Cours de recherche opérationnelle I

Nadia Brauner

Nadia.Brauner@imag.fr

Grenoble, 2015-2016

Auteurs

Ont participé à la rédaction de ce cours (par ordre d'arrivée)

- Nadia Brauner
- Christophe Rapine
- Julien Moncel
- Laurent Beaudou

Ont aidé, corrigé, relu et donné des idées

- Gerd Finke
- Yann Kieffer
- Van Dat Cung

Ont donné les TD et proposé des exercices

- Ayse Akbalik
- Sergei Lenglet
- Aline Parreau
- Guillaume Massonnet

Formations à Grenoble

Formation initiale

- RO à l'UJF (M1 Info, L3 Miage, Polytech'RICM4)
- Gestion de la production à l'UJF (M1 Miage)
- Optimisation pour l'énergie (M2 Miage)
- Outils Formels et Graphes (Polytech'RICM2)
- RO à l'ENSIMAG (1A, 2A)
- RO à l'ENSGI (1A, 2A)
- Master Informatique, parcours **Recherche Opérationnelle, Combinatoire et Optimisation**

Formation continue

- Recherche opérationnelle (tous les ans, 4 jours)
- Graphes et optimisation (tous les ans, 3 jours)

Recherche Opérationnelle : faisons connaissance

Nadia Brauner
Nadia Brauner@imag.fr

Professeur Grenoble I

Laboratoire

- équipe Recherche Opérationnelle
- équipe Opti-Com

Présidente 12-13 de la

- Société Française de RO-AD

Responsable **Master 2 R ROCO**

Recherche Opérationnelle,

Combinatoire et Optimisation

Recherche Opérationnelle : faisons connaissance

Problèmes théoriques

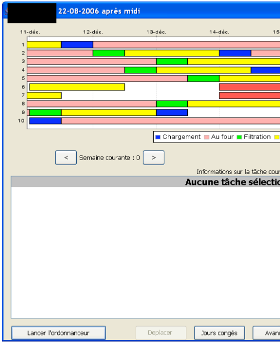
- Ordonnancement high-multiplicity ($\in$ NP ?)
- Ordonnancement dans ateliers robotisées
- OC appliquée à la micro-électronique

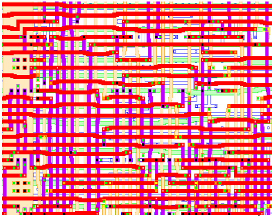
Contrats industriels

- ILOG : Problèmes complexes de transport
- IFP : Planification d'expériences chimiques
- de Facto : Optimisation du test des circuits

Participation à la **création d'une startup**

- OASIC : optimisation de la conception de cellules logiques





La recherche opérationnelle

Plan

Plan

Recherche Opérationnelle ou Science de la Décision

Définitions

Cambridge Dictionary

Operational research UK (US operations research)
The systematic study of how best to solve problems in business and industry

Wikipedia

Operations research, operational research, or simply OR, is the use of mathematical models, statistics and algorithms to aid in decision-making

Roadef

Recherche Opérationnelle : approche scientifique pour la résolution de problèmes de gestion de systèmes complexes

Recherche Opérationnelle

Science du ≪ comment mieux faire avec moins ≫

Des **outils** pour

- rationaliser
- simuler
- optimiser
- planifier

l'architecture et le fonctionnement des systèmes industriels et économiques.

Des **modèles** pour analyser des situations complexes

Permet aux décideurs de faire des **choix efficaces et robustes**

Recherche Opérationnelle

Approche quantitative pour produire les meilleures décisions

- Une discipline à la croisée des mathématiques et de l'informatique
  - prolongement de l'algorithmique
  - manipulant des structures plus élaborées : graphes, polyèdres...
  - domaine d'application de la théorie de la complexité algorithmique
- Une boite à outils de méthodes, tant positives que négatives, pour aborder sainement et sereinement les problèmes d'optimisation

Recherche Opérationnelle

Les outils de RO-AD

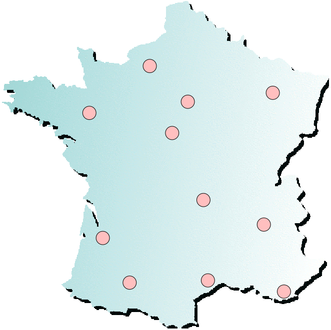
- aident à trouver
  - une solution où l'homme n'en trouvait pas
  - une solution sur des problèmes nouveaux où l'homme n'a aucune expérience
  - plusieurs solutions là où l'homme n'en envisageait qu'une
- aident à juger de la qualité d'une solution
- aident à confirmer / justifier des décisions

Plan

Recherche Opérationnelle

Voyageur de commerce (TSP)

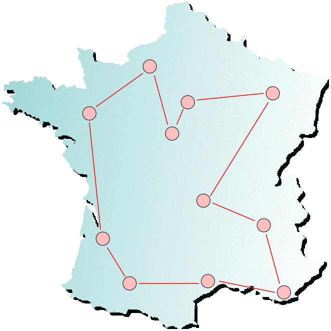
- Un voyageur de commerce, basé à Toulon, doit visiter ses clients à travers la France.
- Il souhaite effectuer la tournée la plus courte possible.



Recherche Opérationnelle

Voyageur de commerce

- Instance : n villes avec une matrice de distances
- Solution : tournée visitant chaque ville et revenant à Toulon



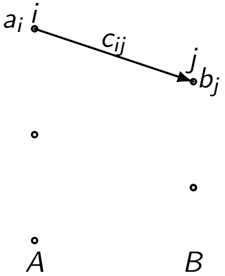
Recherche Opérationnelle

Algorithme Glouton pour le TSP

Recherche Opérationnelle

Transport

- de marchandises
- des entrepôts vers les clients
- coûts de transport, distance sur les arcs
- trouver le meilleur plan de distribution



$$\min \sum c_{ij} x_{ij}$$

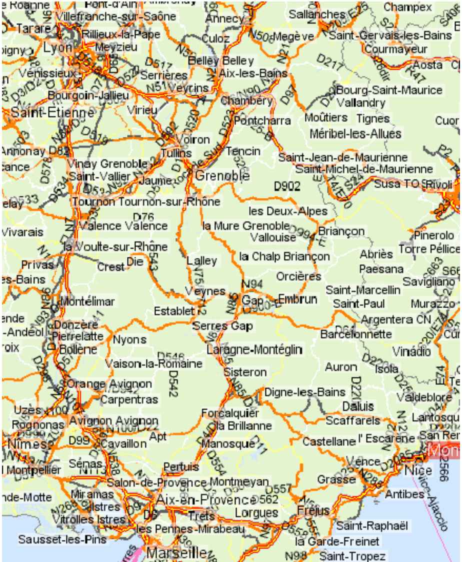
$$\begin{aligned} \sum_{j \in B} x_{ij} &\leq a_i \\ \sum_{i \in A} x_{ij} &\geq b_j \\ x_{ij} &\geq 0 \end{aligned}$$

Recherche Opérationnelle

Applications

Plus court chemin

Quel est le trajet le plus court entre Grenoble et Nice en voiture ?



Recherche Opérationnelle

24h de RO

- 8h : optimisation de la récolte et du dépôt des déchets recyclables
- ...
- 15h : placement automatique des véhicules pour une association de partage de voitures
- 16h : gestion des retards dans les transports publics pour minimiser l'impact sur les passagers
- ...

http://www.24hor.org/

Recherche Opérationnelle

le 15 octobre 2012 :

Économie

ÉCONOMIE | Monde | Crise de l'euro | France | Entreprises | Marchés | Argent et Patrimoine | C

Le prix Nobel d'économie attribué aux Américains Alvin Roth et Lloyd Shapley

Le Monde.fr avec AFP et Reuters | 15.10.2012 à 13h41 • Mis à jour le 15.10.2012 à 18h58

Nobelprize.org
The Official Web Site of the Nobel Prize

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2012
Alvin E. Roth, Lloyd S. Shapley

KUNGL. VETENSKAPS-AKADEMIEN
THE ROYAL SWEDISH ACADEMY OF SCIENCES

English
English (pdf)
Swedish
Swedish (pdf)

Press Release

15 October 2012

The Royal Swedish Academy of Sciences has decided to award The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel for 2012 to

Alvin E. Roth
Harvard University, Cambridge, MA, USA, and Harvard Business School, Boston, MA, USA

and

Lloyd S. Shapley
University of California, Los Angeles, CA, USA

"for the theory of stable allocations and the practice of market design".

Mariages stables

Mariages stables
Des femmes : Alice, Bénédicte, Camille
Des hommes : Elie, François, Gondran

Préférences des femmes

A :	G	E	F
B :	F	E	G
C :	G	E	F

Préférences des hommes

E :	A	B	C
F :	B	C	A
G :	A	C	B

Comment faire les couples ?

Mariages stables

Un couplage est **instable** s'il contient deux personnes A et B non mariées ensemble qui se préfèrent mutuellement à leurs conjoints :

F est mariée avec g
G est marié avec f
F préfère G à g
G préfère F à f

Questions

- Comment vérifier qu'un couplage est stable ?
- Est-ce qu'il existe toujours un couplage stable ?
- Est-ce qu'on sait trouver un couplage stable quand il existe ?

Mariages stables

Applications

Situations où les mécanismes de marchés traditionnels ne fonctionnent pas
Répartition de biens rares, hétérogènes, indivisibles

Affectations de candidats sur des places

- élèves - écoles d'ingénieur
- travailleurs - postes
- internes - hôpitaux
- étudiants - universités

Dons d'organes (reins)

Recherche Opérationnelle

Les challenges ROADEF

`http://challenge.roadef.org/`

- 2010 Gestion d'énergie (EDF)
- 2009 Gestion des perturbations dans le transport aérien (Amadeus)
- 2007 Planification des techniciens et des interventions pour les télécommunications (France Telecom)
- 2005 Ordonnancement de véhicules pour une chaîne de montage automobile (Renault)
- 2003 Gestion des prises de vue réalisées par un satellite d'observation de la Terre (ONERA et CNES)
- 2001 Allocation de fréquences avec polarisation (CELAR, armée)
- 1999 Gestion de stock de matériels (Bouygues)

Recherche Opérationnelle

Le challenges ROADEF/EURO 2012

- Réaffectation de machines
- Proposé par Google
- 82 équipes enregistrées dans 33 pays
- 30 équipes qualifiées
- Vainqueur Junior : équipe polonaise
- Vainqueur Open Source et Senior : équipe bosniaques

Recherche Opérationnelle

Le challenges ROADEF/EURO 2014

- Trains don't vanish !
- Proposé par SNCF
- 35 équipes enregistrées
- Vainqueur Sprint : étudiants du Master

Connexion

Université Joseph Fourier GRENOBLE ESPACE ÉTUDIANTS

SCIENCES, TECHNOLOGIES, SANTE

Site de l'Université | Alumni | Plan vert

Recherche ok

VIE À L'UJF

VIE DES CAMPUS

VIE SPORTIVE ET CULTURELLE

VIE CITOYENNE

FORMATION, ORIENTATION, INSERTION

Accueil › Challenge ROADEF 2014, l'aventure continue...

Challenge ROADEF 2014, l'aventure continue...

ETUDIANT, FORMATION

le 8 avril 2014

L'équipe de Luc Arnaud, Grigori German, Guillaume Pataut et Siao-Leu Phouratsamay,étudiants à l'UJF, classée première des qualifications junior.

Le résultat des qualifications pour la finale du challenge international de Recherche Opérationnelle EURO/ROADEF 2014 proposé par la SNCF a été annoncé pendant le diner de gala de la conférence ROADEF 2014, se déroulant le 27 février 2014 au palais de la bourse à Bordeaux. En novembre 2013, des étudiants de l'UJF avaient gagné le sprint, première étape de ce challenge. Cette même équipe, constituée de Luc Arnaud, Grigori German, Guillaume Pataut et Siao-Leu Phouratsamay, étudiants en Recherche Opérationnelle du Master Mathématiques et Informatique, est première au classement junior des qualifications du challenge !

Quant au laboratoire G-SCOP, il compte deux équipes classées parmi

La Société Française de Recherche Opérationnelle et Aide à la Décision

a organisé une soirée débat - table ronde :

"La Recherche Opérationnelle, clé de la performance des entreprises" ?

à l'Université Paris Dauphine, salle Raymond Aron

le Jeudi 18 avril à 19h00

`http://www.roadef.org/content/roadef/soireeRO.htm`

- Introduction et historique de la RO
- Mesure de performance de la RO
- Ingrédients d'une bonne approche RO
- L'enseignement de la RO
- Le serious game, un outil pour convaincre
- Faut-il un modèle simple ou haute fidélité ? Solutions robustes
- RO, SI et capacités de calcul

- Emmanuel Guyot, Directeur Marketing et Revenue Management **TF1 PUBLICITE**
- Yves Caseau, Executive Vice-Président **BOUYGUES TELECOM**
- Animation : Denis Montaut, Président d'**Eurodécision**
- Nadia Brauner, Présidente de la Roadef, G-SCOP
- Yvon Quérou, Directeur Informatique **AIR FRANCE**
- Jean-Charles Billaut, Professeur à l'Université de Tours
- Jean-Paul Hamon, ex Executive Vice-Président Développement **AMADEUS**

Recherche Opérationnelle

Domaines d'application

- Conception, configuration et exploitation
  de systèmes techniques complexes
  (réseaux de communication, systèmes d'information)
- Gestion de la chaîne logistique
  (transports, production, stocks. . .)
- Gestion stratégique d'investissements
- et aussi
  santé, instruction publique, voirie,
  ramassage et distribution de courrier,
  production et transport d'énergie,
  télécommunications, banques, assurances. . .

Recherche Opérationnelle

Domaines d'application

Production : maximiser le profit selon disponibilité de la main d'œuvre, demande du marché, capacité de production, prix de revient du matériau brut...

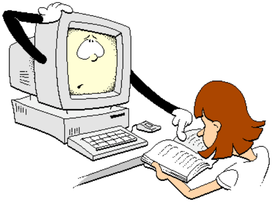
Transport : minimiser distance totale parcourue selon quantités de matériaux à transporter, capacité des transporteurs, points de ravitaillement en carburant...

- grande importance dans le milieu industriel : production, transport, emploi du temps, finance...

Recherche Opérationnelle

Face à un problème pratique de décision

- Aspects mathématiques
  - contraintes, objectifs, simplifications
- Modélisation
  - graphes, programmation linéaire, PPC...
- Analyse des modèles et résolution
  - étude de complexité : que peut-on espérer pour le temps de résolution imparti ?
  - mise au point d'algorithmes
- Implémentation et analyse des résultats
  - valider par rapport à la demande
  - itérer avec le demandeur si nécessaire
- Déploiement des solutions
  - Intégration logicielle



Plan

Recherche Opérationnelle

Programmation linéaire

min le coût / max le profit	min / max $\quad c_1x_1 + c_2x_2 \dots c_nx_n$
satisfaire la demande	$a_1x_1 + a_2x_2 \dots a_nx_n \geq b_1$
avec des ressources limitées	$a'_1x_1 + a'_2x_2 \dots a'_nx_n \leq b'_1$
quantités produites	$x_1, x_2 \dots x_n \geq 0$

Recherche Opérationnelle

Optimisation Combinatoire

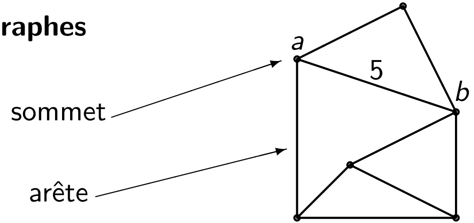
- Trouver la meilleure solution parmi un nombre fini mais très grand de choix
- Un problème d'OC se caractérise par :
    - La présence de choix, à faire parmi un ensemble fini d'alternatives
    - Une notion de coût, ou de gain, ou de perte
    - La nécessité de faire globalement les bons choix, de manière à optimiser la valeur objectif
- exemples : emplois du temps...

