

2023

Optimizarea Lactatelor

PROIECT REALIZAT DE:

POPAZ LUCIAN, GRUPA 323AC

SĂTMAR ELENA, GRUPA 323AC

1. Descrierea problemei

Această problemă vizează planificarea producției și transportului de produse alimentare între fabrici și magazine, cu scopul de a minimiza costurile. Sunt disponibile mai multe fabrici care produc diferite tipuri de alimente, iar magazinele solicită anumite cantități de produse.

Datele de intrare includ lista alimentelor, fabricilor și magazinelor implicate, cantitatea de produse pe care fiecare fabrică o produce într-o oră pentru fiecare aliment, numărul de ore de lucru, capacitatea de producție a fabricilor per total, cererea fiecărui magazin pentru fiecare aliment și costurile asociate fiecărei combinații de produs și fabrică.

Scopul problemei este de a determina cantitățile optime de produse produse de fiecare fabrică și cantitățile optime de produse transportate de la fiecare fabrică la fiecare magazin, astfel încât cererea fiecărui magazin să fie satisfăcută, capacitatea de producție a fabricilor să nu fie depășită și costurile să fie minime.

2. Formularea matematică a problemei

Datele problemei:

$A \in \text{Alimente} = \{\text{Brânză, Cașcaval, Iaurt, Lapte, Unt}\}$

$F \in \text{Fabrici} = \{F1, F2, F3\}$

$M \in \text{Magazine} = \{M1, M2\}$

Fabrică\Aliment	Brânză	Cașcaval	Iaurt	Lapte	Unt
Fabrica 1	17	18	13	10	12
Fabrica 2	20	16	14	8	11
Fabrica 3	16	19	15	9	10

Table 1. Costurile fiecărui aliment din fiecare fabrică

Fabrica\Aliment	Brânză	Cașcaval	Iaurt	Lapte	Unt
Fabrica 1	15	10	12	15	10
Fabrica 2	10	15	18	13	12
Fabrica 3	13	18	15	10	11

Table 2. Numărul maxim de alimente procesate într-o oră

Fabrica	Fabrica 1	Fabrica 2	Fabrica 3
Producție Maximă	200	250	150

Table 3. Producția maximă a unei zile de lucru a fabricilor

Magazin\Aliment	Brânză	Cașcaval	Iaurt	Lapte	Unt
Magazin 1	30	40	20	60	55
Magazin 2	60	30	40	20	35

Table 4. Cererea magazilenor pentru fiecare aliment în parte

Parametrii problemei:

- *ProducerePerOra*: Numărul maxim de alimente procesate din fiecare într-o oră pentru fiecare fabrică
- *CerereMagazine*: Cererea fiecărui magazine asupra cantității fiecărui aliment
- *CapacitateaFabricii*: Producția maximă a unei zile de lucru a fiecărei fabrici;
- *H*: numărul de ore lucrate într-o zi de lucru în fiecare fabrică; $H = 8$;
- *CusturiProdus*: costul fiecărui produs de a fi procesat pentru fiecare fabrică;

Variabile decizionale:

- Cantitatea de alimente **a** produsă de fiecare fabrică **f**: $x_{f,a} \geq 0$;
- Cantitatea de alimente **a** transportată de la fiecare fabrică **f** la fiecare magazin **m**: $t_{f,m,a} \geq 0$;
- Costul fiecărei combinații de fabrică **f** și aliment **a**: $c_{f,a} \geq 0$.

Modelul:

Funcția obiectiv: Minimizare costului total al produselor:

$$\min \sum_{f=1}^F \sum_{a=1}^A c_{f,a} * x_{f,a}$$

Constrângeri:

1. Cantitatea produsă nu poate depăși capacitatea fabricii:

$$\sum_{a=1}^A x_{f,a} \leq \text{CapacitateaFabricii}[f], \forall f$$
2. Cantitatea produsă nu poate depăși cantitatea produsă într-o oră:

$$x_{f,a} \leq \text{ProducerePerOra}[f, a] * H, \forall f, a$$

3. Cererea magazinelor trebuie să fie satisfăcută:

$$\sum_{f=1}^F t_{f,m,a} = \text{CerereMagazine}[f,a], \forall m,a$$

4. Cantitatea transportată nu poate depăși cantitatea produsă:

$$\sum_{m=1}^M t_{f,m,a} \leq \sum_{a=1}^A x_{f,a}, \forall f$$

5. Cantitatea transportată nu poate depăși disponibilitatea din fabrică:

$$t_{f,m,a} \leq x_{f,a} \forall f,m,a$$

6. Cantitatea minimă produsă în fiecare fabrică:

$$x_{f,a} \geq 1, \forall f,a$$

7. Relația cost – cantitate:

$$c_{f,a} = \text{CosturiProdus}[f,a] * x_{f,a}, \forall f,a$$

3. Soluționarea problemei formulate

Soluția problemei implică găsirea unei asignări optime a cantităților de produse produse și transportate, respectând constrângerile de cerere și capacitate, și minimizând costurile asociate. Rezultatele includ cantitățile de produse produse în fiecare fabrică, costurile asociate fiecărei combinații de produs și fabrică, cantitățile de produse transportate de la fiecare fabrică la fiecare magazin și costul total al produselor.

