

Tema 3 Optimizări 2022-2023

Proiect realizat de Niță Daniel-Marius din grupa 322AC

I. Descrierea problemei

Suntem în cadrul echipei de HR al unei firme. Ne propunem să angajăm 6 oameni, câte 3 persoane pentru fiecare departament. Departamentele sunt: Backend Developer și Frontend Developer. Compania își dorește ca tot procesul de angajare, cu scopul de a maximiza costurile salariale posibile, deoarece vrem să recrutăm cei mai buni oameni cu puțință, să se întocmească sub următoarele condiții:

- Salariul lunar maxim pentru toți noii angajați este: 25000 €.
- Fiecare poziție trebuie ocupată.
- Compania dorește ca suma anilor de experiență în domeniu a angajaților pentru Backend să fie minim 9 ani, pentru Frontend să fie minim 6 ani, însă suma anilor de experiență a tuturor să fie minim 18.
- Compania dorește ca cel puțin o fată să fie angajată de-a lungul procesului de recrutare pentru fiecare departament.
- Compania dorește ca cel puțin o persoană să aibă o diplomă de master pentru fiecare departament.
- Compania dorește ca angajații din departamentul de Frontend să dețină împreună măcar un R, un L și un C, iar departamentul de Backend să dețină împreună măcar un F și un L. (vezi tabelul de mai jos pentru prescurtări).
- Dacă candidatul deține următoarele aptitudini i se va oferi un bonus salarial (bonusul este luat în calcul la salariul maxim) conform următorului tabel:

Prescurtare	Aptitudini	Bonus
F	Fullstack Dev Training	1.20 * salariul solicitat
L	Abilități de lider	1.15 * salariul solicitat
R	Reziliența la stres	1.10 * salariul solicitat
C	Comunicare	1.05 * salariul solicitat
O	Nemenționat/Nu deține	salariul solicitat

Candidații au dreptul să pună 0-4 aptitudini (0 înseamnă O), iar bonusul se aplică astfel, în cazul există mai multe aptitudini:

$R+C \Rightarrow \text{Salariul final} = 1.10 * 1.05 * \text{salariul solicitat}$.

Profilurile candidaților care au aplicat pentru postul de Frontend Developer arată astfel:

Nr.	Gen	Diplomă	Experiență (ani)	Aptitudini	Salariu solicitat(€)	Salariu final(€)
1	M	Licență	5	L+R	2000	2530
2	F	Master	3	L+C	2500	3018.75
3	F	Licență	7	O	2300	2300
4	M	Master	1	L+R+C	2600	3453.45
5	M	Master	4	F+L	3000	4140
6	F	Licență	2	L+R	2000	2530
7	F	Licență	3	F+L+R+C	2800	4462.92
8	F	Master	1	R+C	2500	2887.5
9	M	Licență	6	L+R	3000	3795
10	F	Master	4	F+L+C	3200	4638.8

Profilurile candidaților care au aplicat pentru posturile de Backend Developer arată astfel:

Nr.	Gen	Diplomă	Experiență (ani)	Aptitudini	Salariu solicitat(€)	Salariu final(€)
1	M	Licență	5	L	3500	4025
2	F	Master	3	F+C	3300	4158
3	F	Licență	7	R+C	3000	3465
4	M	Master	1	F+L+R+C	3500	5578.65
5	M	Master	3	R	3000	3300
6	F	Licență	4	L+R+C	3000	3984.75
7	M	Licență	2	F+L+C	3100	4491.9
8	F	Master	4	R+C	3200	3696
9	M	Licență	3	O	2800	2800
10	F	Master	6	L+R	3500	4427.5

II. Formularea matematică a problemei

Vrem să calculăm:

Max pentru

$$f(x) = \sum_{l,c}^{2,10} Employee_{l,c} * x_{l,c}$$

Restricții:

- $x_{l,c} \in \{0, 1\}$, unde $l \in \{1,2\}$ și $c \in \{1,2, \dots, 10\}$
- $\sum_{l,c}^{2,10} Employee_{l,c} * x_{l,c} \leq 25000$
- $\sum_c^{10} x_{1,c} = 3$
- $\sum_{c=1}^{10} x_{2,c} = 3$
- $\sum_{c=1}^{10} Experience_{1,c} * x_{l,c} \geq 6$
- $\sum_{c=1}^{10} Experience_{2,c} * x_{2,c} \geq 9$
- $\sum_{l=1,c=1}^{2,10} Experience_{l,c} * x_{l,c} \geq 18$
- $\sum_{w=1}^{W=nr\ femei} x_{1,Women_w} \geq 1$
- $\sum_{w=1}^{W=nr femei} x_{2,Women_w} \geq 1$
- $\sum_{i=1}^{A_{l_r}=nr\ cand\ cu\ aptitudinea\ R} x_{1,Apt_{R_i}} \geq 1$
- $\sum_{i=1}^{A_{l_f}=nr\ cand\ pentru\ Frontend\ cu\ aptitudinea\ L} x_{1,Apt_{l_f_i}} \geq 1$
- $\sum_{i=1}^{A_{l_c}=nr\ cand\ cu\ aptitudinea\ C} x_{1,Apt_{C_i}} \geq 1$
- $\sum_{i=1}^{A_{l_f}=nr\ cand\ cu\ aptitudinea\ F} x_{2,Apt_{F_i}} \geq 1$
- $\sum_{i=1}^{A_{l_b}=nr\ cand\ pentru\ Backend\ cu\ aptitudinea\ L} x_{2,Apt_{l_b_i}} \geq 1$