Combinatoire

- échiquier tronqué
- `http://mathsamodeler.ujf-grenoble.fr/LAVALISE/`

Recherche Opérationnelle

Graphes



Valuation des arêtes = coûts, temps, distance, capacités. . .

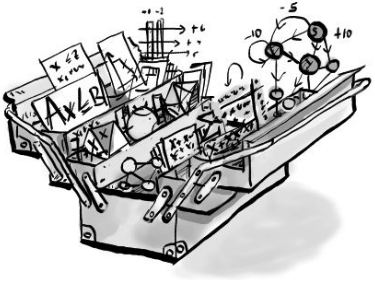
- meilleur chemin de *i* à *j*
- meilleurs parcours
  - passant par chaque ville
  - passant par chaque arête
- . . .

Représentation de réseaux, de précédences en ordonnancement, de compatibilité de produits...

Recherche Opérationnelle

Autre outils

- Files d'attente
- Stochastique
- Simulation



dessin de Lionel Lagarde

À l'interface de

- Informatique : algorithmique
- Mathématiques : modélisation
- Économie : gestion, stratégie

Plan

Recherche Opérationnelle : entreprises en France

Grands groupes avec un pôle R&D en RO

- Airfrance
- La SNCF
- EDF
- France Telecom
- Bouygues
- GDF Suez
- La poste
- Renault
- Air Liquide
- SFR
- Google

Recherche Opérationnelle : entreprises en France

Pour les autres entreprises

- Sociétés de conseil spécialisées
- Logiciels sur étagère
- Laboratoires académiques

Recherche Opérationnelle : entreprises en France

Sociétés de conseil
accompagnent les industriels pour mettre en place des systèmes d'aide à la décision

- EURODECISION
  Conseil en optimisation des ressources et planification de la production, outils d'aide à la décision
- ARTELYS
  Solutions en optimisation
- ...

Recherche Opérationnelle : entreprises en France

Éditeurs de logiciels

librairies dédiées à des problèmes mathématiques

- ILOG (IBM)
  Optimization tools and engines, Visualization software components, Supply chain applications
- COSYTEC
  offrir des solutions logicielles, à base de technologie de programmation par contraintes, pour résoudre des problèmes d'optimisation des ressources
- FICO et ARTELYS
  Fico XPress : logiciels de modélisation de problèmes linéaires ou quadratiques avec variables réelles ou entières
  Knitro : optimiseur non linéaire
  Artelys Kalis : Programmation par contraintes
- ...

Recherche Opérationnelle : entreprises en France

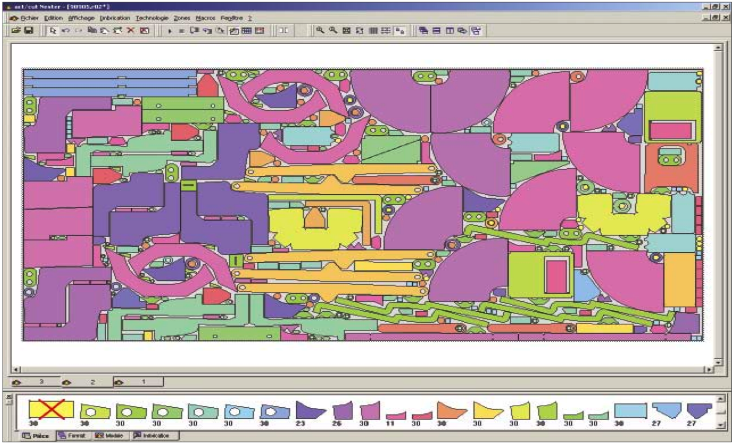
Éditeurs de logiciels

librairies dédiées à des problèmes métiers

- ALMA : Placement et découpe
  ex : petit bateau (habits), chantiers navals
- AMADEUS : Voyage
  plateforme de réservation centralisée pour l'industrie du voyage et outils de gestion des compagnies aériennes
- Optilogistics : transport et logistique
  progiciels d'optimisation de tournées et de planification du transport
- Ordecsys, Oracle...

Recherche Opérationnelle : entreprises en France

Alma : Découpe



Recherche Opérationnelle : en France

Et dans le monde académique

enquête 2010 de la Roadef

- ≈ 75 équipes ou laboratoires
- ≈ 1400 membres
- ≈ 700 chercheurs, enseignants chercheurs, ingénieurs de recherche permanents
- ≈ 500 doctorants

Recherche Opérationnelle : pour en savoir plus

Le Livre Blanc de la Recherche Opérationnelle en France

- Comment les industriels s'organisent
- D'incontestables réussites
- Sociétés de conseil et éditeurs de logiciels

http://www.roadef.org/

Plan

Bibliographie

de Werra, D., Liebling, T.-M., and Hêche, J.-F.
Recherche Opérationnelle pour Ingénieurs, Tome 1.
Presses Polytechniques et Universitaires Romandes, 2003.

Sakarovitch, M.
Optimisation Combinatoire, Graphes et Programmation Linéaire.
Hermann, Enseignement des sciences, Paris, 1984.

Sakarovitch, M.
Optimisation Combinatoire, Programmation Discrète.
Hermann, Enseignement des sciences, Paris, 1984.

Wolsey, L. A.
Integer Programming.
Wiley-Interscience, 1998.

Webographie

Cours

- Poly de cours
  `http://www.g-scop.grenoble-inp.fr/~braunern`
- Compléments au cours
  `Chamilo`, utiliser le lien avec connection
  `CaseInE`, pour les étudiants de Grenoble
- M2R de Recherche Opérationnelle, Combinatoire et Optim.
  `http://roco.g-scop.grenoble-inp.fr`

Vie de la RO en France

- Société française de RO
  `http://www.roadef.org`
- Groupe de Recherche en RO du CNRS
  `http://gdrro.lip6.fr`
- Séminaire de recherche en OC et RO à Grenoble
  `http://www.g-scop.grenoble-inp.fr/`

Webographie

Collection de ressources pour la RO

- http://www2.informs.org/Resources/
- http://www.ensta.fr/~diam/ro/

Logiciels pour la RO

- http://www.coin-or.org/resources.html
- http://www.wior.uni-karlsruhe.de/bibliothek/

Blogs sur la RO

- http://blog.vcu.edu/lamclay/
- http://mat.tepper.cmu.edu/blog/

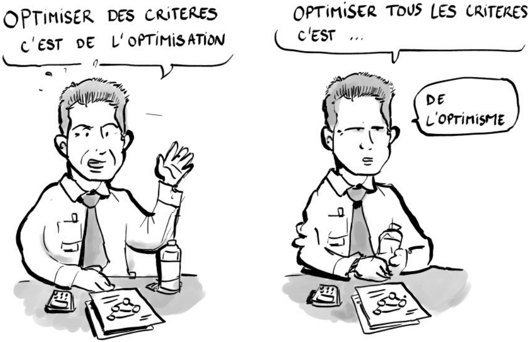
Des challenges industriels internationaux en RO

- http://challenge.roadef.org/

Recherche Opérationnelle

En conclusion

- faire le mieux
  - coût min, meilleur profit, plus courte distance, le plus rapide...
- avec les ressources disponibles
  - temps machine, postes de travail, mémoire, ressource homme, matière première, camions...



Dessins de L. Lagarde

Programmation linéaire

Plan

Plan

Programmation linéaire

Cadre de la PL

Programmation linéaire

nombre fini de variables réelles, contraintes linéaires, objectif linéaire

Variables $x_1, x_2 \dots x_n$ réelles

Contrainte générique (contrainte i) :

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

Fonction-objectif générique (à maximiser / minimiser) :

$$f(x_1, x_2 \dots x_n) = \sum_{j=1}^{n} c_j x_j$$

Programmation linéaire

Exemple : culture de courgettes et navets

Contraintes concernant les quantités d'engrais et d'anti-parasites

- 8ℓ engrais A disponible
  $\rightarrow 2\ell/m^2$ nécessaires pour courgettes, $1\ell/m^2$ pour navets
- 7ℓ engrais B disponible
  $\rightarrow 1\ell/m^2$ nécessaires pour courgettes, $2\ell/m^2$ pour navets
- 3ℓ anti-parasites disponible
  $\rightarrow 1\ell/m^2$ nécessaires pour navets

Objectif : produire le maximum (en poids) de légumes, sachant que rendements $= 4kg/m^2$ courgettes, $5kg/m^2$ navets

Programmation linéaire

Exemple : culture de courgettes et navets

Variables de décision

- x_c : surface de courgettes
- x_n : surface de navets

Fonction objectif $\max 4x_c + 5x_n$

Contraintes

- $2x_c + x_n \leq 8$ (engrais A)
- $x_c + 2x_n \leq 7$ (engrais B)
- $x_n \leq 3$ (anti-parasites)
- $x_c \geq 0$ et $x_n \geq 0$

Programmation linéaire

Intérêt de la PL

Problème général d'optimisation sous contraintes

$\Rightarrow$ **AUCUNE méthode GÉNÉRALE de résolution ! !**

Problème linéaire quelconque
$\Rightarrow$ existence de méthodes de résolution générales et efficaces

Ces méthodes sont efficaces en théorie et en pratique
$\Rightarrow$ existence de nombreux logiciels de résolution :
Excel, CPLEX, Mathematica, LP-Solve...

Cadre restrictif

- variables réelles
- contraintes linéaires
- objectif linéaire

Programmation linéaire

Représentation in extenso

- $\max 4x_c + 5x_n$
- $2x_c + x_n \leq 8$ (engrais A)
- $x_c + 2x_n \leq 7$ (engrais B)
- $x_n \leq 3$ (anti-parasites)
- $x_c \geq 0$ et $x_n \geq 0$

Représentation matricielle

$$\max \quad (4 \quad 5)\begin{pmatrix} x_c \\ x_n \end{pmatrix}$$

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_c \\ x_n \end{pmatrix} \leq \begin{pmatrix} 8 \\ 7 \\ 3 \end{pmatrix}$$

$$x_c \geq 0 \qquad x_n \geq 0$$

Programmation linéaire

Représentation in extenso

$$\begin{array}{llcll}
\max z = & \sum_j c_j x_j & & & \\
s.c. & \sum_j a_{ij} x_j & \left\{ \begin{array}{c} \leq \\ \geq \\ = \end{array} \right\} & b_i & i = 1, 2 \dots m \\
 & x_j & \geq & 0 & j = 1, 2 \dots n
\end{array}$$

Programmation linéaire

- second membre $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$
- n var. de décision $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$
- matrice de format $m \times n$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \ddots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

- coût (ou profit) $c = (c_1, c_2 \dots c_n)$

Représentation matricielle

$$\begin{array}{llll} \max z & = cx & & \\ s.c. & Ax & \left\{ \begin{array}{c} \leq \\ \geq \\ = \end{array} \right\} & b \\ & x & \geq & 0 \end{array}$$

Programmation linéaire

Vocabulaire

- x_i **variable** de décision du problème
- $x = (x_1, \ldots, x_n)$ **solution réalisable** (admissible)
  ssi elle satisfait toutes les contraintes
- ensemble des solutions réalisables = **domaine** ou région admissible
- $x = (x_1, \ldots, x_n)$ **solution optimale**
  ssi elle est réalisable et optimise la fonction-objectif
- **contraintes** inégalité ou égalité linéaire
  - $a_{11}x_1 + a_{12}x_2 \ldots + a_{1n}x_n \leq b_1$
  - $a_{21}x_1 + a_{22}x_2 \ldots + a_{2n}x_n \geq b_2$
  - $a_{31}x_1 + a_{32}x_2 \ldots + a_{3n}x_n = b_3$
- **fonction objectif** (ou fonction économique) linéaire
  - $\max / \min\ c_1x_1 + c_2x_2 \ldots + c_nx_n$

Programmation linéaire

Applications

Feuille de TD : Programmation linéaire

- Exercice Production de vins
- Exercice Publicité
- Exercice Compagnie aérienne
- Exercice Fabrication d'huile d'olives
- Exercice Laiterie
- Exercice Bergamote

Programmation linéaire

Forme canonique d'un PL

- maximisation
- toutes les variables sont non négatives
- toutes les contraintes sont des inéquations du type "$\leq$"

$$\max z = \sum_j c_j x_j$$

$$\begin{array}{lrcll} s.c. & \sum_j a_{ij} x_j & \leq & b_i & i = 1, 2 \ldots m \\ & x_j & \geq & 0 & j = 1, 2 \ldots n \end{array}$$

- forme matricielle

$$\begin{array}{lrcl} \max z = cx & & & \\ s.c. & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

Programmation linéaire

Forme standard d'un PL

- maximisation
- toutes les variables sont non négatives
- toutes les contraintes sont des équations

$$\max z = \sum_j c_j x_j$$

$$\begin{array}{lrcll} s.c. & \sum_j a_{ij} x_j & = & b_i & i = 1, 2 \ldots m \\ & x_j & \geq & 0 & j = 1, 2 \ldots n \end{array}$$

- forme matricielle

$$\begin{array}{lrcl} \max z = cx & & & \\ s.c. & Ax & = & b \\ & x & \geq & 0 \end{array}$$

Programmation linéaire

Passage entre les formes

- équation $\to$ inéquation

$$ax = b \quad \Longleftrightarrow \quad \begin{cases} ax \leq b \\ ax \geq b \end{cases}$$

- max $\leftrightarrow$ min $\qquad \max f(x) = -\min -f(x)$
- inéquation $\to$ équation : ajouter une variable d'écart

$$\begin{array}{lclr} ax \leq b & \Longleftrightarrow & ax + s = b, & s \geq 0 \\ ax \geq b & \Longleftrightarrow & ax - s = b, & s \geq 0 \end{array}$$

- variable non contrainte $\to$ variables positives

$$x \lesseqgtr 0 \quad \Longleftrightarrow \quad \begin{cases} x = x^+ - x^- \\ x^+, x^- \geq 0 \end{cases}$$

Programmation linéaire

Passage entre les formes

Feuille de TD : Programmation linéaire

- Exercice Formes linéaires et canoniques

Programmation linéaire

Linéariser un problème non linéaire

e_i : expression linéaire des variables de décision

- **obj :** $\min \max\{e_1, e_2 \dots e_n\}$

$$\begin{cases} \min y \\ y \geq e_i \qquad i = 1, 2 \dots n \end{cases}$$

- **obj :** $\max \min\{e_1, e_2 \dots e_n\}$

$$\begin{cases} \max y \\ y \leq e_i \qquad i = 1, 2 \dots n \end{cases}$$

- **obj :** $\min |e_1|$

$$|e| = \max(e, -e) \qquad \begin{cases} \min y \\ y \geq e_1 \\ y \geq -e_1 \end{cases} \qquad \begin{cases} \min e^+ + e^- \\ e_1 = e^+ - e^- \\ e^+, e^- \geq 0 \end{cases}$$

Programmation linéaire

Linéariser un problème non linéaire

Feuille de TD : Programmation linéaire

- Exercice Linéarisation

Programmation linéaire

Un peu d'histoire

- années 30-40 : Kantorovitch, économiste soviétique
  $\Rightarrow$ modèles linéaires pour la planification et l'optimisation de la production
- années 40-50 : Dantzig, mathématicien américain
  $\Rightarrow$ algorithme du simplexe
- application historique
  - Opérations Vittles et Plainfare pour ravitaillement de la trizone pendant le blocus de Berlin par pont aérien (23 juin 1948 – 12 mai 1949)
  - simplexe exécuté à la main (des milliers de variables), jusqu'à 12 000 tonnes de matériel par jour !
- 1975 : prix Nobel économie Kantorovitch
- XXIème siècle : logiciels de PL disponibles partout, utilisation de la PL dans tous les domaines industriels...

