution; objective 320
48 MIP simplex iterations
0 branch-and-bound nodes

Which silo(s) stores what product?
isStored [*,*] (tr)
:   A   B   C   D   E    :=
1   1   0   0   0   0
2   0   0   0   1   0
3   0   0   1   0   0
4   1   0   0   0   0
5   0   1   0   0   0
6   0   0   0   0   1
7   0   0   0   1   0
8   0   0   0   1   0
;

Optimal tons of product allocated to each silo:
tonsOfProduct [*,*] (tr)
:    A    B    C    D    E     :=
1   25    0    0    0    0
2    0    0    0   25    0
3    0    0   25    0    0
4   50    0    0    0    0
5    0   50    0    0    0
6    0    0    0    0   20
7    0    0    0    5    0
8    0    0    0   50    0
;
```

2.2.2.1 Analysis of the Output

- Minimized loading cost for 250 tons of 5 products over the 8 silos is 320 (problem does not state cost units).
- Product A stores 25 $tons$ in $silo$ 1 and 50 $tons$ in $silo$ 4
- Product B stores 50 $tons$ in $silo$ 5
- Product C stores 25 $tons$ in $silo$ 3
- Product D stores 25 $tons$ in $silo$ 2, 5$tons$ in $silo$ 7, and and 50 $tons$ in $silo$ 8
- Product E stores 20 $tons$ in $silo$ 6

3 - Problem 2 `(b)`

Additional constraint so that no tanks can be partially filled

Add the decision variable z_s which will allow be used to either fill a silo $s \in SILOS$ or not fill the silo s using two new constraints.

3.1 New Constraints

C6 Possibility 1: For each silo s, its capacity must be at 100%

$$mustBeFull: \sum_{p \in PRODUCTS} tonsOfProduct_{p,s} = capacity_s + (M \times z_s), \ \forall \ s \in SILOS$$

C7 Possibility 2: For each silo s, the capacity must not be utilized (0%).

$$mustBeEmpty: \sum_{p \in PRODUCTS} tonsOfProduct_{p,s} = 0 + [M \times (1 - z_s)], \ \forall \ s \in SILOS$$

C9 For each silo, We must store at 0% or 100%

$$onlyOneConstraint: z[s] = 1;, \ \forall \ s \in SILOS$$

3.2 Code and Output