Exercicis de Modelització

Models i Mètodes de la Investigació Operativa (MMIO)

Pau Soler Valadés

September 15, 2024

La dieta

Donada la seguent taula d'aliments, el seu cost i els seu valor nutricional

Food Item	**Bread**	**Milk**	**Cheese**	**Potato**	**Fish**	**Yogurt**
Cost	2.0	3.5	8.0	1.5	11.0	1.0
Protein, g	4.0	8.0	7.0	1.3	8.0	9.2
Fat, g	1.0	5.0	9.0	0.1	7.0	1.0
Carbohydrates, g	15.0	11.7	0.4	22.6	0.0	17.0
Calories	90	120	106	97	130	180

Troba el conjunt d'elements més econonòmic tal que compleixin

- Al menys 300 calories
- No més de 10 grams de proteïna
- Al menys 10 grams de carbohidrats
- Al menys 8 grams de greix
- Al menys 0.5 unitats de peix
- No més d'1 unitat de llet

Solució

Sempre que hem de modelitzar un problema hem de definir clarament les seves 3 parts principals:

1. Variables: de quina informació disposem al problema per poder modelitzar-lo. És la part més creativa, ja que un problema es pot resoldre diferent segons quines variables s'escullin.
2. Restriccions: quines restriccions hem d'imposar a les variables escollides.
3. Funció Objectiu: com interactuen les variables, i quines hem de maximitzar o minimitzar. Bàsicament, què volem aconseguir en aquest problema.

Variables

Definim:

1. $I = \{1, ..., n\}$ conjunt de menjar a escollir, sent i l'ítem. En el nostre problema, $n = 6$
2. $\forall i \in I, c_i$ és el cost associat al menjar i
3. $\forall i \in I, x_i$ és la quantitat en grams del menjar i
4. $J = \{1, ..., m\}$ conjunt de característiques extra de tots els aliments a comprovar. És a dir, 1 son els carbohidrats, 2 els greixos, 3 les calories... Per tant, $j \in J$ és un atribut.
5. És a dir, anomenem y_{ij} la quantitat de j que té l'ítem i. Per exemple, el Pa (i=1) té 4 grams de proteïna, ergo $y_{11} = 4$, i les calories del formatge, serien $y_{34} = 106$

6. $R = \{r_1, ..., r_q\}$ el conjunt de restriccions, és a dir, què no es pot complir. Un $r \in R$ qualsevol conté una relació entre j i el total. Per exemple, menys de 300 grams de proteina, seria $r = \{1, <, 300\}$.

Funció Objectiu:
És clar que amb 1,2 i 3 ja podem definir la funció objectiu, que és minimitzar:

$$\sum_{i \in I} x_i c_i$$

Restriccions
Seguint la notació introduïda a les variables, tindriem:

1. $\sum_{i \in I} y_{i3} x_i \geq 300$ (calories)
2. $\sum_{i \in I} y_{i1} x_i \leq 10$ (proteïna)
3. $\sum_{i \in I} y_{i4} x_i \geq 10$ (carbohidrats)
4. $\sum_{i \in I} y_{i2} x_i \geq 8$ (greix)
5. $x_5 \geq 0.5$ (peix)
6. $x_2 \leq 1$ (llet)
7. $x_i \geq 0 \; \forall i \in I$ (no negativitat)

On dels ítems 1 al 6 es podrien reescriure com a elements de R, i el 7 fa no negativitat.

Mapa de Colors

Donat un mapa, troba el menor nombre de colors que permet pintar el mapa tal que no hi ha dos països contiguus amb el mateix color.

Parametres i Conjunts

1. $I = \{1, ..., n\}$ el nombre de països que està contingut pel mapa.
2. $I_i \subset I$, concretament, I_i són els veïns (països contigus) al país i
3. $J = \{1, ..., m\}$ són els colors disponibles.

Variables

1. Variable d'ús de colors,

$$y_j = \begin{cases} 1 \text{ si el color j està pintant algun país} \\ 0 \text{ altrament} \end{cases}$$

1. Variable binària de relació

$$x_{ij} = \begin{cases} 1 \text{ si el país i està pintat amb el color j} \\ 0 \text{ altrament} \end{cases}$$

Funció Objectiu
Volem minimitzar

$$\sum_{j \in J} y_j$$

Restriccions
1. S'ha d'usar com a mínim un color, no pots no pintar cap país:

$$\sum_{j \in J} y_j \geq 1$$

2. Un país ha d'estar pintat per només un únic color:

$$\sum_{j \in J} x_{ij} = 1$$

3. Donat el país i, si aquest està pintat amb el color j, aquest no pot estar a cap dels seus veïns:

$$j \in J, k \in I_i, x_{kj} + x_{ij} \leq 1$$

Aquesta última és la restricció complicada de deduïr. Hi ha dos detalls lògics importants:

