

Exam 1  
Adv. Analytics and Metaheuristics

Daniel Carpenter

March 2022

## Contents

<b>1 - Problem 1</b>	<b>2</b>
1.1 Mathematical Formulation . . . . .	2
1.2 Code and Output . . . . .	4
<b>2 - Problem 2</b>	<b>5</b>
2.1 Additions to Model . . . . .	5
2.2 Code and Output . . . . .	6

# 1 - Problem 1

## 1.1 Mathematical Formulation

### 1.1.1 Sets

Set Name	Description
$P$	The three types of products, High-Gloss, Semi-Gloss, and Flat

### 1.1.2 Parameters

Parameter Name	Description
$rawA_p$	The amount of raw ingredient $A$ needed to produce product $p \in P$
$rawB_p$	The amount of raw ingredient $B$ needed to produce product $p \in P$
$demand_p$	The minimum demand to be met for product $p \in P$
$profit_p$	The associated profit for product $p \in P$

### 1.1.3 Decision Variables

Variable Name	Description
$amtToProduce_p$	The amount of product $p \in P$ to produce and is ion the set of integers

### 1.1.4 Objective Function

$$\text{maximize theProfit} : \sum_p amtToProduce_p \times profit_p$$

### 1.1.5 Constraints

**C1:** Meet the minimum demand for each product

$$meetMinDemand : amtToProduce_p \geq demand_p, \forall p \in P$$

**C2:** Cannot exceed the supply of Raw Material A

$$rawSupplyA : \sum_p amtToProduce_p \times rawA_p \leq 4,000$$

**C3:** Cannot exceed the supply of Raw Material B

$$rawSupplyB : \sum_p amtToProduce_p \times rawB_p \leq 6,000$$

**C4:** Ratio of 3:2 for High and Semi Gloss, respectively

- Since  $\frac{3}{2} = 1.5$ , the amount of high gloss produced must always be  $1.5 \times$  semi gloss

$$highToSemiRatio : 1.5 \times amtToProduce_{Semi \in P} = amtToProduce_{High \in P}$$

**C5:** Non-Negativity Constraints and is Integer

$$amtToProduce \geq 0, \in \mathbb{Z}$$

## 1.2 Code and Output

### 1.2.1 Code

```
problem1.mod u x
C:\Users\daniel.carpenter> OneDrive - the Chickasaw Nation > Documents > GitHub > OU-DSA > Metaheuristics > 04 - Exams > Exam 1
1 # Daniel, Carpenter
2 # Exam 1
3 # Problem 1
4
5 reset; # Reset globals
6 options solver cplex; # Using cplex for simplex alg
7
8 # SETS =====
9 set P circular; # The three types of products, High-Gloss, Semi-Gloss, and Flat
10
11 # PARAMETERS =====
12 param rawA {P} >= 0; # Raw ingredient A needed to produce product p in P;
13 param rawB {P} >= 0; # Raw ingredient B needed to produce product p in P;
14 param demand {P} >= 0; # Min demand to be met for product p in P;
15 param profit {P} >= 0; # Associated profit for product p in P;
16
17 # DECISION VARIABLES =====
18 var amtToProduce {P} >= 0 integer; # amount of product p in P to produce
19
20 # OBJECTIVE FUNCTION =====
21 maximize theProfit: sum{p in P} amtToProduce[p] * profit[p];
22
23 # CONSTRAINTS =====
24
25 # C1: Meet the minimum demand for each product
26 s.t. meetMinDemand {p in P}: amtToProduce[p] >= demand[p];
27
28 # C2/3: Cannot exceed the supply of Raw Material A or B
29 s.t. rawSupplyA: sum{p in P} amtToProduce[p] * rawA[p] <= 4000;
30 s.t. rawSupplyB: sum{p in P} amtToProduce[p] * rawB[p] <= 6000;
31
32 # C4: Ratio of 3:2 for High and Semi Gloss, respectively
33 s.t. highToSemiRatio {p in P}: 1.5 * amtToProduce[high] == amtToProduce[semi];
34
35 # CONTROLS =====
36 data problem1.dat;
37 solve;
38
39 print;
40 print "For each product, product the following amounts:";
41 display amtToProduce;
42
```

```
problem1.dat u x
C:\Users\daniel.carpenter> OneDrive - the Chickasaw Nation > Documents > GitHub > OU-DSA > Metaheuristics > 04
1 # Daniel, Carpenter
2 # Exam 1
3 # Problem 1
4
5 data;
6
7 # The three types of products, High-Gloss, Semi-Gloss, and Flat
8 set P := High Flat Semi;
9
10 # rawA The amount of raw ingredient A needed to produce product p in P;
11 # rawB The amount of raw ingredient B needed to produce product p in P;
12 # demand The minimum demand to be met for product p in P;
13 # profit The associated profit for product p in P;
14 param:
15 | | | rawA | rawB | demand | profit :=
16 | High | 2 | 4 | 200 | 30 |
17 | Semi | 3 | 2 | 200 | 20 |
18 | Flat | 5 | 7 | 150 | 50 |
19 ;
```