Plan

Interprétation géométrique

Exemple : culture de courgettes et navets

Variables de décision

- x_c : surface de courgettes
- x_n : surface de navets

Fonction objectif $\max 4x_c + 5x_n$

Contraintes

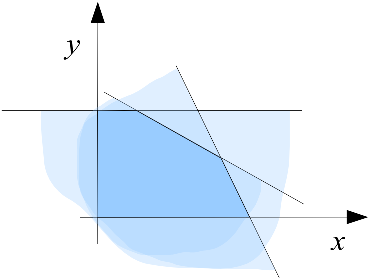
- $2x_c + x_n \leq 8$ (engrais A)
- $x_c + 2x_n \leq 7$ (engrais B)
- $x_n \leq 3$ (anti-parasites)
- $x_c \geq 0$ et $x_n \geq 0$

Interprétation géométrique

Interpréter les contraintes courgettes et navets

- $2x + y \leq 8 \Rightarrow$ demi-plan de $\mathbb{R}^2$
- $x + 2y \leq 7 \Rightarrow$ demi-plan
- $y \leq 3 \Rightarrow$ demi-plan
- $x \geq 0$ et $y \geq 0 \Rightarrow$ demi-plans

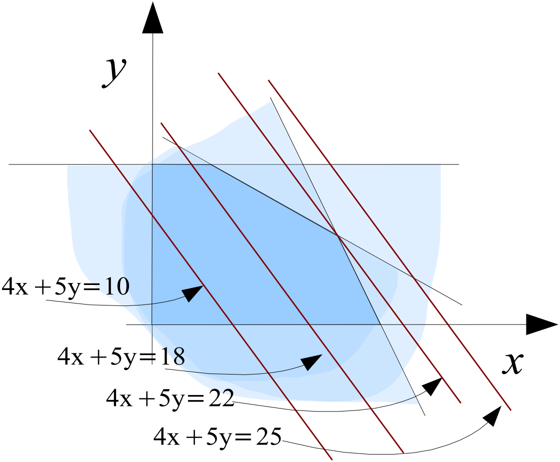
Ensemble des solutions réalisables = intersection de ces demi-plans : **polyèdre**



Interprétation géométrique

Optimiser l'objectif

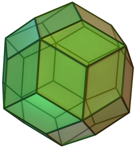
Les **lignes de niveau** $\{4x + 5y = \text{constante}\}$ sont des droites parallèles



Interprétation géométrique

Géométrie d'un PL

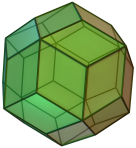
L'ensemble des solutions réalisables est toujours un **polyèdre** (intersection de demi-espaces)



Les lignes de niveau $\{f = \text{constante}\}$ de la fonction-objectif f sont des **hyperplans affines** ($n = 2 \Rightarrow$ droite, $n = 3 \Rightarrow$ plan...)

Interprétation géométrique

Géométrie d'un PL



Optimum atteint au bord

L'optimum de la fonction-objectif, s'il existe, est atteint en (au moins) un **sommet** du polyèdre.

Justification mathématique :
les dérivées partielles de $f(x) = c.x$ ne s'annulent jamais,
et le domaine $\{x \mid \sum_{j=1}^{n} a_{ij}x_j \leq b_i, i = 1, \ldots, m\}$ est compact
$\Rightarrow$ l'optimum est atteint au bord...

Programmation linéaire

Solutions d'un PL

La région admissible peut être

- vide
  - nb solutions optimales : 0
- non vide, bornée
  - nb solutions optimales : 1 ou ∞
- non vide, non bornée
  - nb solutions optimales : 0 ou 1 ou ∞

Proposer des exemples de PL pour chacun des cas

Feuille de TD : Programmation linéaire

- Exercice Résolution graphique
- Exercice Toujours plus de bénéfices !

Plan

Bases et points extrêmes

Rappels

$$\begin{array}{llcl} \max & z & = & cx \\ \text{s.c.} & Ax & \leq & b \\ & x & \geq & 0 \end{array}$$

- A matrice $m \times n$
- $x = (x_1\, x_2 \ldots x_n)$
- $b = (b_1\, b_2 \ldots b_m)$
- $c = (c_1\, c_2 \ldots c_n)$

- Les contraintes définissent un polyèdre
- La solution optimale est un sommet du polyèdre

Comment énumérer les sommets d'un polyèdre ?

Bases et points extrêmes

Passage à la forme standard

Forme standard

On peut rajouter des **variables d'écart** :

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \Leftrightarrow \sum_{j=1}^{n} a_{ij}x_j + e_i = b_i, e_i \geq 0$$

PL standard :

$$\begin{array}{lccc} \max & z(x) & = & c.x \\ \text{s.c} & Ax & = & b \\ & x & \geq & 0 \end{array}$$

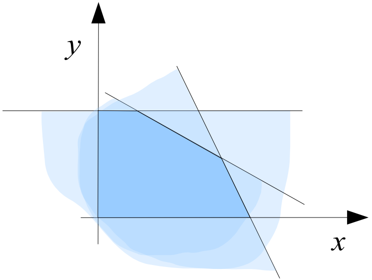
On travaille dans un espace de dimension plus grande, mais toutes les contraintes sont des égalités.

- Manipulations algébriques plus aisées

Bases et points extrêmes

Passage à la forme standard

$$\begin{aligned} \max z &= 4x + 5y \\ \text{s.c.} \quad & 2x + y \leq 8 \\ & x + 2y \leq 7 \\ & y \leq 3 \\ & x, y \geq 0 \end{aligned}$$



$$\begin{aligned} \max z &= 4x + 5y \\ \text{s.c.} \quad & 2x + y + e_1 = 8 \\ & x + 2y + e_2 = 7 \\ & y + e_3 = 3 \\ & x, y, e_1, e_2, e_3 \geq 0 \end{aligned}$$

9 points intéressants (intersection de contraintes)

5 points admissibles

énumération de ces 9 points comme solution de la forme standard (solutions de base)

Bases et points extrêmes

$$
\begin{array}{lccccccccccc}
\text{s.c.} & 2x & + & y & + & e_1 & & & & & = & 8 \\
 & x & + & 2y & & & + & e_2 & & & = & 7 \\
 & & & y & & & & & + & e_3 & = & 3 \\
 & x, & & y, & & e_1, & & e_2, & & e_3 & \geq & 0
\end{array}
$$

x	y	e_1	e_2	e_3	sol de base	admiss.	pt extrême
$\underline{0}$	$\underline{0}$	8	7	3	✔	✔	(0,0)
$\underline{0}$	8	$\underline{0}$	-9	-5	✔	✘	
$\underline{0}$	3.5	4.5	$\underline{0}$	-0.5	✔	✘	
$\underline{0}$	3	5	1	$\underline{0}$	✔	✔	(0,3)
4	$\underline{0}$	$\underline{0}$	3	3	✔	✔	(4,0)
7	$\underline{0}$	-6	$\underline{0}$	3	✔	✘	
	$\underline{0}$			$\underline{0}$	✘	✘	
3	2	$\underline{0}$	$\underline{0}$	1	✔	✔	(3,2)
2.5	3	$\underline{0}$	-1.5	$\underline{0}$	✔	✘	
1	3	3	$\underline{0}$	$\underline{0}$	✔	✔	(1,3)

$\{$**points extrêmes**$\} \iff \{$**solutions de base admissibles**$\}$

Bases et points extrêmes

- Système linéaire Ax=b
- A format $m \times n$, rang A $= m \leq n$
- **Base** de A : sous-matrice $B(m \times m)$ inversible de A
  $A = (B, N)$

$$(B, N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \qquad \text{ou} \qquad Bx_B + Nx_N = b$$

$$\Rightarrow \quad x_B = B^{-1}b - B^{-1}Nx_N$$

- **Solution de base** associée à B :
  - $x_N = 0$ variables hors base
  - $x_B = B^{-1}b$ variables de base

Bases et points extrêmes

Applications

Feuille de TD : Programmation linéaire

- Exercice Bases *2
- Exercice Solutions de bases et points extrêmes

Bases et points extrêmes

Base et solution de base

$$\begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \\ x, y, e_1, e_2, e_3 \geq 0 \end{cases}$$

Base initiale ? $\{e_1, e_2, e_3\}$ par exemple :

$$\begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \end{cases} \Leftrightarrow \begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases}$$

$e_1, e_2, e_3 =$ variables de base, $x, y =$ variables hors base

Bases et points extrêmes

Base et solution de base

$$\begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases}$$

- on met les variables hors base à 0
- on en déduit les valeur des variables de base

$$x = y = 0 \Rightarrow \begin{cases} e_1 = 8 - 2x - y = 8 \\ e_2 = 7 - x - 2y = 7 \\ e_3 = 3 - y = 3 \end{cases}$$

Bases et points extrêmes

- $Ax = b, \qquad x \geq 0$
- $(x_B, 0)$ associée à B est une **solution de base admissible** si $x_B \geq 0$
- {**points extrêmes** du polyèdre} $\Longleftrightarrow$ {solutions de base admissibles du système linéaire correspondant}
- nombre de points extrêmes $\approx C_n^m = \frac{n!}{m!(n-m)!}$
- solution de base dégénérée : certaines variables de base sont nulles
- si A est inversible : solution de base unique

Bases et points extrêmes

Base voisine et pivotage

Bases voisines

Deux sommets voisins correspondent à deux bases B et B' telles qu'on remplace une variable de B pour obtenir B'

- passer à un sommet voisin = changer de base (base voisine)

principe du pivotage

Bases et points extrêmes

Qui faire entrer dans la base ?

Essayons avec y : quelle est la valeur max que pourra avoir y ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

Bilan : $y_{\max} = 3$, pour $y = y_{\max}$ on a $e_1 = 5 - 2x$, $e_2 = 1 - x$, et $e_3 = 0$

▶ candidat pour une nouvelle base :
$\{e_1, e_2, e_3\} \cup \{y\} \setminus \{e_3\} = \{e_1, e_2, y\}$

$(x, y, e_1, e_2, e_3) = (0, 3, 5, 1, 0)$

Plan

L'algorithme du simplexe

Vers un algorithme de résolution

▶ Méthode de résolution "naïve" : énumérer tous les sommets, calculer f sur ces points, prendre le sommet pour lequel f est optimisé :

- fonctionne : nombre fini de sommets
- limitation : ce nombre peut être très grand en général...

L'algorithme du simplexe (G. B. Dantzig 1947) Algorithme itératif permettant de résoudre un problème de programmation linéaire.

L'algorithme du simplexe

Principe d'amélioration locale

À partir d'un sommet, chercher un sommet voisin qui améliore l'objectif.

Principe d'amélioration locale (maximisation) :

Soit x_0 sommet non optimum. Alors il existe x, un sommet **voisin** de x_0, tel que $f(x) > f(x_0)$.

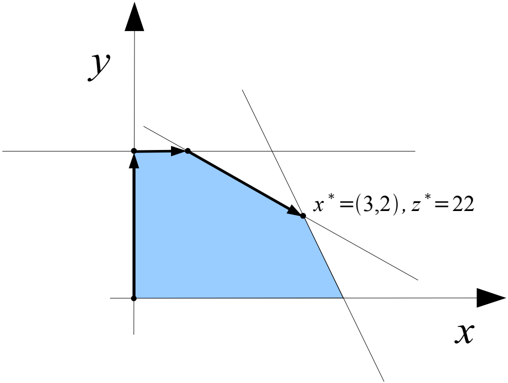
- Méthode de résolution : on part d'un sommet x_0 quelconque, on passe à un sommet voisin pour lequel f augmente, et ainsi de suite.

Remarque : on passe d'un problème **continu** (variables réelles) à un problème **discret** (nombre fini de sommets)...

L'algorithme du simplexe

Illustration 2D : courgettes et navets $z = 4x + 5y$

$x_0 = (0, 0), z = 0 \rightarrow x = (0, 3), z = 15$
$x_0 = (0, 3), z = 15 \rightarrow x = (1, 3), z = 19$
$x_0 = (1, 3), z = 19 \rightarrow x = (3, 2), z = 22$



- plus d'amélioration locale possible $\Rightarrow$ optimum

L'algorithme du simplexe

Illustration concrète

- Standardisation :

Maximiser $z = 4x + 5y$

$$\text{s.c.} \begin{cases} 2x + y \leq 8 \\ x + 2y \leq 7 \\ y \leq 3 \\ x, y \geq 0 \end{cases}$$

Maximiser $z = 4x + 5y$

$$\text{s.c.} \begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \\ x, y, e_1, e_2, e_3 \geq 0 \end{cases}$$

Base initiale ? $\{e_1, e_2, e_3\}$ par exemple :

$$\begin{cases} 2x + y + e_1 = 8 \\ x + 2y + e_2 = 7 \\ y + e_3 = 3 \end{cases} \Leftrightarrow \begin{cases} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{cases}$$

e_1, e_2, e_3 = variables de base, x, y = variables hors base

L'algorithme du simplexe

Solution de base associée

- on met les variables hors base à 0
- on en déduit :
  - valeur des variables de base
  - valeur de z

$$\text{ici : } x = y = 0 \Rightarrow \left\{ \begin{array}{l} e_1 = 8 - 2x - y = 8 \\ e_2 = 7 - x - 2y = 7 \\ e_3 = 3 - y = 3 \end{array} \right. \quad \text{et } z = 4x + 5y = 0$$

L'algorithme du simplexe

Changement de base

Observation essentielle : $z = 4x + 5y = 0 \Rightarrow$ on peut augmenter z si x ou y rentre dans la base.

Essayons avec y : quelle est la valeur max que pourra avoir y ?

- $e_1 = 8 - 2x - y \geq 0 \Rightarrow y \leq 8$
- $e_2 = 7 - x - 2y \geq 0 \Rightarrow y \leq 3.5$
- $e_3 = 3 - y \geq 0 \Rightarrow y \leq 3$

Bilan : $y_{\max} = 3$, pour $y = y_{\max}$ on a $e_1 = 5 - x$, $e_2 = 1 - x$, et $e_3 = 0$

▶ candidat pour une nouvelle base :
$\{e_1, e_2, e_3\} \cup \{y\} \setminus \{e_3\} = \{e_1, e_2, y\}$

L'algorithme du simplexe

Nouvelle base $\{e_1, e_2, y\}$

$$\left\{\begin{array}{l} e_1 = 8 - 2x - y \\ e_2 = 7 - x - 2y \\ e_3 = 3 - y \end{array}\right. \Rightarrow \left\{\begin{array}{l} e_1 = 8 - 2x - y = 5 - 2x + e_3 \\ e_2 = 7 - x - 2y = 1 - x + 2e_3 \\ y = 3 - e_3 \end{array}\right.$$

Exprimons z en fonction des variables hors base

- $z = 4x + 5y = 15 + 4x - 5e_3$

Solution de base associée

$$x = e_3 = 0 \Rightarrow \left\{\begin{array}{l} e_1 = 5 - 2x + e_3 = 5 \\ e_2 = 1 - x + 2e_3 = 1 \\ y = 3 - e_3 = 3 \end{array}\right. \quad \text{et} \quad z = 15$$

L'algorithme du simplexe

Itération

$z = 15 + 4x - 5e_3$ peut encore augmenter si x entre dans la base

Si x entre, qui sort ?
Valeur max de x :

- $e_1 = 5 - 2x + e_3 \geq 0 \Rightarrow x \leq 2.5$
- $e_2 = 1 - x + 2e_3 \geq 0 \Rightarrow x \leq 1$
- $y = 3 - e_3 \geq 0 \Rightarrow$ aucune contrainte sur x

Bilan : $x_{\max} = 1$ et e_2 sort.
Nouvelle base $\{e_1, y, x\}$

$$\begin{cases} e_1 = 3 + 2e_2 - 3e_3 \\ x = 1 - e_2 + 2e_3 \\ y = 3 - e_3 \\ z = 19 - 4e_2 + 3e_3 \end{cases}$$

L'algorithme du simplexe

Itération (suite)

$z = 19 - 4e_2 + 3e_3$ peut encore augmenter si e_3 entre dans la base

Si e_3 entre, qui sort ?
Valeur max de e_3 :

- $e_1 = 3 + 2e_2 - 3e_3 \geq 0 \Rightarrow e_3 \leq 1$
- $x = 1 - e_2 + 2e_3 \geq 0 \Rightarrow$ aucune contrainte sur e_3
- $y = 3 - e_3 \geq 0 \Rightarrow e_3 \leq 3$

Bilan : $e_{3_{\max}} = 1$, e_1 sort. Nouvelle base $\{e_3, y, x\}$:

$$\begin{cases} e_3 = 1 + 2/3e_2 - 1/3e_1 \\ x = 3 + 1/3e_2 - 2/3e_1 \\ y = 2 - 2/3e_2 + 1/3e_1 \\ z = 22 - 2e_2 - e_1 \end{cases}$$

L'algorithme du simplexe

Terminaison

On a $z = 22 - 2e_2 - e_1$, donc $z^* \leq 22$

Or la solution de base $x = 3, y = 2, e_3 = 1$ donne $z = 22$

- optimum

La condition de terminaison concerne les coefficients de z exprimée avec les variables hors base.