III. Soluționarea problemei formulate

Cod Julia:

```
using JuMP
using HiGHS

Departamente = ["Frontend", "Backend"]
Frontend=[2530, 3018.75, 2300, 3453.45, 4140, 2530, 4462.92, 2887.5, 3795,
4638.8] #salariile cerute inițial de candidații ce au aplicat pentru
departamentul de Frontend
Backend=[4025, 4158, 3465, 5578.65, 3300, 3984.75, 4491.9, 3696, 2800, 4427.5]
#salariile cerute intial de candidații ce au aplicat pentru departamentul de
Backend
Experience=[5 3 7 1 4 2 3 1 6 4;
            5 3 7 1 3 4 2 4 3 6] #matricea cu anii de experiența a fiecarui
candidat(pe linia 1 a celor de la Frontend și pe linia 2 a celor la Backend)

Women=[2, 3, 6, 7, 8, 10] #vectorul pozitiilor din lista a candidatilor de sex
feminin (din motive de simplitate e acelasi la ambele departamente)
W=length(Women) #lungimea vectorului Women

Master=[2, 4, 5, 8,10] #vectorul pozitiilor din listă a candidaților cu diplomă
de master (din motive de simplitate e acelasi la ambele departamente)
M=length(Master) #lungimea vectorului Master

Apt_R=[1, 4 ,6, 7, 8, 9] #vectorul pozitiilor din listă a candidților pentru
Frontend ce dețin aptitudinea R
A_r=length(Apt_R) #lungimea vectorului Apt_R

Apt_L_f=[1, 2, 4, 5, 6, 7, 9, 10] #vectorul pozitiilor din listă a candidților
pentru Frontend ce dețin aptitudinea L
A_l_f=length(Apt_L_f) #lungimea vectorului Apt_L_f

Apt_C=[2, 4, 7, 8, 10] #vectorul pozitiilor din listă a candidților pentru
Frontend ce dețin aptitudinea C
A_c=length(Apt_C) #lungimea vectorului Apt_C

Apt_F=[2, 4, 7] #vectorul pozitiilor din listă a candidților pentru Backend ce
dețin aptitudinea F
A_f=length(Apt_F) #lungimea vectorului Apt_F

Apt_L_b=[1, 4, 6, 7, 10] #vectorul pozitiilor din listă a candidților pentru
Backend ce dețin aptitudinea L
A_l_b=length(Apt_L_b) #lungimea vectorului Apt_L_b
```

```

C=length(Frontend)    #C este numarul de aplicanti pentru fiecare departament
L=length(Departamente)    #L este numarul de departamente

#Cream matricea cu salariile finale pentru fiecare departament
Employee = Array{Float64}(undef, 2, C)
for i in 1:C
    Employee[1,i]=Frontend[i]
    Employee[2,i]=Backend[i]
end

#Afisam matricea
println("Matricea salariilor finale solicitate este: ")
println(Employee)

#Realizam modelul
IC = Model(HiGHS.Optimizer)

#Alegem variabilele integrale si pozitive
@variable(IC, x[l=1:L, c=1:C], Bin)

#Scriem functia
@objective(IC, Max, sum(Employee[l,c] * x[l,c] for l in 1:L, c in 1:C))

#Scriem Restrictiile
@constraint(IC, sum(Employee[l,c] * x[l,c] for l in 1:L, c in 1:C) <= 25000) #Ne
asiguram ca salariul tuturor angajatilor nu depaseste 21000
@constraint(IC, sum(x[1,c] for c in 1:C) == 3) #Ne asiguram ca angajam 3 Frontend
Developeri
@constraint(IC, sum(x[2,c] for c in 1:C) == 3) #Ne asiguram ca angajam 3 Backend
Developeri
@constraint(IC, sum(Experience[1,c] * x[1,c] for c in 1:C) >= 6) #Experienta
pentru Frontend
@constraint(IC, sum(Experience[2,c] * x[2,c] for c in 1:C) >= 9) #Experienta
pentru Backend
@constraint(IC, sum(Experience[l,c] * x[l,c] for l in 1:L, c in 1:C) >= 18)
#Experienta combinata
@constraint(IC, sum(x[1,Women[w]] for w in 1:W) >= 1) #Macar o femeie e angajata
ca Frontend Dev
@constraint(IC, sum(x[2,Women[w]] for w in 1:W) >= 1) #Macar o femeie e angajata
ca Backend Dev
@constraint(IC, sum(x[1,Master[m]] for m in 1:M) >= 1) #Macar o persoana cu
diploma de master e angajata ca Frontend Dev
@constraint(IC, sum(x[2,Master[m]] for m in 1:M) >= 1) #Macar o persoana cu
diploma de master e angajata ca Backend Dev