### 1.2.2 Output

- Not High to Semi is a 3:2 ratio and all demand and supply constraints are satisfied.

```
ampl: model 'C:\Users\daniel.carpenter\OneDrive - the Chickasaw Nation > Documents > GitHub > OU-DSA > Metaheuristics > 04 - Exams > Exam 1 > problem1.mod';
CPLEX 20.1.0.0: optimal integer solution; objective 42700
4 MIP simplex iterations
0 branch-and-bound nodes

For each product, product the following amounts:
amtToProduce [*] :=
High 810
Flat 152
Semi 540
;
```

## 2 - Problem 2

### 2.1 Additions to Model

Please assume that the mathematical formulation in the course videos are present as well (see code), just named objects are named differently but hopefully are convenient to interpret

#### 2.1.1 Parameters

Name	Description
$M$	“Big M,” which is a large scaler used to help model disjunctive constraints

#### 2.1.2 New Decision Vars

Name	Description
$z$	Determines which constraint to activate.
$totalProduction$	Simple variable that is Zappers + Spacerays

#### 2.1.3 New Constraints

Description	Constraint
Set Total Production to Zappers + Spacerays	$setTotalProd : totalProduction = spaceRays + zappers$
If total production is $\geq 400$	$isGreaterThanOr400 : totalProduction \geq 400 + M \times z$
then zappers must be 70% of total	$zappersAre70Perc : zappers \geq 0.70 \times totalProduction + M \times z$
If total Production is $\leq 400$	$isLessThanOr400 : totalProduction \leq 400 + M \times (1 - z)$
then no zappers at all	$noZappers : spaceRays \geq totalProduction - M \times (1 - z)$

## 2.2 Code and Output

### 2.2.1 Code

```
problem2.mod M X
C: > Users > daniel.carpenter > OneDrive - the Chickasaw Nation > Documents > GitHub > OU-DSA > Metaheuristics > 04 - Exam

1  # Daniel Carpenter
2  # Exam 1
3  # Problem 2
4
5  reset;
6
7  # Set up options and the solver
8  option solver cplex;
9
10 # PARAMETERS -----
11 param M := 100000000; # Big M scaler
12
13 # DECISION VARIABLES -----
14 var spaceRays >= 0; # Number of Space Ray Products
15 var zappers >= 0; # Number of Zapper Products
16 var totalProduction >= 0; # Zappers + Spacerays
17 var z binary; # Determines whether or not constraints are active or not.
18
19 # OBJECTIVE -----
20 maximize profit: (8*spaceRays) + (5*zappers);
21
22 # CONSTRAINTS -----
23 s.t. plastic: (2*spaceRays) + (1*zappers) <= 1000;
24 s.t. labor: (3*spaceRays) + (4*zappers) <= 2400;
25 s.t. production: (spaceRays) + (zappers) <= 700;
26 s.t. management: (spaceRays) - (zappers) <= 350;
27
28
29 # New Constraints
30
31 ## Set totalProduction = Zappers + Spacerays
32 s.t. setTotalProd: totalProduction == spaceRays + zappers;
33
34 ## If total Production is >= 400,
35 s.t. isGreaterThan400: totalProduction >= 400 + M*z;
36 s.t. zappersAre70Perc: zappers >= 0.70*totalProduction + M*z;
37
38 ## If total Production is <= 400, then no zappers
39 s.t. isLessThan400: totalProduction <= 400 + M*(1 - z);
40 s.t. noZappers: spaceRays >= totalProduction - M*(1 - z);
41
42 # SOLVE -----
43
44 ## Solve the model
45 solve;
46
47 print;
48 print 'Produce this amount of Space Rays and Zappers';
49 display spaceRays,zappers, totalProduction;
50
51 print 'If Zappers used, then must be 70% of total prod:';
52 display zappers / totalProduction;
53
```

### 2.2.2 Output

- Optimal solution included more than 400, so zappers are 70% of production

```
ampl: model 'C:\Users\daniel.carpenter\OneDrive - the Chi
CPLEX 20.1.0.0: sensitivity
CPLEX 20.1.0.0: optimal solution; objective 3827.027027
3 dual simplex iterations (1 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;

Produce this amount of Space Rays and Zappers
spaceRays = 194.595
zappers = 454.054
totalProduction = 648.649

If Zappers used, then must be 70% of total prod:
zappers/totalProduction = 0.7
```