

O aplicatie intr-o fabrica de otel

Problema bazata pe Mixed-Integer Linear Programming (MILP)

Programarea liniară cu număr întreg mixt (MILP) este adesea utilizată pentru analiza și optimizarea sistemului, deoarece prezintă o metodă flexibilă și puternică pentru rezolvarea problemelor mari, complexe, cum ar fi cazul de simbioză industrială și integrarea proceselor.



Detalierea problemei:

Dorim sa amestecam otelurile cu diverse compozitii chimice pentru a obtine 25 de tone de otel cu o compozitie chimica specificata. Rezultatul ar trebui sa aiba 5% carbon si 5% molybdenum (molybdenul este un metal alb-argintiu, foarte rezistent la coroziune si are unul dintre cele mai înalte puncte de topire dintre toate elementele) pe greutate, asta inseamna $25 \text{ tone} * 5\% = 1.25 \text{ tone}$ de carbon si 1.25 tone de molybdenum.

$$\sum_i greutate(i) = 25$$

Obiectivul: este de a minimiza costurile pentru amestecarea otelului.

Patru lingouri de otel sunt disponibile pentru cumparare. Este disponibil doar unul din fiecare lingou:

Lingou	Greutatea in tone	%Carbon	%Molybdenum	<u>Pret</u> <u>Tona</u>
1	5	5	3	350 ron
2	3	4	3	330 ron
3	4	5	4	310 ron
4	6	3	4	280 ron

Sunt disponibile pentru cumpărare trei clase de oțel aliat și o clasă de fier vechi. Oțelurile aliaje și resturi pot fi achiziționate în cantități fracționate:

Aliaj	%Carbon	%Molybdenum	<u>Pret</u> <u>Tona</u>
1	8	6	500 ron
2	7	7	450 ron
3	6	8	400 ron
Resturi	3	9	100 ron

Formularea problemei

Pentru a formula problema, mai întâi decidem asupra variabilelor de control. Luam variabilele lingouri (1) = 1 pentru a însemna că achiziționăm lingoul 1 și lingouri (1) = 0 pentru a însemna că nu cumpărăm lingoul. În mod similar, variabilele lingouri (2) până la lingouri (4) sunt variabile binare care indică dacă achiziționăm lingouri de la 2 la 4.

Variabilele aliaje (1) până la aliajele (3) sunt cantitățile în tone de aliaje 1, 2 și 3 pe care le achiziționăm. Resturile reprezintă cantitatea de fier vechi pe care o achiziționăm.

```

otelproblema = optimproblem%optimproblem= creaza o problema de optimizare
% cu proprietatile implicite
% variabilele lingoului sunt întregi si sunt binare, deoarece exista doar
% una din fiecare lingou disponibila
% optimvar=pentru a crea o variabila binara; creeaza o variabila de optimizare scalara.
% O variabila de optimizare este un obiect simbolic care ne permite sa
% cream expresii pentru functia obiectiva si constrângerile problemei în ceea ce priveste
variabila.
lingouri = optimvar('lingouri',4,'Type','integer','LowerBound',0,'UpperBound',1);
%celelalte varibile sunt nonnegative
aliaje = optimvar('aliaje',3,'LowerBound',0); % varibilele pozitive au doar limita inferioara
resturi = optimvar('resturi','LowerBound',0);
% Cream expresii pentru costurile asociate variabilelor.
greutatea_lingourilor = [5,3,4,6];
pretul_lingourilor = greutatea_lingourilor.*[350,330,310,280];
pretul_aliajelor = [500,450,400];
pretul_resturilor = 100;
pret = pretul_lingourilor*lingouri + pretul_aliajelor*aliaje + pretul_resturilor*resturi;
% Includem costul ca functie obiectiva în problema.
otelproblema.Objective = pret;
% Problema are trei constrângeri de egalitate.
% Prima constrângere este ca greutatea totala este de 25 de tone.
% Calculam greutatea oțelului.
greutatea_totala = greutatea_lingourilor*lingouri + sum(aliaje) + resturi;
% A doua constrângere este ca greutatea carbonului este de 5% din 25 de tone, sau 1,25 tone.