1. Que la condició compleixi per a tot veí de i no és $\sum_{k \in I_i}$ ... sinó que imposant les condicions dos a dos ja funciona, és a dir: $\forall k \in I_i ...$, per tant és lineal.
2. Ha de ser menor o igual que 1, dient-nos que si passa una, no passa l'altre i que quan passen les dues és impossible, com aquesta taula de veritat:

x_{kj}	x_{ij}	**Suma**	**Interpretació**
0	0	0	No es fa servier el color j ni a i ni als veïns (Factible)
0	1	1	El país i té color j i el veí k tampoc (Factible)
1	0	1	El veí k té el color j i el país i no (Factible)
1	1	2	Ambdós països pintats amb color j (Impossible)

Vigilant de Museu

Pensar i escriure

Mínim cost en un graf dirigit

Donat un graf dirigit amb demanda (un conjunt d'usuaris volen anar d'un punt a un altre), costos (com de llarg és el camí) i capacitat (quants usuaris poden anar a la carretera en un moment determinat), troba el camí amb cost mínim per enviar flux per la xarxa.

Variables

Definim:

- $G = (N, A)$ graf dirigit
- $N = \{1, ..., n\}$ nodes
- $A = \{(i, j) \mid i, j \in N\}$ vertexs
- Demanda: $b \in \mathbb{R}^m \ \forall i \in N$
- Cost: $c \in \mathbb{R}^m \ \forall (i, j) \in A \ c_{ij}$
- Capacitat: $d \in \mathbb{R}^m$ capacity $(i, j) \in A$
- Variable x_{ij}: flux pel vertex que uneix els nodes i, j

Funció Objectiu

Volem trobar el mínim cost d'enviar el flux per la xarxa, per tant:

$$\min \sum_{i,j \in A} c_{ij} x_{ij}$$

Restriccions

1. Conservació del flux: per a un node qualsevol $i \in N$, tot el flux que li ha arribat quan s'ha acabat el problema, ha de ser la seva demanda. Això s'escriu tal que així:

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ji} = b_i \quad \forall i \in N$$

Concretament, $\sum_{\{j:(i,j)\in A\}} x_{ij}$ és tot el flux dels veïns que apunten a i, i $\sum_{\{j:(j,i)\in A\}} x_{ji}$ és tot el flux que surt del node i. La resta d'aquests dos termes ha de ser la seva demanda b_i. És a dir, si és un node de pas ($b_i = 0$), el flux d'entrada és igual al de sortida, però si és un destí o un origen, les quantitats no seran iguals.

2. Capacitat i no negativitat: no pots enviar més flux en un vertex que la seva capacitat.

$$0 \leq x_{ij} \leq d_{ij} \quad \forall (i,j) \in A$$

3. Suma de demandes: s'imposa que s'ha de satisfer tota la demanda.

$$\sum_{i\in N} b_i = 0$$

Màxim flux en un graf dirigit

Donat un graf dirigit amb les característiques del problema anterior, ara deteminem els nodes $s, t \in N$, sent s l'inici (*source*) i t el final (*taget*) que son l'inici i el final de la demanda. El problema ara és de maximització del flux del node s al node t mai superant les capacitats de la xarxa.

Variables

Definim G, N, A, d, x_{ij} exactament igual que el problema anterior, i fem notar que no necessitem ni c ni b, ja que el problema només parla de maximitzar el flux entre dos nodes, no de reduïr el cost

De segur, ens falta la variable flux, que denominarem v_{ij} com a volum. Concretament, el nostre problema, direm directament $v = v_{sn}$

Funció Objectiu

La funció objectiu és doncs, clarament, la maximització de v

$$\max v$$

Restriccions

Ara ja no estem fent servir tots els nodes, i el matís d'aquest problema està en la distinció del node d'inici i el node fi dels altres.

1. Restricció de capacitat: aquesta val per a tot vertex, mai es pot superar la capcitat d'un vèrtex.

$$0 \leq x_{ij} \leq d_{ij} \quad \forall (i,j) \in A$$

2. Igualtat de flux: tot el flux que entra en un node, ha de sortir d'aquest mateix, a menys que el node sigui s o t.

$$\sum_{\{j:(i,j)\in A\}} x_{ij} = \sum_{\{j:(j,i)\in A\}} x_{ij} \quad \forall i \in N \setminus \{s,t\}$$

3. Flux de sortida de s: tot el flux que surt d's és el maxim flux de la xarxa, ja que s és l'únic inici

$$\sum_{j:(s,j)\in A} x_{sj} = v$$

4. Flux d'entrada a t: tot el flux que arriba a t és el màxim flux de la xarxa, ja que només pot sortir per t.

$$\sum_{j:(j,t)\in A} x_{jt} = v$$

Ubicació d'hospitals

Donat un conjunt de potencials ubicacions d'hospitals, troba la millor permutació de P hospitals tal que la distància dels usuaris més llunyans de l'hospital sigui mínima. És a dir, el pitjor cas (la persona que està més lluny de l'hospital) ha de ser el menor possible.

Intuïció i Raonament

Obviament, la dificultat d'aquest problema està en que, sense imposar que sigui lineal, la funció objectiu intuïtiva és:

$$\min \max d_{ij} \forall i \in I$$

És a dir, el mínim de les distàncies dels usuaris que estan més lluny donat un hospital. No cal dir que això no és lineal.