L'algorithme du simplexe

$$
\begin{array}{lrcrcr}
\max z = & 20x_1 & + & 10x_2 & & \\
s.c. & x_1 & + & 2x_2 & \leq & 120 \\
 & x_1 & + & x_2 & \leq & 100 \\
 & x_1 & & & \leq & 70 \\
 & & & x_2 & \leq & 50 \\
 & x_1, & & x_2 & \geq 0 &
\end{array}
$$

forme standard

$$
\begin{array}{llrcrcrcrcrcrl}
\multicolumn{2}{l}{\max z} & & & & & & & & & & & & \\
\text{s.c.} & z & -20x_1 & - & 10x_2 & & & & & & & & & = 0 \\
 & & x_1 & + & 2x_2 & + & s_1 & & & & & & & = 120 \\
 & & x_1 & + & x_2 & & & + & s_2 & & & & & = 100 \\
 & & x_1 & & & & & & & + & s_3 & & & = 70 \\
 & & & & x_2 & & & & & & & + & s_4 & = 50
\end{array}
$$

L'algorithme du simplexe

Forme standard

$$
\begin{array}{lllllllll}
\max z & & & & & & & & \\
\text{s.c.} \;\; z & -20x_1 & - \; 10x_2 & & & & & = 0 \\
& x_1 & + \; 2x_2 & + \; s_1 & & & & = 120 \\
& x_1 & + \; x_2 & & + \; s_2 & & & = 100 \\
& x_1 & & & & + \; s_3 & & = 70 \\
& & x_2 & & & & + \; s_4 & = 50
\end{array}
$$

Forme tableau

	z	x_1	x_2	s_1	s_2	s_3	s_4	
z	1	-20	-10	0	0	0	0	0
s_1	0	1	2	1	0	0	0	120
s_2	0	1	1	0	1	0	0	100
s_3	0	1	0	0	0	1	0	70
s_4	0	0	1	0	0	0	1	50

L'algorithme du simplexe

Coûts réduits

B, une base de $Ax = b$

la fonction objectif :

$$
\begin{aligned}
z &= cx = c_B x_B + c_N x_N \\
&= c_B B^{-1} b - (c_B B^{-1} N - c_N) x_N \\
&= z_0 - \sum_{j=1}^{n} (c_B B^{-1} a^j - c_j) x_j \\
&= z_0 - \sum_{j=1}^{n} (z_j - c_j) x_j
\end{aligned}
$$

$z_j - c_j = c_B B^{-1} a^j - c_j$ est le coût réduit de la variable hors base x_j

L'algorithme du simplexe

à chaque itération

	z	x_N	x_B	
z	1	coûts réduits	0	z_0
x_B	0 $\vdots$ 0	$\ddots$	Id	$\oplus$

à l'optimum

	z	x_N	x_B	
z	1	$\oplus$	0	z_0^*
x_B	0 $\vdots$ 0	$\ddots$	Id	$\oplus$

L'algorithme du simplexe

Principe heuristique : faire rentrer en base la variable avec le coefficient "le plus négatif" $\rightarrow x_1$

	z	$\downarrow$ x_1	x_2	s_1	s_2	s_3	s_4	
z	1	-20	-10	0	0	0	0	0
s_1	0	1	2	1	0	0	0	120
s_2	0	1	1	0	1	0	0	100
s_3	0	1	0	0	0	1	0	70
s_4	0	0	1	0	0	0	1	50

Qui faire sortir ?

L'algorithme du simplexe

Principe du quotient minimal

colonne pivot x_1	second membre ≥ 0	quotient
$a_1 \leq 0$	b_1	-
$a_2 > 0$	b_2	$\frac{b_2}{a_2}$
$a_3 > 0$	b_3	$\frac{b_3}{a_3}$
$a_4 = 0$	b_4	-

ligne r $\quad \frac{b_r}{a_r} = \min\left\{\frac{b_i}{a_i} | a_i > 0\right\}$ $\quad \rightarrow$ faire sortir s_3

	z	x_1	x_2	s_1	s_2	s_3	s_4	
z	1	-20	-10	0	0	0	0	0
s_1	0	1	2	1	0	0	0	120
s_2	0	1	1	0	1	0	0	100
s_3	0	$\boxed{1}$	0	0	0	1	0	70
s_4	0	0	1	0	0	0	1	50

L'algorithme du simplexe

- exprimer la contrainte z avec les variables hors base x_2 et s_3

$$z - 10x_2 + 20s_3 = 1400$$

- diviser la ligne pivot par le coefficient de la variable entrante

$$x_1 + s_3 = 70$$

- supprimer x_1 des autres contraintes

$$\begin{aligned} 2x_2 + s_1 - s_3 &= 50 \\ x_2 + s_2 - s_3 &= 30 \end{aligned}$$

$$\begin{array}{lccc} & c & \cdots & a \\ & \vdots & & \vdots \\ \text{ligne pivot} - & p & \cdots & b \\ & | & & \\ & \text{colonne pivot} & & \end{array} \Longrightarrow a \to a - \frac{b}{p}c$$

L'algorithme du simplexe

	z	x_1	x_2	s_1	s_2	s_3	s_4	
z	1	0	-10	0	0	20	0	1400
s_1	0	0	2	1	0	-1	0	50
s_2	0	0	1	0	1	-1	0	30
x_1	0	1	0	0	0	1	0	70
s_4	0	0	1	0	0	0	1	50

x_1, s_1, s_2, s_4 en base et x_2, s_3 hors base

sol de base $(70, 0, 50, 30, 0, 50)$ de valeur 1400

Faire rentrer x_2

quotient min $\rightarrow$ faire sortir s_1

L'algorithme du simplexe

	z	x_1	x_2	s_1	s_2	s_3	s_4	
z	1	0	-10	0	0	20	0	1400
s_1	0	0	$\boxed{2}$	1	0	-1	0	50
s_2	0	0	1	0	1	-1	0	30
x_1	0	1	0	0	0	1	0	70
s_4	0	0	1	0	0	0	1	50

	z	x_1	x_2	s_1	s_2	s_3	s_4	
z	1	0	0	5	0	15	0	1650
x_2	0	0	1	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	25
s_2	0	0	0	$-\frac{1}{2}$	1	$-\frac{1}{2}$	0	5
x_1	0	1	0	0	0	1	0	70
s_4	0	0	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	1	25

x_1, x_2, s_2, s_4 en base et s_1, s_3 hors base

sol de base $(70, 25, 0, 5, 0, 25)$ de valeur 1650
optimale car $z = 1650 - 5s_1 - 15s_3$ et $s_1 = s_3 = 0$

L'algorithme du simplexe

Phase II

Données : un programme linéaire et une solution de base admissible
Résultat : une solution de base admissible optimale ou déclarer "PL non borné"

1. Choix d'une colonne (variable) entrante
   - choisir une variable hors base x_j (colonne) ayant un coût réduit négatif
   - s'il n'existe pas de colonne entrante : STOP, la solution de base est optimale
2. Choix d'une ligne (variable) sortante
   - Choisir une ligne r minimisant le quotient
   - s'il n'existe pas de ligne sortante : STOP le tableau courant est non borné
3. Mise à jour de la base et du tableau
   - pivoter autour de a_{rj} et retourner en (1)

L'algorithme du simplexe

- Solution de base dégénérée si une ou plusieurs variables de base sont zéros (plus de bijection entre les solutions de base admissibles et les points extrêmes)
- Si toutes les solutions de base admissibles sont non dégénérées, l'algorithme du simplexe termine après un nombre fini d'itérations

L'algorithme du simplexe

Phase I

Feuille de TD : Programmation linéaire

- Exercice Phase 1 du simplexe

Dualité

Plan

Dualité

Nouveau concept en Programmation Linéaire

Primal

- données A, b, c
- minimiser

Dual

- mêmes données A, b, c
- maximiser

Plan

Plan

Problème primal ($\mathcal{P}$)

Une famille utilise 6 produits alimentaires comme source de vitamine A et C

	produits (unités/kg)						demande
	1	2	3	4	5	6	(unités)
vitamine A	1	0	2	2	1	2	9
vitamine C	0	1	3	1	3	2	19
Prix par kg	35	30	60	50	27	22	

But : minimiser le coût total

Modélisation

Problème dual $(\mathcal{D})$ associé à $(\mathcal{P})$

Un producteur de cachets de vitamine synthétique veut convaincre la famille d'acheter ses vitamines.

Quel prix de vente w_A et w_C ?

- pour être compétitif
- et maximiser le profit

Modélisation

Modélisation matricielle

Problème primal

famille : acheter des produits alimentaires à coût minimum et satisfaire la demande en vitamine A et C

Modélisation sous forme matricielle

Problème dual

producteur de vitamines synthétiques : être compétitif vis-à-vis des produits alimentaires comme source de vitamine et maximiser le profit de vente

Modélisation sous forme matricielle

Généralisation de l'illustration économique

	ressource i	demande j
produit j	a_{ij}	c_j
coût i	b_i	

Problème primal (demandeur de produit) : quelle quantité x_i de ressource i acheter pour satisfaire la demande à coût minimum ?

$$\min \sum_i b_i x_i \qquad s.c. \quad \sum_i a_{ij} x_i \geq c_j \quad \forall j$$

Problème dual (vendeur de produit) : à quel prix proposer les produits pour maximiser le profit tout en restant compétitif ?

$$\max \sum_j c_j w_j \qquad s.c. \quad \sum_j a_{ij} w_j \leq b_i \quad \forall i$$

Plan

Comment prouver l'optimalité ?

Objectif : démontrer l'optimalité d'une solution

$$\max z = x_1 + x_2$$
$$\begin{array}{rcrcl} 4x_1 & + & 5x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 6 \\ & & x_2 & \leq & 2 \end{array}$$
$$x_1, x_2 \geq 0$$

Idée : trouver une combinaison valide des contraintes permettant de borner terme à terme la fonction objectif

Comment prouver l'optimalité ?

$\max z = x_1 + x_2$

$$
\begin{array}{rcrcll}
4x_1 & + & 5x_2 & \leq & 20 & \times y_1 \\
2x_1 & + & x_2 & \leq & 6 & \times y_2 \\
 & & x_2 & \leq & 2 & \times y_3 \\
\hline
(4y_1 + 2y_2)x_1 & + & (5y_1 + y_2 + y_3)x_2 & \leq & 20y_1 + 6y_2 + 2y_3 & \\
 & & & \uparrow & & \\
 & & & \multicolumn{2}{l}{y_1, y_2, y_3 \geq 0} &
\end{array}
$$

Finalement,

$$
\begin{array}{lrcrcrcl}
\min & 20y_1 & + & 6y_2 & + & 2y_3 & & \\
\text{s.c.} & & & & & & & \\
 & 4y_1 & + & 2y_2 & & & \geq & 1 \\
 & 5y_1 & + & y_2 & + & y_3 & \geq & 1 \\
 & y_i & \geq & 0 & & & &
\end{array}
$$

(borne sup minimale)

(borner terme à terme l'objectif)

Plan

Forme canonique de dualité

Donnée A, b, c

$$(\mathcal{P}) \quad \left\{ \begin{array}{ll} \min & z = cx \\ s.c. & Ax \geq b \\ & x \geq 0 \end{array} \right.$$

$$(\mathcal{D}) \quad \left\{ \begin{array}{ll} \max & v = wb \\ s.c. & wA \leq c \\ & w \geq 0 \end{array} \right.$$

Tableau des signes

min	**max**
primal	dual
dual	primal
variable ≥ 0	contrainte $\leq$
variable $\lesseqgtr 0$	contrainte $=$
variable ≤ 0	contrainte $\geq$
contrainte $\leq$	variable ≤ 0
contrainte $=$	variable $\lesseqgtr 0$
contrainte $\geq$	variable ≥ 0

L'écriture du Dual est automatique :

- les variables
- la fonction objectif
- les contraintes

Écrire le dual

Écrire le programme dual

$\max z = 4x_1 + 5x_2 + 2x_3$

$$\begin{array}{rcrcrcl} 2x_1 & + & 4x_2 & & & = & 3 \\ & & & & 2x_3 & \geq & 2 \\ 3x_1 & + & x_2 & + & x_3 & \leq & 2 \\ & & x_2 & + & x_3 & \leq & 1 \end{array}$$

$x_1 \geq 0 \quad x_2 \leq 0 \quad x_3 \geq 0$

Plan

Propriétés

Propriété

Le dual du dual est équivalent au primal

vérifier sur un exemple

$$\max z = 2x_1 + 3x_2 + 4x_3$$

$$\begin{array}{ccccccc} 2x_1 & + & x_2 & & & \leq & 3 \\ & & & & x_3 & \geq & 2 \\ 3x_1 & + & x_2 & + & x_3 & \leq & 2 \\ & & x_2 & & & \leq & 1 \end{array}$$

$$x_1, x_2, x_3 \geq 0$$

Propriétés

$$(\mathcal{P}) \quad \left\{ \begin{array}{ll} \min & z = cx \\ s.c. & Ax \geq b \\ & x \geq 0 \end{array} \right. \qquad (\mathcal{D}) \quad \left\{ \begin{array}{ll} \max & v = wb \\ s.c. & wA \leq c \\ & w \geq 0 \end{array} \right.$$

Théorème de dualité faible

Pour chaque paire de solutions admissibles x de $(\mathcal{P})$ et w de $(\mathcal{D})$

$z = cx \geq wb = v$

Conséquence : que se passe-t-il si l'un est non borné ?

Et l'optimalité ?

Certificat d'optimalité

Si

$$z = cx = wb = v$$

pour des solutions admissibles x de $(\mathcal{P})$ et w et $(\mathcal{D})$, alors x et w sont optimales

Théorème de dualité forte

Si $(\mathcal{P})$ a des solutions et $(\mathcal{D})$ a des solutions, alors

$$cx^* = w^* b$$

Propriété des écarts complémentaires

Pour l'exemple des vitamines

- écrire le primal avec les variables d'écart (s_i)
- écrire le dual avec les variables d'écart (t_i)
- trouver une solution du primal optimale
- trouver une solution du dual optimale
- écrire les paires de variables (s_i, w_i) et (x_j, t_j)
- que remarquez-vous ?

