Name: Abdulrahman Alturki

EBGN645 Computational Economics

HW#1/Part A

Q1. BennyBakery Profit Maximization (Code is provided through Github)

Part a:

```
* BennyBakery Profit Maximization With Two Scenarios

Set i / roll, croissant, bread/;

Parameters

Rev(i) "--$/item-- revenue per unit sold"/ roll 2.25, croissant 5.5, bread 10/

C(i)   "--$/item-- cost per unit produced"/ roll 1.5, croissant 2, bread 5/

H(i)   "--hours/item-- time per unit sold"/ roll 1.5, croissant 2.25, bread 5/;

Scalar  Hbar "hours limit" /40/;

parameter X_base(i),X_meal(i);

scalar profit_base, profit_meal;

scalar sw_meal "Base=0, and roll_constraints=1" / 1 /;

Positive Variable  X(i);

Variable  Profit;

Equations Obj, Hours, Meal;

*Total profit is calcuated in $

Obj..   Profit=e= sum(i,(Rev(i)- C(i)) * X(i));

*Total hours is calcuated in hrs

Hours.. sum(i, H(i) * X(i)) =l= Hbar;
```

Profit can be written as follows:

- Profit= ∑(Rev(i)-C(i)) * X(i), for each item selected by the model.

Total hours can be written in math as follows:

- Hours= ∑(i,H(i) * X(i)), for each item selected by the model.

Part b:

* Counterfactual Rule: Roll to be sold with every croissant and Rolls can still be sold individually.

Meal.. X('croissant') =l= X('roll') + (1-sw_meal)*(x('croissant') - x('roll')) ;

This constraint can be written in match as follows:

Meal= X('croissant') ≤ X('roll') + (1-sw_meal)*(x('croissant') - x('roll')) ;

Part c:

If sw_meal=1, constraints will be applicable, and equation will be Meal= X('croissant') ≤ X('roll'). However, if sw_meal=0, then X('roll') will be eliminated and the equation will be = X('croissant') ≤ X('croissant'), which means no constraint.

	Rev ($/item)	Cost ($/item)	Hour (hour/item)	Profit/Hour Ratio
roll	2.25	1.5	1.5	0.5
croissant	5.5	2	2.25	1.56
bread	10	5	5	1

- Sw_meal=0 (no meal constraints): Model will choose the highest rate (croissant), with total items chosen of (40/2.25) 17.78. Thus, total profit= 17.78 item * (5.5-2) $/item= $ 62.22.
- Sw-meal=1 (X('croissant') ≤ X('roll')): Since roll has less profit than croissant, the model will choose similar number of items for both, being still within the constraints. Total hours of roll & croissant equal 1.5 hour/item + 2.25 hours/item = 3.75 hours/item. Total items chosen equal 40 hour/3.75 hours per item= 10.67 items. Thus, Profit equals 10.67 items * (2.25-1.5)+(5.5-2)) $/item = $ 45.3.

  In summary, applying more strict constraints usually decreases the profit.

Q.2

Part a:

* Jellybean factory optimization with two machines (X1 and X2) and five different types.

Set b "beans/colors" /yellow, blue, green, orange, purple /;

Set m "machines" / x1,x2 /;

alias (b,bb);

Parameters

Rev(b) "--$/item-- revenue per unit sold"/ yellow 1, blue 1.05, green 1.07, orange 0.95, purple 0.90/;

Scalars

rate "--quantity/machines/hour-- production rate/hour/machine"/ 100 /

hours "--hours/week-- total hours per week"/ 40 /

Capacity "--quantity/week-- weekly production rate"

Threshold "allowed deviation" / 0.05 /;

Capacity = rate * hours * 2;

* Capacity is multiplied by 2 because we have 2 machines

Positive Variables y(b);

* z unit is $ as it shows the total profit

Variable z;

equations

Objective, CapConstraint,

eq_prodlimit_upper(b,bb),

eq_prodlimit_lower(b,bb);

Objective.. z=e= sum(b, Rev(b) * y(b));

CapConstraint.. sum(b, Y(b)) =l= Capacity;

part b:

This is a case without any constraints, in which the model will choose the highest profit color while keeping the maximum capacity of weekly quantity. Thus, the model will choose Green color ($ 1.07/bean) and 8,000 bean/week. Total profit equals $ 1.07/bean * 8000 bean/week = $ 8,560/bean.

Part c:

In this scenario, the model will choose all five colors with keeping 5% maximum difference between any two colors. This means the highest quantity of most profitable color should not exceed 5% of the least profitable color. In addition, total weekly quantity produced should not exceed 8,000 beans. The model shows a drop in total profit to $ 7,962.72/weak.

Part d:

For this scenario, each machine will choose the highest profitable color, not exceeding 4,000 beans/week. Thus, first machine will choose green color, and second machine will choose yellow color. The math will be done as follows:

Profit= $ 1.07/green bean * 4,000 beans + $ 1.00/yellow bean * 4,000 beans = $ (4,280+4,000) = $ 8,280