Solució 1

Aquesta és la solució que se l'hi va acudir a l'autor i que conté una tècnica de modelització interessant, tot i que després es reescriu simplificada en la solució 2.

Conjunts i Paràmetres

1. $I = \{1, ..., n\}$ usuaris disponibles
2. $J = \{1, ..., m\}$ ubicacions de possibles hospitals
3. d_{ij} distancia entre l'usuari i i l'hospital j
4. P nombre d'hospitals que s'han d'obrir

Variables

1. Quins hospitals tenim oberts:

$$y_j = \begin{cases} 1 \text{ si l'hospital } j \text{ és obert} \\ 0 \text{ al contrari} \end{cases}$$

1. Quin usuari va a quin hospital:

$$Z_{ij} = \begin{cases} 1 \text{ si el client } i \text{ està assignat a l'hospital } j \\ 0 \text{ viceversa} \end{cases}$$

1. Indicador de l'usuari més llunyà donat un hospital:

$$X_{ji} = \begin{cases} 1 \text{ si l'usuari } i \text{ és el més llunyà de l'hospital } j \\ 0 \text{ altrament} \end{cases}$$

Funció Objectiu

Minimitza la distància màxima:

$$\min \sum_{j \in J} \sum_{i \in I} d_{ij} X_{ji}$$

Constraints

1. Hem d'obrir exactament P hospitals:

$$\sum_{j \in J} y_j = P$$

2. Cada usuari pot acudir a un sol hospital:

$$\sum_{j\in J} Z_{ij} = 1 \quad \forall i \in I$$

3. Un usuari només pot ser assignat a un hospital que estigui obert:

$$Z_{ij} \leq y_j \quad \forall i \in I, j \in J$$

4. Elecció de la màxima distància:

$$d_{kj}Z_{kj} \leq d_{ij}Z_{ij} + M\left(1 - X_{ji}\right) \quad \forall i, k \in I, j \in J$$

Aquesta restricció una "M", on aquesta és una constant suficientment gran, com per exemple la distància més gran entre un usuari i un hosptial. Quan $X_{ji} = 1$, l'equació ens assegura que per l'usuari i aquesta és la distància més llarga; quan és zero, mai ho pot ser perque la constant M ens assegura que qualsevol d_{kj} será com a mínim M vegades més petita. Aquest truc ens permet ser molt flexibles amb la distància, ja que el valor d'M es fixa d'avantmà.

5. Un usuari només pot ser el més llunyà per un hosptial si l'usuari està assignat a aquell hospital:

$$X_{ji} \leq Z_{ij} \quad \forall i \in I, j \in J$$

Solució millorada

La solució 1 és perfectament correcta, tot i que en aquest cas, n'existeix una altra molt més elegant, que passa per canviar la funció objectiu i eliminar una variable. Aleshores, deixant exactament els mateixos parametres que ja teniem, defninim:

Funció Objectiu

Com que seguim volent fer un maxim d'un mínim, escribim:

$$\min z$$

com a funció objectiu.

Variables

Només necessitem les variables y_{ij} i Z_{ij}

Restriccions

1. z ha de ser el valor més gran possible.

$$z \geq d_{ij}Z_{ij} \quad \forall i \in I, \forall j \in J$$

Això és equivalent a que la distància sigui màxima.

Les restriccions 1. 2. i 3. anteriors es mantenen exactament igual, i s'eliminen la 4 i la 5.

Common Mathematical Programming Tricks

Big-M Method

When we need to activate/deactivate constraints or model logical implications:

1. Basic form:

$$a \leq b + M(1 - x) \quad x \text{ binary}$$

1. For "if x = 1 then a <= b":

$$a \leq b + M(1 - x)$$

When x = 1, becomes a <= b When x = 0, constraint is relaxed

Either-Or Constraints

For modeling "either constraint A or constraint B must hold":

1. Example: either $a \leq 5$ or $b \leq 3$:

$$a \leq 5 + M_1 y$$

$$b \leq 3 + M_2(1 - y)$$

$$y \text{ binary}$$

Min/Max Linearization

To model z = min(x,y):

$$z \leq x$$

$$z \leq y$$

$$z \geq x - M(1 - w)$$

$$z \geq y - Mw$$

$$w \text{ binary}$$

Absolute Value

For modeling z = |x|:

$$x = p - n$$

$$z = p + n$$

$$p, n \geq 0$$

Where p represents the positive part and n the negative part of x

Fixed Costs

For problems with setup costs and variable production:

$$\text{production} \leq M \cdot y$$

$$\text{cost} = \text{fixed\_cost} \cdot y + \text{variable\_cost} \cdot \text{production}$$

$$y \text{ binary}$$

Notes

1. M should be a sufficiently large constant but not too large to avoid numerical issues
2. These patterns can be combined to model more complex logical relationships
3. Choice of M is important: too small might cut off feasible solutions, too large might cause numerical instability