

Projet : Optimisation de production

Programmation linéaire et maximisation de profit

Introduction

Les entreprises doivent maximiser leur profit tout en respectant des contraintes de ressources limitées. La programmation linéaire offre une approche mathématique robuste pour résoudre ces problèmes de gestion de production.

Objectifs

- Formuler un problème d'optimisation linéaire réaliste.
- Utiliser le solveur linprog de MATLAB.
- Interpréter la solution optimale et analyser la sensibilité.

Justification du choix des méthodes

Programmation linéaire / Simplex : Choisie car :

- Garantit une solution optimale pour les problèmes linéaires.
- Très efficace même sur de grandes instances industrielles.
- Standard en optimisation opérationnelle depuis les années 1940.
- Mise en œuvre robuste dans MATLAB via linprog.

Méthodologie

Formulation mathématique

Minimiser $c^T x$ sous contraintes $Ax \leq b$, $Aeq \cdot x = beq$, $lb \leq x \leq ub$.

Pour la maximisation de profit, on minimise $-\text{profit}$.

Code MATLAB

```

1 clc; clear;
2
3 % --- Param tres du probl me ---
4 f = double([-40; -30]);          % Maximisation du profit -> minimiser -
    profit
5 A = double([1, 2; 3, 2]);        % Contraintes de ressources
6 b = double([8; 12]);            % Capacit s maximales
7 lb = double([0; 0]);            % Limites inf rieures
8 ub = [];                        % Pas de limites sup rieures explicites
9 x0 = zeros(2,1);                % Point initial
10
11 % Options du solveur
12 options = optimoptions('linprog', 'Display', 'iter');
13 % R solution
14 [x_opt, fval] = linprog(f, A, b, [], [], lb, ub, x0, options);
15 % R sultats
16 profit_max = -fval; % on a minimis -profit
17 fprintf('== Optimisation Lin aire ==\n');
18 fprintf('Production optimale: A = %.2f, B = %.2f\n', x_opt(1), x_opt(2));
19 fprintf('Profit maximal: %.2f\n', profit_max);

```

Tests et r sultats

Code de g neration des tests

```

1 % Visualisation de la r gion admissible et solution optimale
2 x1 = linspace(0, 10, 100);
3 x2_1 = (8 - 1*x1) / 2;          % Contrainte 1
4 x2_2 = (12 - 3*x1) / 2;         % Contrainte 2
5
6 figure;
7 plot(x1, x2_1, 'b-', 'LineWidth', 2, 'DisplayName', 'x1 + 2x2 <= 8');
8 hold on;
9 plot(x1, x2_2, 'r-', 'LineWidth', 2, 'DisplayName', '3x1 + 2x2 <= 12');
10 fill([0, 0, 4], [0, 4, 0], 'g', 'FaceAlpha', 0.3, 'DisplayName', 'R gion
    admissible');
11 plot(x_opt(1), x_opt(2), 'ko', 'MarkerSize', 10, 'DisplayName', 'Solution
    optimale');
12 xlim([0, 10]); ylim([0, 6]);
13 xlabel('Produit A');
14 ylabel('Produit B');
15 title('Programmation lin aire - Zone admissible et optimum');
16 legend show;
17 grid on;
18 saveas(gcf, 'optimisation_lp.png');
19
20 % Analyse de la sensibilit : variation prix
21 prix_A = 30:5:60;
22 profits = [];
23 for p = prix_A
24     f_temp = [-p; -30];
25     [x_temp, fval_temp] = linprog(f_temp, A, b, [], [], lb, []);
26     profits = [profits, -fval_temp];
27 end
28

```

```

29 figure;
30 plot(prix_A, profits, 'bo-', 'LineWidth', 1.5);
31 xlabel('Prix unitaire A');
32 ylabel('Profit maximal');
33 title('Analyse sensibilit : impact du prix sur profit');
34 grid on;
35 saveas(gcf, 'sensibilite_prix.png');

```

Visualisation

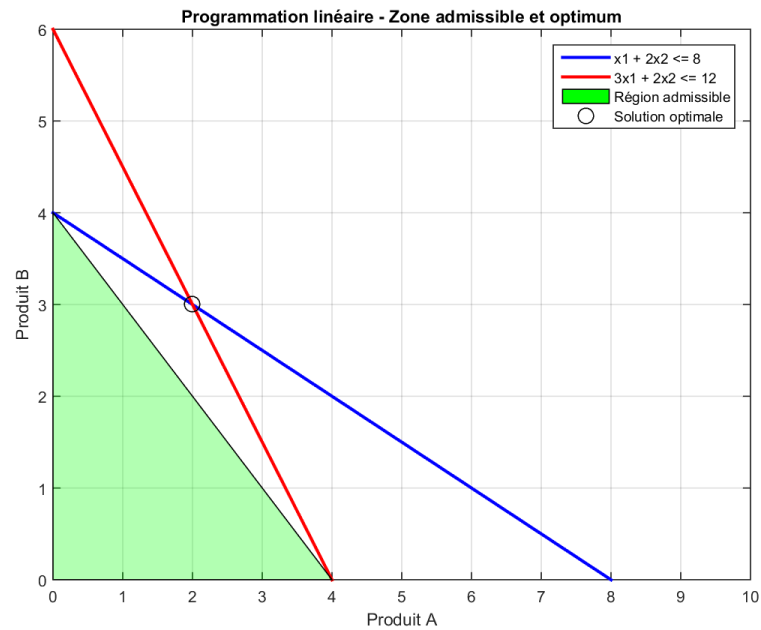


Figure 1 : Zone admissible (verte) définie par les contraintes. La solution optimale est au sommet du polyèdre.

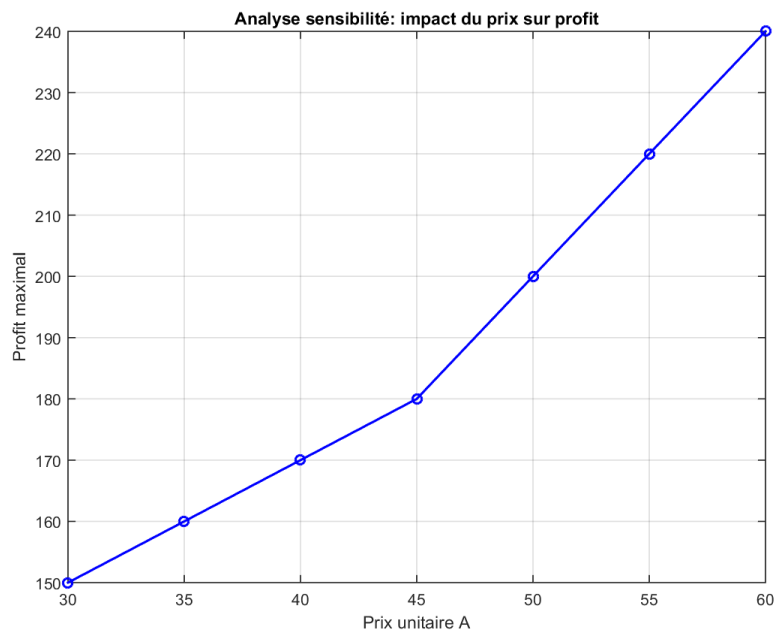


Figure 2 : Analyse de sensibilité montrant l'évolution du profit maximal en fonction du prix du produit A.

Conclusion

La programmation linéaire est un outil puissant et incontournable pour la gestion optimale des ressources en entreprise. Elle permet une prise de décision éclairée et quantifiée.