Astfel, am realizat un cod în Julia^[1] în care am introdus datele problemei, paramterii acesteia și paramterii decizionali, funcția obiectiv pe care ne-o dorim și constrângeri prezentate la pasul anterior. Apoi, am rezolvat problema de optimizare utilizând HiGHS Optimizer și am obținut o soluție optimă.

La rularea programului, acesta va afișa următoarele:

Numărul de alimente din fiecare tip produse de fiecare fabrică în parte și costul acestora:

Status final: OPTIMAL

Valoarea funcției obiectiv: 3169.0

Costul total al produselor: 150847.0

Produsele produse în fabrica F1:

- Au fost produse 1 bucăți de Branza cu un cost de 17.0
- Au fost produse 1 bucăți de Cascaval cu un cost de 18.0
- Au fost produse 38 bucăți de Iaurt cu un cost de 494.0
- Au fost produse 1 bucăți de Lapte cu un cost de 10.0
- Au fost produse 1 bucăți de Unt cu un cost de 12.0

Produsele produse în fabrica F2:

- Au fost produse 1 bucăți de Branza cu un cost de 20.0
- Au fost produse 38 bucăți de Cascaval cu un cost de 608.0
- Au fost produse 1 bucăți de Iaurt cu un cost de 14.0
- Au fost produse 58 bucăți de Lapte cu un cost de 464.0
- Au fost produse 1 bucăți de Unt cu un cost de 11.0

Produsele produse în fabrica F3:

- Au fost produse 58 bucăți de Branza cu un cost de 928.0
- Au fost produse 1 bucăți de Cascaval cu un cost de 19.0
- Au fost produse 1 bucăți de Iaurt cu un cost de 15.0
- Au fost produse 1 bucăți de Lapte cu un cost de 9.0
- Au fost produse 53 bucăți de Unt cu un cost de 530.0

Numărul de produse transportate de la fiecare fabrică la fiecare magazin în parte:

Produsele transportate de la fabrica F1 la magazinul M1:

- Au fost transportate 1 bucăți de Branza
- Au fost transportate 1 bucăți de Cascaval
- Au fost transportate 18 bucăți de Iaurt
- Au fost transportate 1 bucăți de Lapte
- Au fost transportate 1 bucăți de Unt

Produsele transportate de la fabrica F1 la magazinul M2:

- Au fost transportate 1 bucăți de Branza
- Au fost transportate 1 bucăți de Cascaval
- Au fost transportate 38 bucăți de Iaurt
- Au fost transportate 1 bucăți de Lapte

Produsele transportate de la fabrica F2 la magazinul M1:

- Au fost transportate 1 bucăți de Branza
- Au fost transportate 38 bucăți de Cascaval
- Au fost transportate 1 bucăți de Iaurt
- Au fost transportate 58 bucăți de Lapte
- Au fost transportate 1 bucăți de Unt

Produsele transportate de la fabrica F2 la magazinul M2:

- Au fost transportate 1 bucăți de Branza
- Au fost transportate 28 bucăți de Cascaval
- Au fost transportate 1 bucăți de Iaurt
- Au fost transportate 18 bucăți de Lapte

Produsele transportate de la fabrica F3 la magazinul M1:

- Au fost transportate 28 bucăți de Branza
- Au fost transportate 1 bucăți de Cascaval
- Au fost transportate 1 bucăți de Iaurt
- Au fost transportate 1 bucăți de Lapte
- Au fost transportate 53 bucăți de Unt

Produsele transportate de la fabrica F3 la magazinul M2:

- Au fost transportate 58 bucăți de Branza
- Au fost transportate 1 bucăți de Cascaval
- Au fost transportate 1 bucăți de Iaurt
- Au fost transportate 1 bucăți de Lapte
- Au fost transportate 35 bucăți de Unt

4. Contribuția fiecărui membru al echipei în rezolvarea temei

- Research asupra problemelor de optimizare: Popaz Lucian, Sătmăru Elena
- Crearea unor idei de problemă: Popaz Lucian, Sătmăru Elena
- Formularea matematică a problemei: Popaz Lucian
- Implementarea soluției în Julia: Sătmăru Elena
- Testarea programului: Popaz Lucian, Sătmăru Elena
- Scrierea documentației: Popaz Lucian, Sătmăru Elena