Propriété

Propriété des écarts complémentaires

Pour x^* optimale de $(\mathcal{P})$ et w^* optimale de $(\mathcal{D})$ alors

- une contrainte de $(\mathcal{P})$ est serrée à égalité
  OU
- la variable associée à cette contrainte est nulle dans w^*

idem dans l'autre sens

$$x_j t_j = 0 \text{ et } s_i w_i = 0$$

preuve

Propriété des écarts complémentaires

Intérêt Si on connaît x^* optimal de $(\mathcal{P})$, alors on peut trouver y^* en appliquant le théorème des écarts complémentaires (et ainsi prouver l'optimalité de x^*)

essayer sur un exemple

$\max z = x_1 + x_2$

$$\begin{array}{rcrcl} 4x_1 & + & 5x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 6 \\ & & x_2 & \leq & 2 \end{array}$$

$x_1, x_2 \geq 0$

avec $x_1 = 2$ et $x_2 = 2$

Petite philosophie de la dualité

À quoi servent les trois théorèmes de dualité

- Dualité faible : pour faire la **preuve d'optimalité**
- Écarts complémentaires : pour trouver une solution optimale du dual connaissant une solution optimale du primal
- Dualité forte : **garantit** qu'une preuve d'optimalité (utilisant la dualité) est possible

Excel et analyse post-optimale

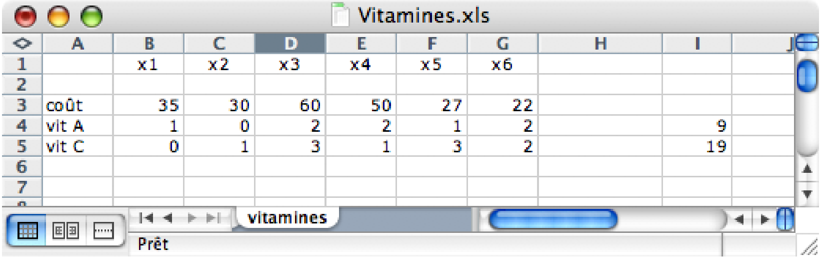
Plan

Plan

Utilisation du solveur d'Excel

Résoudre l'exercice Vitamines avec le solveur d'Excel

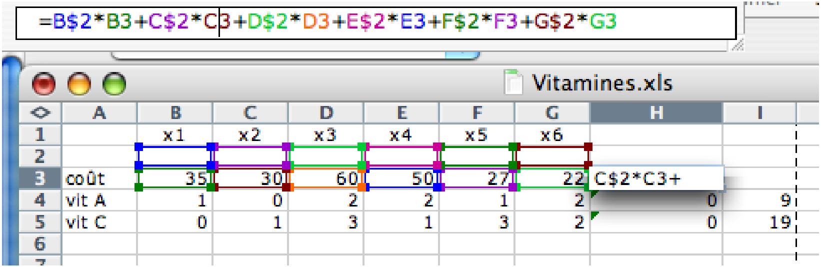
Description des données


Vitamines.xls

	A	B	C	D	E	F	G	H	I
1		x1	x2	x3	x4	x5	x6		
2									
3	coût	35	30	60	50	27	22		
4	vit A	1	0	2	2	1	2		9
5	vit C	0	1	3	1	3	2		19
6									
7									

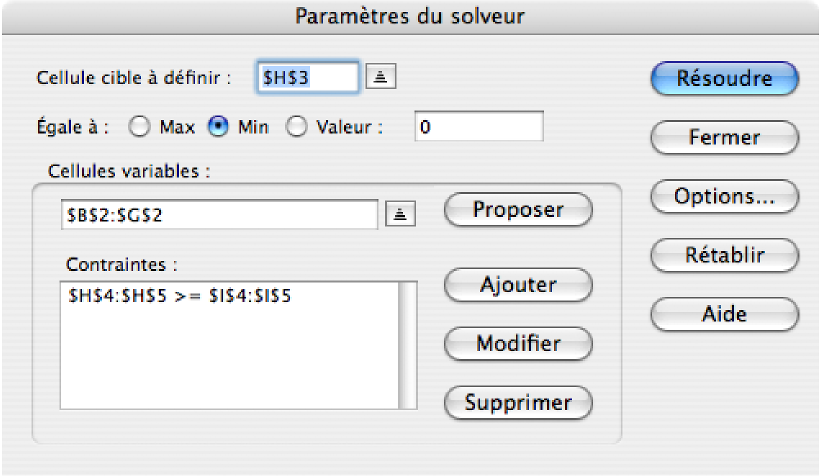
Utilisation du solveur d'Excel

Formules



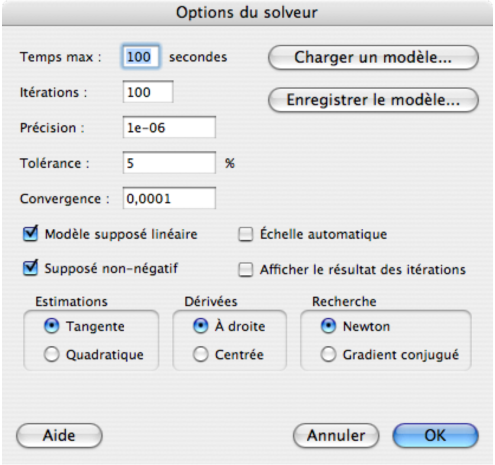
Utilisation du solveur d'Excel

Paramétrage du solveur



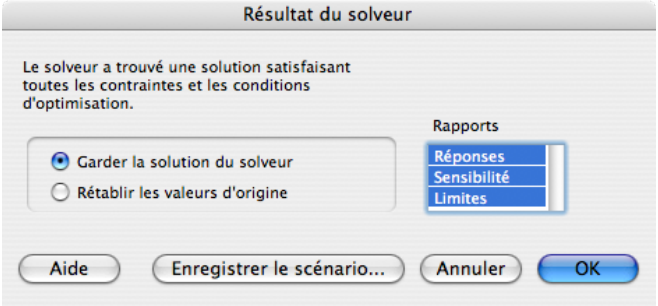
Utilisation du solveur d'Excel

Options du solveur



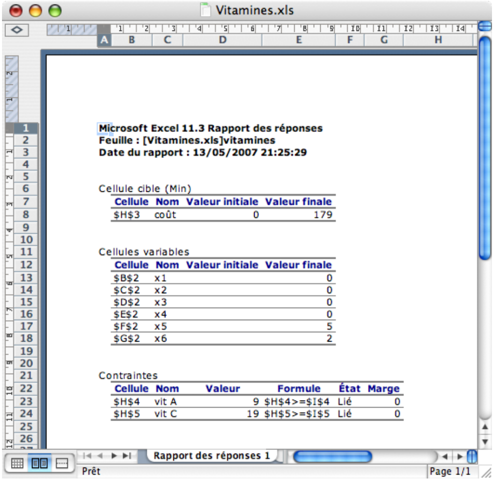
Utilisation du solveur d'Excel

Résultat



Utilisation du solveur d'Excel

Rapport de réponse



Microsoft Excel 11.3 Rapport des réponses
Feuille : [Vitamines.xls]vitamines
Date du rapport : 13/05/2007 21:25:29

Cellule cible (Min)

Cellule	Nom	Valeur initiale	Valeur finale
$H$3	coût	0	179

Cellules variables

Cellule	Nom	Valeur initiale	Valeur finale
$B$2	x1		0
$C$2	x2		0
$D$2	x3		0
$E$2	x4		0
$F$2	x5		5
$G$2	x6		2

Contraintes

Cellule	Nom	Valeur	Formule	État	Marge
$H$4	vit A	9	$H$4>=$I$4	Lié	0
$H$5	vit C	19	$H$5>=$I$5	Lié	0

Utilisation du solveur d'Excel

Rapport de sensibilité

Vitamines.xls

Microsoft Excel 11.3 Rapport de la sensibilité
Feuille : [Vitamines.xls]vitamines
Date du rapport : 13/05/2007 21:25:30

Cellules variables

Cellule	Nom	Finale Valeur	Réduit Coût	Objectif Coefficient	Admissible Augmentation	Admissible Réduction
$B$2	x1	0	32	35	1E+30	32
$C$2	x2	0	22	30	1E+30	22
$D$2	x3	0	30	60	1E+30	30
$E$2	x4	0	36	50	1E+30	36
$F$2	x5	5	0	27	6	16
$G$2	x6	2	0	22	28,8	4

Contraintes

Cellule	Nom	Finale Valeur	Ombre Coût	Contrainte à droite	Admissible Augmentation	Admissible Réduction
$H$4	vit A	9	3	9	10	2,666666667
$H$5	vit C	19	8	19	8	10

Rapport de la sensibilité 1 | Rapp

Prêt Page 1/1

Plan

Analyse post-optimale

On modifie légèrement les coefficients de l'objectif ou des contraintes : doit-on refaire un simplexe ?

- Variation des seconds membres
- Variation des coefficients de la fonction objectif
- Coûts réduits

Analyse post-optimale

Exemple : produire des confitures de rhubarbe et de fraise

- Un pot de rhubarbe nécessite 1kg de rhubarbe et 3kg de sucre et rapporte (marge) 3 euros
- Un pot de fraise nécessite 2kg de fraise et 2kg de sucre et rapporte (marge) 5 euros
- Les quantités disponibles sont 4kg de rhubarbe, 12kg de fraise et 18kg de sucre

Modéliser le problème avec un programme linéaire

Trouver la solution optimale graphiquement

Analyse post-optimale

Résoudre à l'aide du solveur d'Excel

Variation des seconds membres

Si on augmente la capacité disponible d'une ressource, quel est l'impact sur la valeur optimale de la fonction objectif ?

Le prix caché y_i mesure l'augmentation de la fonction objectif si l'on accroît d'une unité la capacité disponible b_i.

Augmenter la quantité de rhubarbe à 5 kg disponibles

- calculer le point optimal
- calculer l'objectif
- calculer le prix caché

Variation des seconds membres

Augmenter la quantité de fraise à 13 kg disponibles

- calculer le point optimal
- calculer l'objectif
- calculer le prix caché

Augmenter la quantité de sucre à 19 kg disponibles

- calculer le point optimal
- calculer l'objectif
- calculer le prix caché

Variation des seconds membres : analyse de sensibilité

Calcul des limites de validité des prix cachés
Jusqu'où peut-on monter (ou descendre) ces valeurs avec les mêmes coûts réduits ?

- De combien peut-on diminuer la quantité de rhubarbe avec le même prix caché ?
- Donner le domaine de validité du prix caché de la rhubarbe.
- Calculez les intervalles pour les fraises et le sucre.
- Pour les contraintes non serrées, quel est le prix caché ?
- Ça vous rappelle quelque chose ?

Variation des coefficients objectifs

Si on augmente le prix de vente unitaire ou si l'on diminue le coût de production unitaire, quel est l'impact sur la valeur de l'objectif ?

La valeur de la j-ème variable à l'optimum (x_j^*) mesure l'augmentation de la fonction objectif si l'on accroît d'une unité la marge unitaire c_j.

Augmenter la marge du pot de rhubarbe à 4 euros

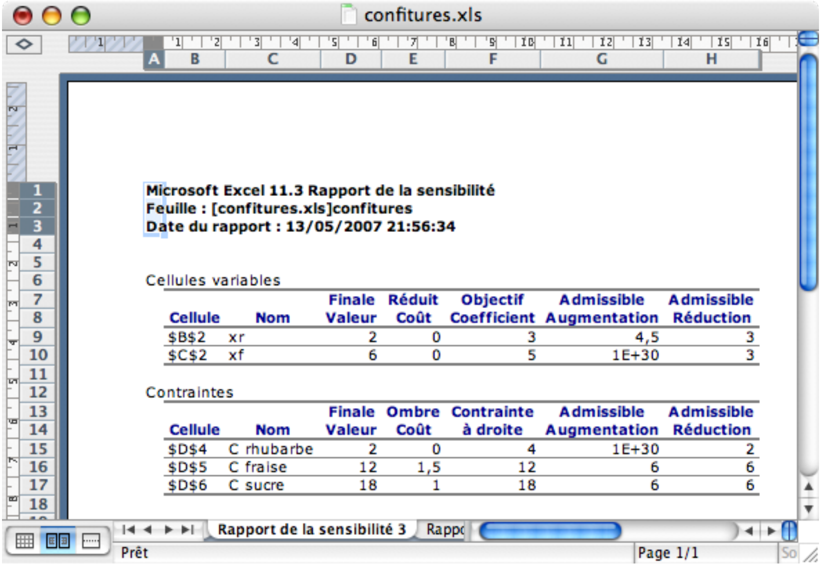
- calculer le point optimal
- calculer l'objectif
- calculer l'augmentation de l'objectif

Variation des coefficients objectifs : analyse de sensibilité

Variation maximum de c_1 autour de 3 tel que le sommet optimal ne change pas.

- De combien peut-on diminuer c_1 ?
- De combien peut-on augmenter c_1 ?
- Idem pour c_2

L'analyse de sensibilité dans Excel



Microsoft Excel 11.3 Rapport de la sensibilité
Feuille : [confitures.xls]confitures
Date du rapport : 13/05/2007 21:56:34

Cellules variables

Cellule	Nom	Finale Valeur	Réduit Coût	Objectif Coefficient	Admissible Augmentation	Admissible Réduction
$B$2	xr	2	0	3	4,5	3
$C$2	xf	6	0	5	1E+30	3

Contraintes

Cellule	Nom	Finale Valeur	Ombre Coût	Contrainte à droite	Admissible Augmentation	Admissible Réduction
$D$4	C rhubarbe	2	0	4	1E+30	2
$D$5	C fraise	12	1,5	12	6	6
$D$6	C sucre	18	1	18	6	6

Plan

Découpe

- Rouleaux de papier de longueur standard 180 cm
- Couteaux de découpe (nombre et position arbitraires)
- Couper des rouleaux de même diamètre
- Liste des commandes pour la prochaine période

longueur	nombre de rouleaux
80	200
45	120
27	130

Trouver les schémas de découpe qui minimisent la perte

Découpe

Étapes de la résolution

- Schémas de découpe
- Variables et contraintes
- Fonction objectif 1, résolution avec Excel et analyse
- Fonction objectif 2, interprétation et résolution avec Excel
- . . . et la contrainte d'intégralité ?

Programmation linéaire en nombres entiers

Introduction

- Programmation Linéaire (PL)
  - Variables de décision continues (réels)
  - Algorithme du Simplexe efficace

- Programmation Linéaire en Nombres Entiers (PLNE)
  - Variables de décision discrètes (entiers, booléens $\{0, 1\}$)
  - Choix d'une bonne formulation souvent difficile
  - Pas de méthode générale efficace de résolution

  $\Rightarrow$ Algorithme de *Branch & Bound*, *Branch & Cut*...