```

```

% Calculam greutatea carbonului din otel.
lingou_de_carbon = [5,4,5,3]/100;
aliaj_de_carbon = [8,7,6]/100;
rest_de_carbon = 3/100;
carbonul_total = (greutatea_lingourilor.*lingou_de_carbon)*lingouri + aliaj_de_carbon*aliaje
+ rest_de_carbon*resturi;
% A treia constrângere este ca greutatea molybdenului este de 1,25 tone.
% Calculam greutatea molibdenului în otel.
lingou_de_molyb = [3,3,4,4]/100;
aliaj_de_molyb = [6,7,8]/100;
rest_de_molyb = 9/100;
totalul_molybului = (greutatea_lingourilor.*lingou_de_molyb)*lingouri +
aliaj_de_molyb*aliaje + rest_de_molyb*resturi;

% Includem constrângerile în problema.
otelproblema.Constraints.consgr = greutatea_totala == 25;
otelproblema.Constraints.conscarb = carbonul_total == 1.25;
otelproblema.Constraints.consmolyb = totalul_molybului == 1.25;
Tabelulingourilor=table(greutatea_lingourilor,pretul_lingourilor,lingou_de_carbon,lingou_de
_molyb)
Tabelulaliajelor=table(pretul_aliajelor,aliaj_de_carbon,aliaj_de_molyb)
Tabelulresturilor=table(pretul_resturilor,rest_de_carbon,rest_de_molyb)
showproblem(otelproblema)
% Acum ca avem toate intrarile, apelam solutia
[sol,fval] = solve(otelproblema); %solve-rezolva problema de optimizare sau problema
ecuatiei
sol.lingouri %sol-variabila de tip struct
sol.aliaje
sol.resturi
fval %fval-este o variabila de tip double

```

Tabelullingourilor =

1×4 [table](#)

greutatea_lingourilor				pretul_lingourilor				lingou_de_carbon			
5	3	4	6	1750	990	1240	1680	0.05	0.04	0.05	0.03

Tabelulaliajelor =

1×3 [table](#)

pretul_aliajelor			aliaj_de_carbon			aliaj_de_molyb		
500	450	400	0.08	0.07	0.06	0.06	0.07	0.08

Tabelulresturilor =

1×3 [table](#)

pretul_resturilor	rest_de_carbon	rest_de_molyb
100	0.03	0.09

[OptimizationProblem](#) :

Solve for:

aliaje, lingouri, resturi

minimize :

1750*lingouri(1) + 990*lingouri(2) + 1240*lingouri(3) + 1680*lingouri(4)
+ 500*aliaje(1) + 450*aliaje(2) + 400*aliaje(3) + 100*resturi

subject to consgr:

5*lingouri(1) + 3*lingouri(2) + 4*lingouri(3) + 6*lingouri(4) + aliaje(1)
+ aliaje(2) + aliaje(3) + resturi == 25

subject to conscarb:

0.25*lingouri(1) + 0.12*lingouri(2) + 0.2*lingouri(3) + 0.18*lingouri(4)
+ 0.08*aliaje(1) + 0.07*aliaje(2) + 0.06*aliaje(3) + 0.03*resturi == 1.25

subject to consmolyb:

0.15*lingouri(1) + 0.09*lingouri(2) + 0.16*lingouri(3) + 0.24*lingouri(4)
+ 0.06*aliaje(1) + 0.07*aliaje(2) + 0.08*aliaje(3) + 0.09*resturi == 1.25

```

variable bounds:
    0 <= aliaje(1)
    0 <= aliaje(2)
    0 <= aliaje(3)

    0 <= lingouri(1) <= 1
    0 <= lingouri(2) <= 1
    0 <= lingouri(3) <= 1
    0 <= lingouri(4) <= 1

    0 <= resturi

```

Intlinprog-Scrierea funcțiilor obiective pentru probleme liniare sau quadratice

Solving problem using intlinprog.

LP: Optimal objective value is 8125.600000.

Cut Generation: Applied 3 mir cuts.
 Lower bound is 8495.000000.
 Relative gap is 0.00%.

Optimal solution found.

```

ans =

    1.0000
    1.0000
         0
    1.0000

ans =

    7.2500
         0
    0.2500

ans =

    3.5000

fval =

    8.4950e+03

```

Achiziția optimă costă 8.495 ron. Cumpăram lingourile 1, 2 și 4, dar nu 3 și cumpăram 7.25 tone de aliaj 1, 0.25 tone de aliaj 3 și 3.5 tone de fier vechi.