

Diet Problem Solution

February 21, 2020

1 Problem

For the following problems, provide the "Essential Components of an Optimization Solution" (objective in words, variable definition, algebraic formulation (using summation notation and with labels for the objective function and constraints), summary of results). For your implementation, implement and solve the problem using Python/Pyomo and turn in your code (.py or .ipynb).

- (Albright and Winston, 13.30). Maggie Stewart loves desserts, but due to weight and cholesterol concerns, she has decided that she must plan her desserts carefully. There are two possible desserts she is considering: snack bars and ice cream. After reading the nutrition labels on the snack bar and ice cream packages, she learns that each serving of a snack bar weighs 37 grams and contains 120 calories and 5 grams of fat. Each serving of ice cream weighs 65 grams and contains 160 calories and 10 grams of fat. Maggie will allow herself no more than 450 calories and 25 grams of fat in her daily desserts, but because she loves desserts so much, she requires at least 120 grams of dessert per day. Also, she assigns a "taste index" to each gram of each dessert, where 0 is the lowest and 100 is the highest. She assigns a taste index of 95 to ice cream and 85 to snack bars (because she prefers ice cream to snack bars).

2 Data

1. Let $D = \{\text{snack bars}, \text{ice cream}\}$ be the set of desserts,
2. taste_i = taste index for dessert i , $i \in D$
3. weight_i = weight of dessert i , $i \in D$
4. calories_i = calories for dessert i , $i \in D$
5. fat_i = fat for dessert i , $i \in D$
6. weightReq = minimum weight (grams) of dessert required
7. caloriesAllowed_i = maximum calories allowed per day
8. fatAllowed_i = maximum grams of fat required per day

3 Objective in Words

Decide how much of each of each dessert to consume per day so that taste index is maximized and calories and grams of fat are minimized, subject to the following constraints:

- Total calories do not exceed 450
- Total grams of fat do not exceed 25

- Each day requires 120 grams (weight) of desserts

4 Algebraic Formulation

Decision Variables:

Let x_i = the # servings of dessert i per day, for $i \in D$.

Objectives and constraints:

$$\max \sum_{i \in D} \text{weight}_i \text{taste}_i x_i \quad \begin{array}{l} \text{(Servings)} \\ \text{(taste_servings)} \end{array}$$

$$\text{s.t. } \sum_{i \in D} \text{weight}_i x_i \geq \text{weightReq} \quad \text{(Weight Constraint)}$$

$$\sum_{i \in D} \text{calories}_i x_i \leq \text{caloriesAllowed} \quad \text{(Calorie Constraint)}$$

$$\sum_{i \in D} \text{fat}_i x_i \geq \text{fatAllowed} \quad \text{(Grams of Fat Constraint)}$$

$$x_i \leq \text{demand}_i, i \in D \quad \text{(Demand)}$$

$$x_i \geq 0, i \in D \quad \text{(Nonnegativity Constraint)}$$

5 Implementation

See attached file, HW_Dessert.ipynb, for the implementation and solution of the model using Pyomo and GLPK in Python.

6 Results and Interpretation

The optimal solution is for Maggie Stewart to eat 1.25 snack bars and 1.875 servings of ice cream each day to get the optimal objective function taste index of 15509.375.

Manufacturing Problem Solution

February 21, 2020

1 Problem

For the following problems, provide the "Essential Components of an Optimization Solution" (objective in words, variable definition, algebraic formulation (using summation notation and with labels for the objective function and constraints), summary of results). For your implementation, implement and solve the problem using Python/Pyomo and turn in your code (.py or .ipynb).

- (Albright and Winston, 13.33). A manufacturing company makes two products. Each product can be made on either of two machines. The time (in hours) required to make each product on each machine is listed in the file P13_33.xlsx. Each month, 500 hours of time are available on each machine. Each month, customers are willing to buy up to the quantities of each product at the prices also given in the same file. The company's goal is to maximize the revenue from selling units during the next two months.

2 Data:

Times (hours) needed to product each product on each machine

	Machine 1	Machine 2
Product 1	0.4	0.3
Product 2	0.7	0.4

Product	Demands		Prices	
	Month 1	Month 2	Month 1	Month 2
Product 1	1000	1900	\$55	\$12
Product 2	1400	1300	\$65	\$32

Let

1. $P = \{\text{Product 1, Product 2}\}$ be the set of products
2. $M = \{\text{Machine 1, Machine 2}\}$ be the set of machine hours for machine j , $j \in M$
3. $T = \{\text{Month 1, Month 2}\}$ be month k , $k \in T$
4. $price_{ik}$ = price (\$ per month) for product i , $i \in P$; month k , $k \in T$
5. $demand_{ik}$ = demand of product i , $i \in P$; month k , $k \in T$

6. $time_{ij}$ = amount of time (hours) it takes to produce of product i , with machine j ; $i \in P, j \in M$.

3 Objective in Words:

Decide how much of each product to produce on each machine, each month,

so that revenue is maximized and demand is met for the two months.

- 500 hours of time is available on each machine per month
- Demand is met for each month
- (At most 1000 units of product 1 are bought during month 1, At most 1900 units of product 1 are bought during month 2, At most 1400 units of product 2 are bought during month 1, At most 1300 units of product 2 are bought during month 2)

4 Algebraic Formulation

Decision Variables:

Let x_{ijk} = amount produced of product i , with machine j , in month k ; $i \in P, j \in M, k \in T$.

Objective and constraints:

$$\max \sum_{i \in P} \sum_{j \in M} \sum_{k \in T} x_{ijk} * price_{ik}, i \in P, k \in T \quad (\text{Profit})$$

$$\sum_{j \in M} x_{ijk} \geq demand_{ik}, i \in P, k \in T \quad (\text{Demand})$$

$$\sum_{i \in P} time_{ij} * x_{ijk} \leq 500, j \in M, k \in T \quad (\text{Hours})$$

5 Implementation

See attached file, Manufacturing.ipynb, for the implementation and solution of the model using Pyomo and GLPK.

6 Interpret the Results

The optimal solution is for Manufacturing Company to produce ____ of Product 1 and ____ of Product 2 and the optimal objective function value is \$____.