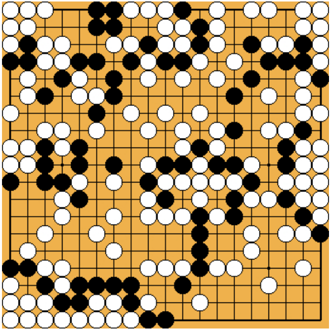
- Programme Linéaire Mixte (MIP pour *Mixed Integer Program*)

  $\Rightarrow$ A la fois des variables réelles et entières

Introduction

Combinatoire

- Structure discrète
- Très grand nombre de possibilités



Introduction

Problème d'optimisation combinatoire

Un problème d'optimisation combinatoire typique

- INSTANCE

  Un ensemble d'objets $1, \ldots, n$, avec des poids c_i

- SOLUTIONS REALISABLES

  Un ensemble $\mathcal{F}$ de parties de $\{1, \ldots, n\}$

- CRITERE

$$\text{maximiser} \quad c(S) = \sum_{i \in S} c_i$$

- L'ensemble $\mathcal{F}$ est en général défini par des contraintes.
- Son cardinal peut être très grand (ici potentiellement 2^n)

Plan

Plan

Problèmes classiques d'optimisation combinatoire

Le sac à dos

Un beau jour de vacances, vous avez décidé de partir en randonnée dans le Vercors. Vous voulez remplir votre sac de capacité 3kg avec les objets les plus utiles :

objets	utilité	poids (g)
carte	10	200
gourde	7	1500
2ème gourde	3	1500
pull	6	1200
Kway	2	500
tomme	4	800
fruits secs	5	700

Problèmes classiques d'optimisation combinatoire

Le sac à dos

Problème générique de SAC À DOS

- un ensemble d'objets $N = \{1, 2 \ldots n\}$
- à chaque objet est associé
  - une utilité u_i
  - un poids w_i
- un randonneur dispose d'un sac-à-dos dont le poids total ne doit pas dépasser W (capacité du sac-à-dos)
- déterminer quels objets prendre pour maximiser l'utilité

Problèmes classiques d'optimisation combinatoire

Le sac à dos

Problème d'optimisation classique

- Utiliser au mieux une capacité
- Choix d'un portefeuille d'investissement

Modélisation

- INSTANCE :
- SOLUTIONS :
- SOLUTIONS REALISABLES :
- CRITERE :

Problèmes classiques d'optimisation combinatoire

Le sac à dos

variables	$x_i = 1$ si l'objet i est choisi, 0 sinon
objectif	$\max \sum_{i \in N} u_i x_i$
contraintes	$\sum_{i \in N} w_i x_i \leq W$
	$x_i \in \{0, 1\} \qquad \forall i \in N$

Problèmes classiques d'optimisation combinatoire

Remplissage de boîtes (bin packing)

Un déménageur souhaite empaqueter des objets en minimisant le nombre de boîtes de capacité $W = 6$ nécessaires

	taille
un livre	2
un autre livre	2
un pull	3
des chaussettes	1
des chaussures	2
des assiettes	5
des verres	6

1. Décrivez une solution réalisable pour le déménageur
2. Proposez une modélisation avec un PLNE

Problèmes classiques d'optimisation combinatoire

Remplissage de boîtes (bin packing)

- des articles $N = \{1, 2 \ldots n\}$ de taille $\{s_1, s_2 \ldots s_n\}$
- à ranger dans des boîtes de capacité W
- en utilisant le moins de boîtes possible

Problèmes classiques d'optimisation combinatoire

Couverture d'ensembles

On souhaite choisir les intervenants dans un projet afin d'avoir toutes les compétences nécessaires en minimisant le coût

	Alice	Babar	Casimir	Donald	Elmer
Coût (h ou €)	10	4	5	6	7
Rech. Op.	1	1	1	0	0
Java	1	0	1	1	0
Bases de données	0	1	1	1	0
Théorie des graphes	1	0	0	0	1
UML	0	1	0	0	1

1. Décrivez une solution réalisable pour le projet
2. Proposez une modélisation avec un PLNE

Problèmes classiques d'optimisation combinatoire

Couverture d'ensembles

- matrice $A = (a_{ij})_{i=1..n,j=1..m}$ à coefficients 0 ou 1
- $c_j > 0$, le coût de la colonne j
- une colonne j couvre une ligne i si $a_{ij} = 1$
- trouver un sous-ensemble des colonnes de A de coût minimum tel que chaque ligne de A soit couverte au moins une fois

Problèmes classiques d'optimisation combinatoire

Partition d'ensembles

- matrice $A = (a_{ij})_{i=1..n, j=1..m}$ à coefficients 0 ou 1
- $c_j > 0$, le coût de la colonne j
- une colonne j couvre une ligne i si $a_{ij} = 1$
- trouver un sous-ensemble des colonnes de A de coût minimum tel que chaque ligne de A soit couverte exactement une fois

Problèmes classiques d'optimisation combinatoire

Affectation

- N_1 et N_2 deux ensembles de même cardinal n
- $A \subseteq N_1 \times N_2$: un collection de couples de nœuds représentant toutes les affectations possibles
- c_{ij} : coût du couple $(i, j) \in A$
- trouver une affectation de coût minimum tel que chaque élément de N_1 est affecté à un et un seul élément de N_2

Problèmes classiques d'optimisation combinatoire

Plus court chemin

- Trouver un chemin de distance minimum entre deux nœuds, s et t d'un réseau donné.

Plan

Techniques générales de modélisation

- La PLNE permet de résoudre beaucoup de problèmes combinatoires
- mais ATTENTION à l'efficacité de la résolution...

Les variables entières sont introduites

- Pour décrire des structures discrètes

  sous-ensemble $S \subseteq \{1, \ldots, n\}$

$\Rightarrow$ vecteur indicateur $(x_1, \ldots, x_n) \in \{0, 1\}^n$

- Pour linéariser des expressions non linéaires

Techniques générales de modélisation

Restriction à un ensemble discret de valeurs

- x doit prendre sa valeur parmi $\{p_1, p_2 \dots p_k\}$
- On introduit une variable y_i indicatrice de $\{x = p_i\}$
  $y_i \equiv 1$ ssi $x = p_i$, et 0 sinon

$$\begin{cases} \displaystyle\sum_{i=1}^{k} y_i = 1 \\ x = \displaystyle\sum_{i=1}^{k} p_i y_i \\ y_i \in \{0, 1\} \quad \text{pour } i = 1, 2 \dots k \end{cases}$$

Techniques générales de modélisation

Contraintes de seuil : si $x > 0$ alors $x \geq K$ (constante)

$$\begin{cases} x \leq My \\ x \geq Ky \\ y \in \{0,1\} \end{cases} \quad \text{où } M \text{ est une constante plus grande que } x$$

Implication logique : $x = 1 \ \Rightarrow \ y = 1$
avec x et y deux variables booléennes $\{0,1\}$

$$x \leq y$$

OU logique : x ou y doit être à $\textsc{Vrai}$
avec x et y deux variables booléennes $\{0,1\}$

$$x + y \geq 1$$

Techniques générales de modélisation

x : une variable de décision

Objectif avec coût fixe (fonction affine) : min $f\mathbf{1}_{\{x>0\}} + cx$

- Le coût est composé d'un coût unitaire c et d'un coût fixe f payé uniquement si $x > 0$
- On introduit une variable y indicatrice de $\{x > 0\}$
  $y \equiv 1$ ssi $x > 0$, et 0 sinon

$$\begin{cases} \min fy + cx \\ x \leq My \\ y \in \{0, 1\} \end{cases}$$

où M est une constante $\geq x$

Techniques générales de modélisation

Contraintes disjonctives

- deux taches de durées d_i et d_j doivent être usinées sur une même ressource
$$\begin{cases} t_i + d_i \leq t_j & \text{si } i \text{ est réalisée avant } j \\ t_j + d_j \leq t_i & \text{si } j \text{ est réalisée avant } i \end{cases}$$

$$\begin{cases} t_i + d_i \leq t_j + M(1 - y_{ij}) \\ t_j + d_j \leq t_i + My_{ij} \\ y_{ij} \in \{0, 1\} \end{cases}$$

Techniques générales de modélisation

Termes quadratiques

- linéariser xx' avec $x, x' \in \{0,1\}$
- On introduit une variable $y \equiv xx'$
  On doit traduire $y = 1$ ssi ($x = 1$ et $x' = 1$)

$$\begin{cases} y \leq x \\ y \leq x' \\ x + x' - 1 \leq y \\ y \in \{0,1\} \end{cases}$$

Plan

Formulation

Problème combinatoire à résoudre

$$\max\{cx \mid x \in X\} \text{ avec } X \subseteq \mathbb{Z}^n$$

Une modélisation du problème en PLNE
$\Rightarrow$ définit un polyèdre $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$

Définition

Un PLNE est une formulation de X ssi $X = P \cap \mathbb{Z}^n$

Illustration graphique

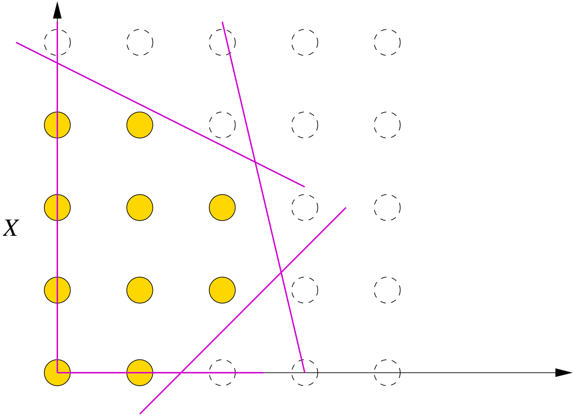
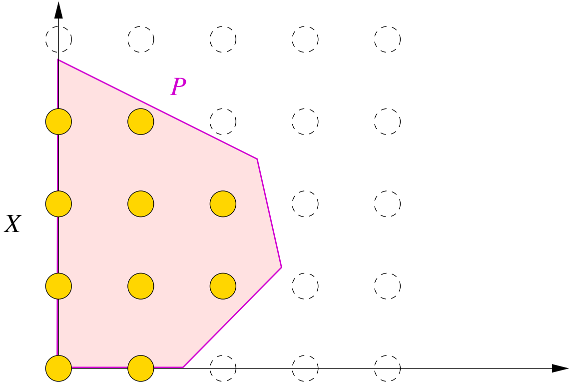


Illustration graphique



Relaxation Linéaire

Pour résoudre un PLNE

- une idée simple est d'oublier que les variables sont entières
- on recherche alors l'optimum du PL sur le polyèdre P
- on peut utiliser l'algorithme du simplexe

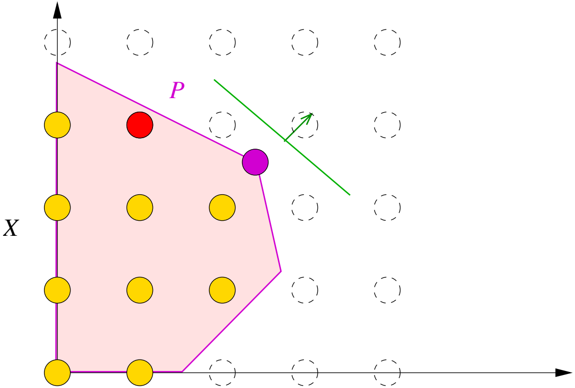
Définition

La relaxation linéaire d'une formulation en PLNE est le PL

$$\max\{cx \mid Ax \leq b \ , \ x \in \mathbb{R}^n\}$$

Lien entre l'optimum du PL et l'optimum du PLNE ?

Illustration graphique de la relaxation



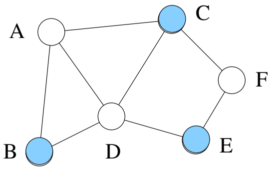
Exemple I

$$
\begin{array}{llcccccl}
\max z = & 4x_1 & + & x_2 & & & & \\
s.c. & 7x_1 & + & x_2 & \leq & 36 & & \\
 & x_1 & + & 4x_2 & \leq & 22 & & \\
 & x_1, & & x_2 & \geq & 0 & & \text{entiers}
\end{array}
$$

1. Trouvez graphiquement l'optimum fractionnaire
2. Trouvez graphiquement l'optimum entier

Exemple II

Stable maximum



- Ensemble S de sommets d'un graphe
- 2 à 2 non adjacent

1. Quel est l'optimum entier sur un triangle ?
2. Quel est l'optimum fractionnaire sur un triangle ?

- la relaxation linéaire donne peu d'indication !

Exemple III

$$
\begin{aligned}
\min z = \quad & x_1 \\
s.c. \quad & x_1 - 17x_2 = 3 \\
& x_1 - 11x_3 = 4 \\
& x_1 - 6x_4 = 5 \\
& x_1, x_2, x_3, x_4 \geq 0 \quad \text{entiers}
\end{aligned}
$$

1. Trouvez l'optimum fractionnaire, son arrondi et l'optimum entier

Propriété de la relaxation linéaire

Pour une formulation en PLNE

$$z^*_{IP} = \max\{cx \mid Ax \leq b\ ,\ x \in \mathbb{Z}^n\}$$

La relaxation linéaire

$$z^*_L = \max\{cx \mid Ax \leq b\ ,\ x \in \mathbb{R}^n\}$$

vérifie

1. $z^*_{IP} \leq z^*_L$
2. Si la solution optimale de la relaxation linéaire est entière, alors c'est aussi une solution optimale pour le PLNE

Plan

Méthodes énumératives

- Nombre fini de solutions

$$\mathcal{F} = \{S_1, S_2, \ldots, S_N\}$$

```
- Parcourir toutes les solutions
- Pour chaque S ∈ F, évaluer c(S)
- Retenir la meilleure solution
```

Problème

Le nombre de solutions potentielles est fini mais gigantesque

Espérance de vie du soleil $\simeq$ 5 milliards d'années $< 2^{58}$ secondes

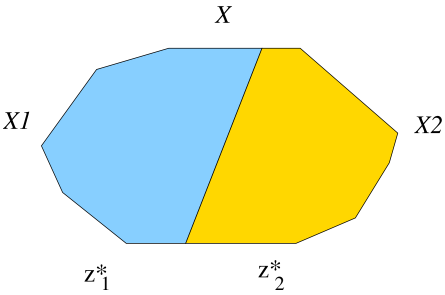
Challenge de l'optimisation combinatoire

Comment trouver la meilleure solution sans parcourir toutes les solutions ?

- Énumération implicite : éliminer *a priori* des solutions
- Détecter que des solutions sont "mauvaises" ou irréalisables sans les évaluer explicitement.

Principe du Branch & Bound

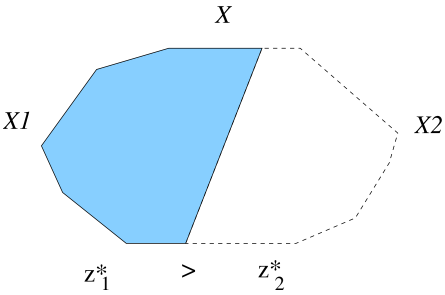
- On veut résoudre $z^* = \max\{cx \mid x \in X\}$
- Si on partitionne X en (X_1, X_2)
- Alors $z^* = \max\{z_1^*, z_2^*\}$



Principe du Branch & Bound

- Si $z_1^* \ > \ z_2^*$
- Alors il est inutile d'explorer le sous-ensemble X_2

$\Rightarrow$ X_2 ne contient pas de solution optimale.



Borne supérieure

- Comment déterminer qu'il est inutile d'explorer X_2 sans calculer z_2^* ?

$\Rightarrow$ Estimation [par excès] de la valeur de z_2^*

Définition

Une fonction des instances dans $\mathbb{R}$ est une *borne supérieure* ssi elle est supérieure à la valeur optimum pour chaque instance.

Pour un PLNE, une borne supérieure est donnée par sa relaxation linéaire

Énumération arborescente implicite

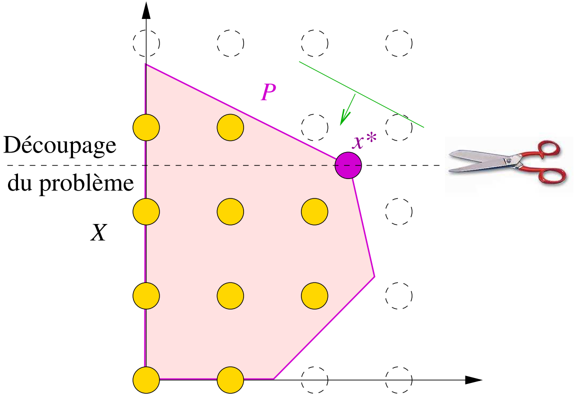
Pour résoudre $z^* = \max\{cx \mid x \in X\}$

- On découpe l'ensemble des solutions X
- Sur chaque $Y \subseteq X$, on calcule une borne supérieure $B(Y)$ de l'optimum $z^*(Y)$.
- Si $B(Y) \leq$ à la meilleure solution trouvée, alors on élague Y
- Sinon on découpe récursivement Y

Comment découper l'espace des solutions ?