5. Anexă

1. Programul în Julia:

```
2. using JuMP
3. using HiGHS
4.
5. # Date de intrare
6. Alimente = ["Branza", "Cascaval", "Iaurt", "Lapte", "Unt"]
7. A = length(Alimente)
8. Fabrici = ["F1", "F2", "F3"]
9. F = length(Fabrici)
10. Magazine = ["M1", "M2"]
11. M = length(Magazine)
12.
13. ProducerePerOra = [15 10 12 15 10;
14.                    10 15 18 13 12;
15.                    13 18 15 10 11] # Cantitatea de produse pe care fiecare
    fabrică o produce într-o oră pentru fiecare aliment
16.
17. H = 8 # Numărul de ore lucrate
18.
19. CapacitateaFabricii = [200; 250; 150] # Capacitatea de producție a
    fabricilor per total
20.
21. CerereMagazine = [30 40 20 60 55;
22.                   60 30 40 20 35]
23.
24. CosturiProdus = [17 18 13 10 12;
25.                  20 16 14 8 11;
26.                  16 19 15 9 10] # Matricea costurilor pentru fiecare produs
    și fabrică
27.
28. # Model
29. model = Model(optimizer_with_attributes(HiGHS.Optimizer)) # Utilizăm HiGHS
    Optimizer cu presolve activat
30.
31. @variable(model, x[f = 1:F, a = 1:A] >= 0, Int) # Cantitatea de alimente
    produsă
32. @variable(model, t[f = 1:F, m = 1:M, a = 1:A] >= 0, Int) # Cantitatea de
    alimente transportată de la fiecare fabrică la fiecare magazin
33. @variable(model, c[f = 1:F, a = 1:A] >= 0) # Costul fiecărei combinații de
    fabrică și aliment
34.
```

```
35.@objective(model, Min, sum(c[f, a] for f in 1:F, a in 1:A)) # Minimizarea
    costului total al fabricilor
36.
37.@constraint(model, [f = 1:F], sum(x[f, a] for a in 1:A) <=
    CapacitateaFabricii[f])
38.@constraint(model, [f = 1:F, a = 1:A], x[f, a] <= ProducerePerOra[f, a] * H)
39.
40.# Constrângeri pentru transport
41.@constraint(model, [m = 1:M, a = 1:A], sum(t[f, m, a] for f in 1:F) ==
    CerereMagazine[m, a]) # Cererea magazinelor trebuie să fie satisfăcută
42.@constraint(model, [m = 1:M, a = 1:A], sum(t[f, m, a] for f in 1:F) <=
    sum(x[f, a] for f in 1:F)) # Cantitatea transportată nu poate depăși
    cantitatea produsă
43.@constraint(model, [f = 1:F, m = 1:M, a = 1:A], t[f, m, a] <= x[f, a]) #
    Cantitatea transportată nu poate depăși disponibilitatea din fabrică
44.
45.# Cantitate minimă produsă în fiecare fabrică
46.@constraint(model, [f = 1:F, a = 1:A], x[f, a] >= 1)
47.
48.# Relația cost - cantitate
49.@constraint(model, [f = 1:F, a = 1:A], c[f, a] == CosturiProdus[f, a] * x[f,
    a])
50.
51.# Rezolvare
52.optimize!(model)
53.println("Status final: ", termination_status(model))
54.
55.if termination_status(model) == MOI.OPTIMAL
56.    println("Valoarea funcției obiectiv: ", objective_value(model))
57.
58.    @expression(model, total_cost, sum(c[f, a] * value(x[f, a]) for f in 1:F,
    a in 1:A)) # Expresie pentru costul total
59.    println("Costul total al produselor: ", JuMP.value(total_cost), "\n")
60.
61.    for f in 1:F
62.        println("Produsele produse în fabrica ", Fabricii[f], ":")
63.        for a in 1:A
64.            if value(x[f, a]) > 0.1 # Printează alimentele produse
65.                println("Au fost produse ", round(Int64, value(x[f, a])), "
    bucăți de ", Alimente[a], " cu un cost de ", round(value(c[f, a]), digits =
    2))
66.            end
67.        end
68.    println()
69.    end
```



```
70.
71.     for f in 1:F
72.         for m in 1:M
73.             println("\nProdusele transportate de la fabrica ", Fabrici[f], "
              la magazinul ", Magazine[m], ":")
74.             for a in 1:A
75.                 if value(t[f, m, a]) > 0.1 # Printează alimentele
                    transportate
76.                     println("Au fost transportate ", round(Int64, value(t[f,
                    m, a])), " bucăți de ", Alimente[a])
77.                 end
78.             end
79.         end
80.     end
81.
82. else
83.     println("Nicio soluție disponibilă")
84. end
85.
```