```

```

@constraint(IC, sum(x[1,Apt_R[i]] for i in 1:A_r) >= 1) #Macar o persoana are
aptitudinea R ca Frontend Dev
@constraint(IC, sum(x[1,Apt_L_f[i]] for i in 1:A_l_f) >= 1) #Macar o persoana are
aptitudinea L ca Frontend Dev
@constraint(IC, sum(x[1,Apt_C[i]] for i in 1:A_c) >= 1) #Măcar o persoana are
aptitudinea C ca Frontend Dev
@constraint(IC, sum(x[2,Apt_F[i]] for i in 1:A_f) >= 1) #Macar o persoana are
aptitudinea F ca Backend Dev
@constraint(IC, sum(x[2,Apt_L_b[i]] for i in 1:A_l_b) >= 1) #Macar o persoana are
aptitudinea L ca Backend Dev
@constraint(IC, [l=1:L, c=1:C], x[l,c] <= 1) #Ne asiguram ca rezultatul sa fie
cat mai aproape de realitate astfel incat x va fi 1 daca candidatul e angajat,
respectiv 0 daca NU este angajat

#Optimizam modelul
optimize!(IC)

println("Status: $(termination_status(IC))")
if termination_status(IC) == MOI.OPTIMAL
    println("Valoarea functiei obiectiv: $(objective_value(IC))")

    #Afisam matricea x ca sa ne dam seama daca au fost facute bine calculele
    println("Matricea x este: ")
    for l in 1:L
        for c in 1:C
            print("| $(value(x[l,c])) |")
        end
        println("")
    end

    #Punem niste instructiuni de afisare mai usor de vazut si inteles
    for l in 1:L
        println("Candidații angajați pentru departamentul de $(Departamente[l]): ")
        for c in 1:C
            if value(x[l,c]) >= 0.9999
                println("Candidatul nr. $(c) a fost angajat.")
            end
        end
    end
else
    println("Nu exista solutii disponibile")
end

```

Output:

Running HiGHS 1.5.1 [date: 1970-01-01, git hash: 93f1876e4]

Copyright (c) 2023 HiGHS under MIT licence terms

Presolving model

15 rows, 20 cols, 129 nonzeros

15 rows, 20 cols, 119 nonzeros

Objective function is integral with scale 100

Solving MIP model with:

15 rows

20 cols (20 binary, 0 integer, 0 implied int., 0 continuous)

119 nonzeros

Nodes				B&B Tree				Objective Bounds				Dynamic Constraints				Work													
Proc.		InQueue				Leaves		Expl.				BestBound		BestSol				Gap				Cuts		InLp		Confl.			
LpIters		Time																											
	0	0	0	0.00%	73683.22	-inf	inf	0	0	0	0	0.0s																	
R	0	0	0	0.00%	25000	24915.65	0.34%	0	0	0	6	0.0s																	
L	0	0	0	0.00%	25000	24955.9	0.18%	72	8	13	21	0.0s																	
L	0	0	0	0.00%	25000	24972.45	0.11%	72	5	13	21	0.0s																	
T	0	0	0	0.00%	25000	24988.1	0.05%	72	5	13	72	0.0s																	
T	122	8	50	58.59%	25000	24994.95	0.02%	107	6	388	690	0.1s																	

Solving report

Status Optimal

Primal bound 24994.95

Dual bound 24994.95

Gap 0% (tolerance: 0.01%)

Solution status feasible

24994.95 (objective)

0 (bound viol.)

5.84395683373e-13 (int. viol.)

0 (row viol.)

Timing 0.10 (total)

0.00 (presolve)

0.00 (postsolve)

Nodes 220

LP iterations 1212 (total)

297 (strong br.)

58 (separation)

110 (heuristics)

Status: OPTIMAL

Valoarea functiei obiectiv: 24994.9500000016

Matricea x este:

| -7.529937622158547e-14 || 0.0 || 0.0 || 0.0 || 0.0 || 1.0 || 0.0 || 0.0 || 1.0 || 1.0 |

| 0.99999999999999091 || 0.0 || 0.0 || 1.0 || 0.0 || 0.0 || 0.0 || 5.843956833726911e-13 || 0.0 || 1.0 |

Candidații angajați pentru departamentul de Frontend:

Candidatul nr. 6 a fost angajat.

Candidatul nr. 9 a fost angajat.

Candidatul nr. 10 a fost angajat.

Candidații angajați pentru departamentul de Backend:

Candidatul nr. 1 a fost angajat.

Candidatul nr. 4 a fost angajat.

Candidatul nr. 10 a fost angajat.

Matricea xOPTIM este:

- 7.529937622158547e- 14	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0
0.99999999999999091	0.0	0.0	0.0	1.0	0.0	0.0	5.843956833726911e- 13	0.0	1.0