On résout la relaxation linéaire du problème sur X à l'optimum

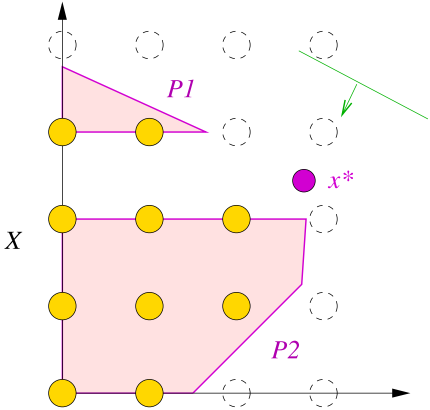
- Si la solution x^* est entière, on a trouvé l'optimum sur X
- Sinon pour une variable (au moins) on a : $a < x_i^* < a + 1$



Branchement sur une variable fractionnaire

On partitionne X en deux nouveaux sous-problèmes :

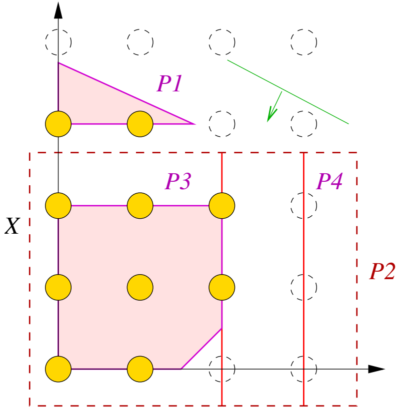
- $X_1 = x \in X$ et $x_i \leq a$
- $X_2 = x \in X$ et $a + 1 \leq x_i$



Exploration de l'ensemble X_2 de solutions

On recherche la meilleure solution sur X_2 :

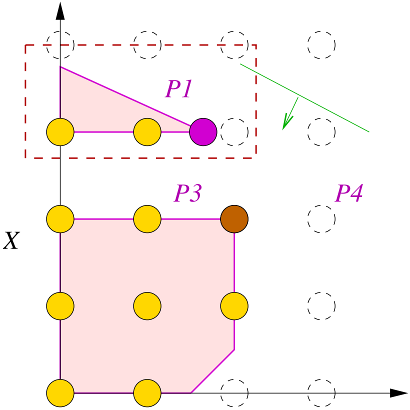
- On résout la relaxation linéaire sur P_2
- On partitionne en 2 nouveaux sous-problèmes



Exploration de l'ensemble X_1 de solutions

On a trouvé la solution optimale sur X_2

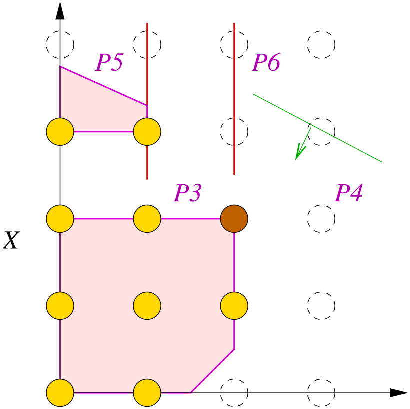
- Existe-t-il une meilleure solution sur X_1 ?
- La borne supérieure ne nous permet pas d'élaguer X_1



Exploration de l'ensemble X_1 de solutions

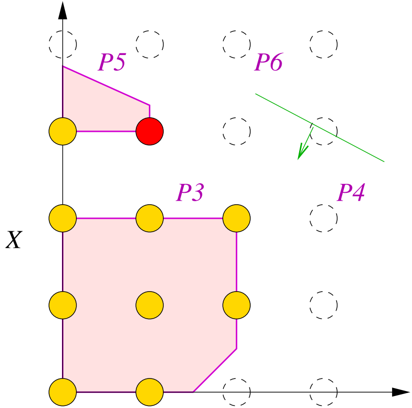
On recherche la meilleure solution sur X_1 :

- On partitionne en 2 nouveaux sous-problèmes



Fin du Branch & Bound

- La solution optimale sur X est la meilleure des 2 solutions trouvées sur X_1 et X_2.



Branch & Bound

1. résoudre la relaxation linéaire
2. brancher sur une variable non entière (à choisir)
   $\rightarrow$ 2 sous problèmes
3. diviser à nouveau un nœud fils en deux ($\neq$ choix possibles)
4. continuer à séparer sur les nœuds dont la valeur est $>$ à la borne inf jusqu'à ce qu'il n'y ait plus de branchement possible

On coupe une branche si

- La relaxation linéaire n'a pas de solution
- la relaxation linéaire donne une solution entière
- la valeur de la borne supérieure est inférieure à la valeur de la meilleure solution entière obtenue

Note : On ne peut rien couper tant qu'on n'a pas de solution disponible

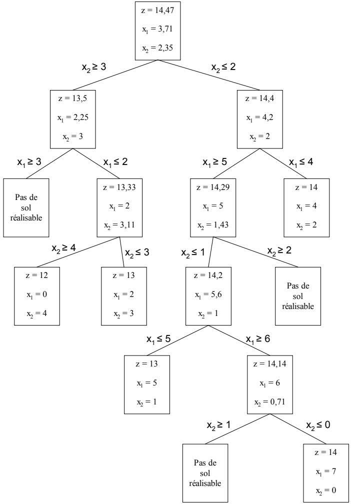
Branch & Bound

$$z = 2x_1 + 3x_2$$

sc.

$$\begin{cases} 5x_1 + 7x_2 \leq 35 \\ 4x_1 + 9x_2 \leq 36 \\ x_1, x_2 \geq 0 \quad \text{entiers} \end{cases}$$

$\boxed{\text{faire le dessin}}$



Approvisionnement des stations service

Une compagnie pétrolière souhaite déterminer les emplacements possibles pour ses dépôts (destinés à fournir ses stations service). Les stations service sont au nombre de n et on a m dépôts. On a un seul produit.

- c_{ij} : coût unitaire de transport entre un dépôt i et la station service j
- f_i : coût fixe d'ouverture du dépôt i
- s_i : capacité du dépôt i
- d_j : demande de la station service j (peut être satisfaite par plusieurs dépôts)

Formulez un programme linéaire qui permet de minimiser les coûts tout en respectant les contraintes.

Application

Mélange de maximum 4 charbons (exo de D. de Wolf)

On mélange des charbons dans un haut fourneau où ensuite, une réaction à haute température produit le coke. Il y a 8 charbons disponibles. Ces charbons sont entrés par des bandes porteuses qui sont au nombre de 4 (au maximum 4 charbons différents dans le mélange). Si un charbon est dans le mélange, il doit l'être à hauteur de minimum 5%. On exige que la teneur du mélange en Silicium soit d'au plus 1,8 %. Le tableau suivant reprend les prix et teneur en Si des charbons.

Charbon	Prix	Teneur Si
Charbon 1	12	2 %
Charbon 2	14	2,5 %
Charbon 3	17	1 %
Charbon 4	10	5 %

Charbon	Prix	Teneur Si
Charbon 5	13	1 %
Charbon 6	9	5 %
Charbon 7	15	2 %
Charbon 8	11	1,5 %

On veut déterminer un mélange qui est de coût minimum.

Application

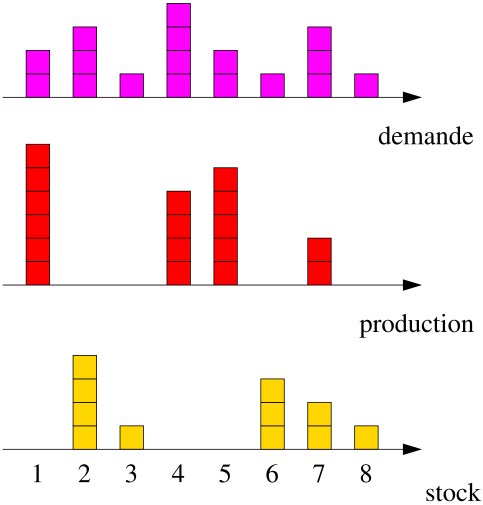
Dimensionnement de lots (DLS)

- Une demande journalière d_t sur un horizon T
- Coût de production $p_t(x) = f_t + a_t x$
- Coût de stockage unitaire h_t (par jour par unité)
- Quel plan de production choisir pour minimiser les coûts ?

1. Comment décrire une solution ?
2. Comment décrire une solution réalisable ?

Application

Dimensionnement de lots (DLS)



Application

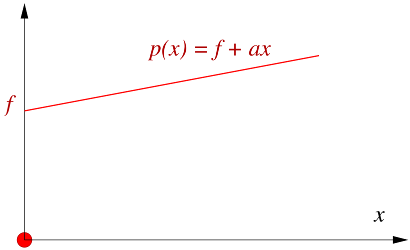
Dimensionnement de lots (DLS)

- Une demande journalière d_t sur un horizon T
- Coût de production $p_t(x) = f_t + a_t x$
- Coût de stockage unitaire h_t (par jour par unité)
- Quel plan de production choisir pour minimiser les coûts ?

Application

Dimensionnement de lots (DLS)

Modélisation du coût de production, non linéaire



Variables de décision

- $y_t \in \{0, 1\}$ indicatrice des instants de production
  $y_t \equiv 1$ ssi $x_t > 0$, et 0 sinon
- Comment traduire le lien entre y et x ?

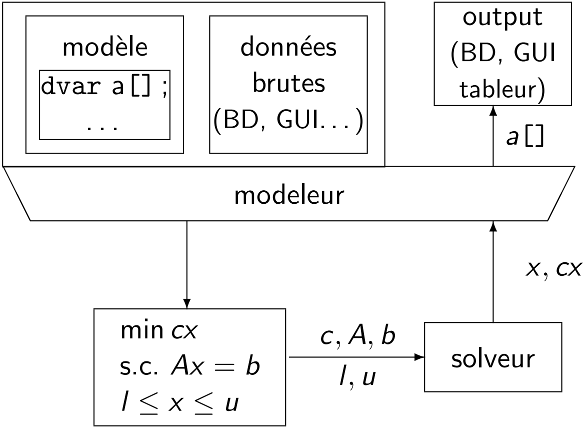
Utiliser un solveur via un modeleur

OPL 5.1 et Cplex

Plan

Plan

Modeleur et solveur



- Solveurs : CPLEX, LPSolve, XPRESS, MINOS, Gurobi...
- Langages de modélisation : GAMS (pionnier), OPL, AMPL, AIMMS...

Le langage de modélisation OPL

- OPL = *Optimization Programming Language*
- Langage pour les problèmes d'optimisation
- Supporte des modèles de programmation mathématiques pour
  - contraintes ou objectifs linéaires ou quadratiques
  - variables entières ou réelles
- Typage avancé pour l'organisation des données
- Se connecte à SGBDR ou tableur
- Script pour récupérer des données et résolutions itératives

L'environnement de développement

IDE : *Integrated Development Environment*

- Organiser des projets
- Saisir des données et des modèles OPL
- Visualiser les données et les solutions
- Contrôler l'optimisation
- + outils pour le debuggage et aide en ligne

Intégrer un modèle dans une application

- Développer un modèle OPL avec OPL IDE
  (modèle et données séparés)
- Compiler dans OPL IDE
- Écrire code dans langage préféré pour
  - générer dynamiquement le fichier de données
  - lire le modèle et les données
  - résoudre le problème
  - récupérer la solution

  (C++, MS.net, Java, ASP.net, JSP)

Plan

Développer un modèle simple

On souhaite produire des confitures de rhubarbe et de fraise

- Un pot de rhubarbe nécessite 1kg de rhubarbe et 3kg de sucre et rapporte (marge) 3 euros
- Un pot de fraise nécessite 2kg de fraise et 2kg de sucre et rapporte (marge) 5 euros
- Les quantités disponibles sont 4kg de rhubarbe, 12kg de fraise et 18kg de sucre.

$$
\begin{array}{lcccccc}
\max & 3x_r & + & 5x_f & & \\
s.c. & x_r & & & \leq & 4 \\
 & & & 2x_f & \leq & 12 \\
 & 3x_r & + & 2x_f & \leq & 18 \\
 & x_r, & x_f & \geq 0 & &
\end{array}
$$

Développer un modèle simple

$$
\begin{array}{lcccccc}
\max & 3x_r & + & 5x_f & & \\
s.c. & x_r & & & \leq & 4 \\
 & & & 2x_f & \leq & 12 \\
 & 3x_r & + & 2x_f & \leq & 18 \\
 & x_r, & x_f & \geq 0 & &
\end{array}
$$

Création du projet "Confitures" puis, description du modèle

- Les variables de décision
- La fonction objectif
- Les contraintes

```
dvar float+ xr ;
maximize 3*xr + 5*xf;
subject to {
   CSucre: 3*xr + 2*xf <= 18;
}
```

Développer un modèle simple

L'éditeur

```
/*********************************************
 * OPL 5.0 Model
 * Author: Nadia
 * Creation Date: 04/05/2007 at 20:42
 *********************************************/

dvar float+ xr ;
dvar float+ xf ;

maximize 3*xr+5*xf ;

subject to{
   cr: xr <= 4 ;
   cf: 2*xf <= 12 ;
   cs: 3*xr+2*xf <= 18 ;
}
```

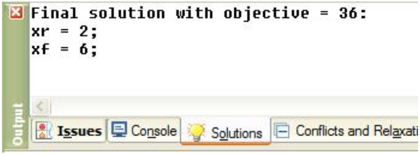
- Commentaires en vert
- Mots clés en bleu

Résoudre un modèle simple

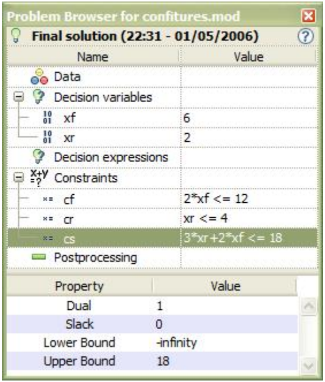
Lancer la résolution et visualiser la solution

Notification

Console



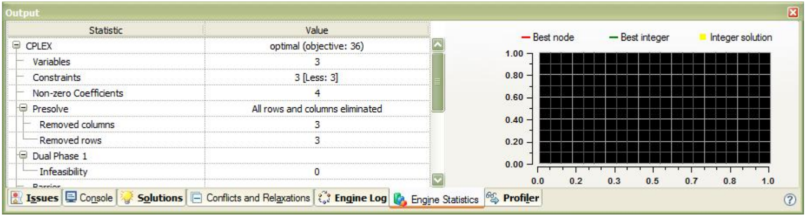
Problem Browser



Plan

L'environnement

Output

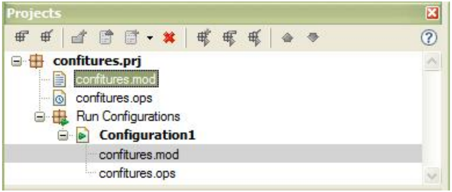


- *Issues*
- *Console*
- *Solutions*
- *Conflicts and Relaxations*
- *Engine Log*
- *Engine Statistics*
- *Profiler*

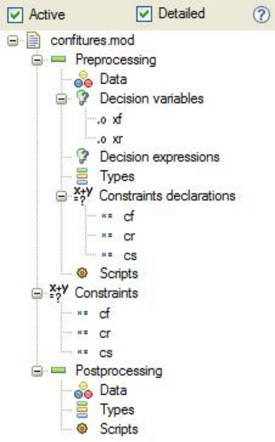
L'environnement

Barres d'outils

Projets (configurations)

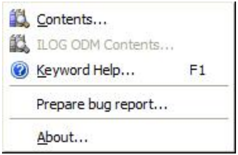


Model Outline



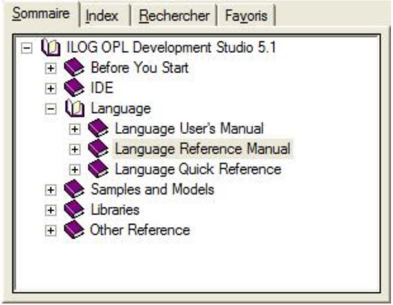
L'aide

Menu Aide



noter l'aide sur un mot clé (*keyword help*)

Sommaire de l'aide



Plan

Séparation du modèle et des données

Dans l'exercice confiture, séparer les données du modèle

Déclaration des données dans le fichier modèle

Produits	`{string} Produits = {"rhubarbe", "fraise", "sucre"};`
Pots	`{string} Pots = {"ConfRhubarbe", "ConfFraise"};`
Profit	`int Profit[Pots] = [3, 5];`
Besoin	`int Besoin[Pots][Produits] = [[1, 0, 3],[0, 2, 2]];`
Quantités dispo.	`int Dispo[Produits] = [4, 12, 18];`

Séparation du modèle et des données

Déclaration des contraintes

```
constraint cap[Produits] ;
```

Déclaration des variables de décision

```
dvar float+ x[Pots] ;
```

Objectif : maximiser le profit

```
maximize sum(po in Pots) Profit[po]*x[po] ;
```

Contraintes : respecter les quantités disponibles

```
subject to{
 forall (pr in Produits)
 cap[pr]: sum(po in Pots)
  Besoin[po][pr]*x[po] <= Dispo[pr] ;
}
```

Séparation du modèle et des données

Dans l'exercice confiture, saisir les données dans un fichier *.dat*

Déclaration des données dans un fichier .dat

```
Produits = {"rhubarbe", "fraise", "sucre"} ;

Pots = {"ConfRhubarbe", "ConfFraise"};

Besoin = ♯[
   "ConfRhubarbe":[1, 0, 3],
   "ConfFraise":[0, 2, 2]
]♯;

Profit = [3, 5];

Dispo = [4, 12, 18] ;
```

Séparation du modèle et des données

Déclaration de données externes dans un modèle

```
{string} Produits = ... ;
{string} Pots = ...;
int Besoin[Pots][Produits] = ...;
int Profit[Pots] = ...;
int Dispo[Produits] = ...;
```

Ajouter le fichier de donnés au projet

Ajouter le fichier de données à la configuration

Debuggage

Outils de debuggage des modèles :

- Décrire des contraintes avec données
- Tracer l'exécution
- Utiliser le graphique de *Engine Statistics*
- Mettre en pause pour voir solution courante

Plan

Production de moteurs d'avions

Production de deux composantes (A et B) d'un moteur d'avion.

- Notification des besoins pour les trois prochains mois.

	avril	mai	juin
A	1000	3000	5000
B	1000	500	3000

- capacités

	machine (h)	hommes (h)	stock (m^3)
avril	400	300	10 000
mai	500	300	10 000
juin	600	300	10 000

- capacités

	machine (h/unité)	homme (h/unité)	stock (m^3/unité)
A	0.10	0.05	2
B	0.08	0.07	3

Production de moteurs d'avions

Production de deux composantes (A et B) d'un moteur d'avion.

- coûts de production : 20 par unités de A et 10 par unités de B
- coût de stockage : 1,5% de la valeur
- horaire mensuel de base : 225
- coût de l'heure supplémentaire de travail : 10
- stock fin mars : 500 A et 200 B
- stock minimum imposé fin juin : 400 A et 200 B

Trouver un plan de production des trois prochain mois qui minimise les coûts.

Proposer une modélisation mathématique de ce problème

Production de moteurs d'avions

Variables

- production : $x[produit, mois]$
- stock : $s[produit, mois]$
- heures supplémentaires $I[mois]$

Objectif : production + stock + heures supplémentaires
Contraintes

- définition du stock
- stock minimum fin juin
- capacités des machines
- capacités des hommes
- capacités des stocks
- définition des heures supplémentaires

Production de moteurs d'avions

Modéliser ce problème avec OPL et le résoudre avec CPLEX

Solution fractionnaire : coût 224724.2857

	Mars	Avril	Mai	Juin
Produit A	-	500	3000	5400
Produit B	-	2857,14	1214,29	428,671
Stock A	500	0	0	400
Stock B	200	2057,14	2771,43	200
Heures supp		0	10	75

Production de moteurs d'avions

Solution entière : coût 224724.5

	Mars	Avril	Mai	Juin
Produit A	-	500	3000	5400
Produit B	-	2858	1214	428
Stock A	500	0	0	400
Stock B	200	2058	2772	200
Heures supp		0.06	9.98	74.96
Heures supp ent		1	10	75

Production de moteurs d'avions

Programme linéaire avec 15 variables et 20 contraintes

- 2 produits
- 3 mois
- 1 type de machines
- 1 type d'hommes
- 1 type de stockage

Pour aller plus loin

- Données dans Excel
- Ilog script
- Application VB
- Web appli et Java
- AMPL : un autre langage de modélisation
  `http://www.ampl.com`

Formulations et coupes

Plan

Plan

Remplissage de Boite (bin packing)

- Un ensemble de n objets de hauteur h_i
- A ranger dans des boîtes de hauteur H
- Minimiser le nombre de boîtes utilisées

Formulation P en PLNE

- $x_{ij} \equiv 1$ si i est rangé dans la boîte j
- $y_j \equiv 1$ si la boîte j est utilisée

$$\min \sum_{j \in N} y_j$$

$$\begin{cases} \sum_j x_{ij} = 1 & \forall i \in N \\ \sum_i h_i x_{ij} \leq H y_j & \forall j \in N \\ y_j, x_{ij} \in \{0,1\} & \forall i,j \in N \end{cases}$$

Remplissage de Boite (bin packing)

- Énormément de symétries sont présentes
- Si l'optimum utilise 3 boîtes, autant prendre les 3 premières !

Quelle contrainte ajouter ?

Résolution des 2 formulations

- Le premier PLNE est une formulation du BinPacking
- Ajouter les contraintes de symétries, n'est-ce pas redondant ?

Essayons de résoudre l'instance

- 15 objets à ranger dans des boîtes de hauteur $H = 20$
- hauteurs | 6 | 7 | 8 | 9 | 10 | en trois exemplaires chacun

- (très) petit exemple
- Quelle est la solution optimale ?

Résolution des 2 formulations

- 15 objets à ranger dans des boîtes de hauteur $H = 20$
- hauteurs $3\times$ $\boxed{6} \boxed{7} \boxed{8} \boxed{9} \boxed{10}$

Résolution sous OPL

Formulation I (Cuts off)

temps	> 3h
nœuds	> 35 millions

Formulation II (Cuts off)

temps	129s
nœuds	500000

Formulation I (Cuts on)

temps	3s
nœuds	2000

Formulations d'un PLNE

Problème combinatoire à résoudre

$$\max\{cx \mid x \in X\} \text{ avec } X \subseteq Z^n$$

Une modélisation du problème en PLNE
$\Rightarrow$ polyèdre $P = \{x \in R^n \mid Ax \leq b\}$

Définition

Un PLNE est une formulation de X ssi $X = P \cap Z^n$

Il existe une infinité de formulations pour un problème

Illustration graphique

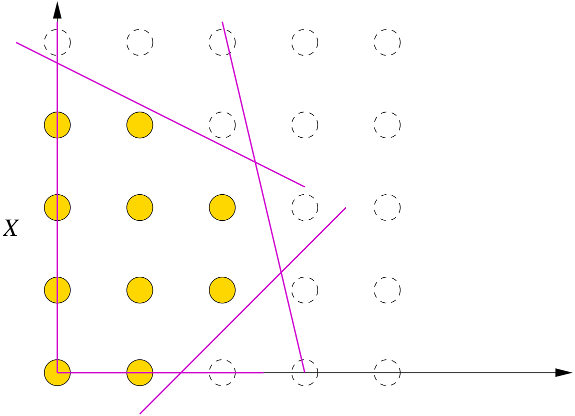
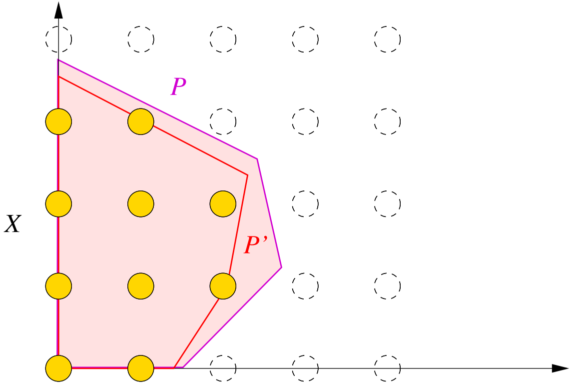
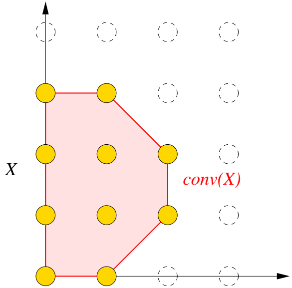


Illustration graphique



Formulation Idéale

- Une formulation P est "meilleure" que P' si $P \subset P'$
- La formulation idéale est la formulation la plus proche de X
- C'est l'enveloppe convexe $conv(X)$



Formulation Idéale

Propriété

$$\max\{cx \mid x \in X\} \;=\; \max\{cy \mid y \in conv(X)\}$$

A gauche, un problème combinatoire (discret)
A droite, un Programme Linéaire (continu)

- Si l'on a une formulation qui décrit $conv(X)$

$\Rightarrow$ la relaxation linéaire résout le problème à l'optimum pour tout objectif linéaire

Moralité

- Dans une formulation en PLNE, il ne faut pas être économe de ses contraintes !

$\Rightarrow$ Améliore les bornes des relaxations linéaires

$\Rightarrow$ Diminue le nombre de nœuds visités

- L'idéal étant que la relaxation donne directement une solution entière sans brancher

Existe-t-il des méthodes pour trouver des contraintes qui améliorent la formulation ?
Peut-on décrire $conv(X)$ **?**

Plan

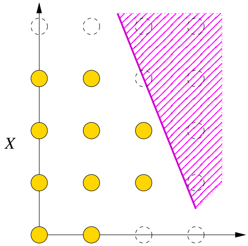
Inégalité valide

Problème combinatoire à résoudre

$$\max\{cx \mid x \in X\} \text{ avec } X \subseteq Z^n$$

Définition

Une inégalité valide est une inégalité $\pi x \leq \pi_0$ vérifiée par tous les points de X



Une remarque

Si on a une inégalité valide

$$y \leq b$$

y une variable entière, b un réel. Alors

$$y \leq \lfloor b \rfloor$$

est aussi une inégalité valide

Cette remarque permet de générer bien des coupes !

Coupes de Chvátal-Gomory

Programme linéaire $\max\{cx \mid Ax \leq b,\ x \text{ entier}\}$. Pour une ligne i de la matrice on a

$$\sum_i a_{ij} x_j \leq b_i$$

Pour tout réel $\lambda > 0$

$$\sum_i \lambda a_{ij} x_j \leq \lambda b_i$$

L'inégalité suivante est donc valide $(x \geq 0)$

$$\sum_i \lfloor \lambda a_{ij} \rfloor x_j \leq \lambda b_i$$

En appliquant la remarque, on obtient une coupe de C-G

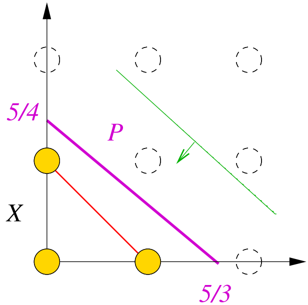
$$\sum_i \lfloor \lambda a_{ij} \rfloor x_j \leq \lfloor \lambda b_i \rfloor$$

Exemple

- Problème à 2 variables x et y entières
- Formulation

$$3x + 4y \leq 5$$

- Objectif $\max 9x + 10y$



Quel est l'optimum de la relaxation linéaire ?
Quel est l'optimum entier ?
Quelles coupes de Chvátal-Gomory trouve-t-on ?

Ajouts de coupes

- Il existe de nombreuses familles de coupes dans la littérature (*Flow Cover, Mixed Integer Rounding, ...*)
- Leur ajout renforce la formulation

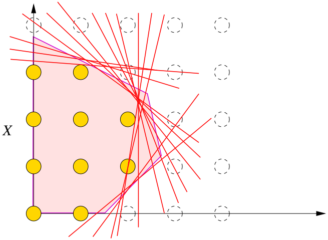
  Mais
- Si le problème est difficile, décrire $conv(X)$ demande un nombre exponentiel de contraintes !

Que faire si une bonne formulation nécessite trop de coupes ?

Plan

Problématique

- Formulation initiale
  $P = \{x \in R^n \mid Ax \leq b\}$
- Famille $\mathcal{F}$ de coupes
- On veut améliorer la formulation pour décrire *conv*(X)



Le plus simple : reformuler en ajoutant $\mathcal{F}$ à P
Le problème : $|\mathcal{F}| >> 1$

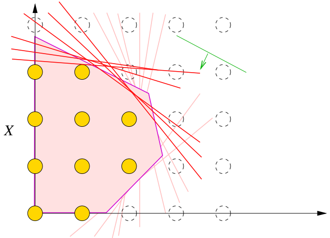
Ajouter toutes les coupes a priori est déraisonnable

Algorithme de Plan Sécant (Cutting Plane)

Problème combinatoire

$$\max\{cx \mid x \in X\} \text{ avec } X \subseteq Z^n$$

- La description complète de $conv(X)$ est inutile
- Seule la description autour de l'optimum nous intéresse



Idée

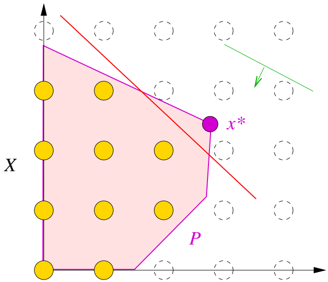
rajouter les inégalités valides uniquement dans la région de l'optimum

Algorithme de Séparation

- Evidemment on ne sait pas où est l'optimum
- On connaît l'optimum x^* de la relaxation linéaire
- Séparation : Trouver une inégalité valide $\pi x \leq \pi_0$ de $\mathcal{F}$ coupant x^* :

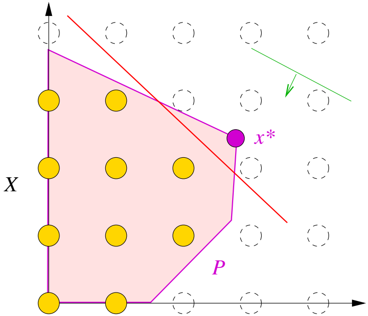
$$\pi x^* > \pi_0$$

- Ajouter cette inégalité pour améliorer la relaxation linéaire

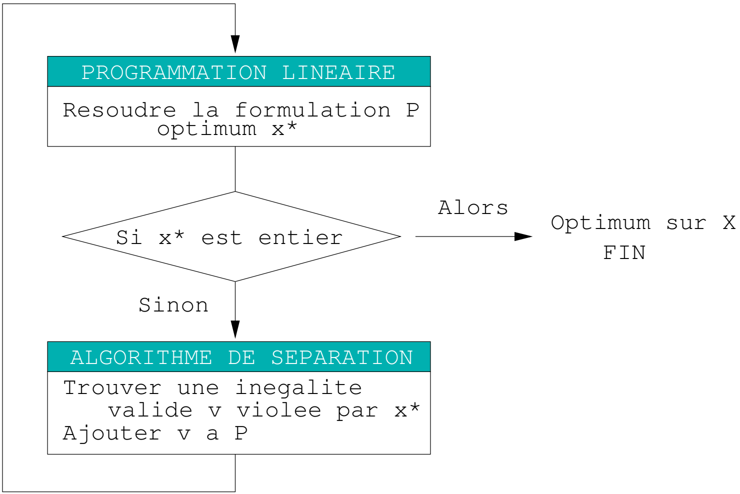


Algorithme de Plan Sécant

- On résout le relaxation linéaire sur la nouvelle formulation
- On cherche une nouvelle inégalité coupant x'^*
- On itère jusqu'à obtenir une solution x^* entière



Algorithme de Plan Sécant



Terminaison de l'algorithme

Un algorithme de Plan Sécant termine

- Soit en trouvant une solution entière : optimum sur X
- Soit en cas d'échec de l'algorithme de séparation

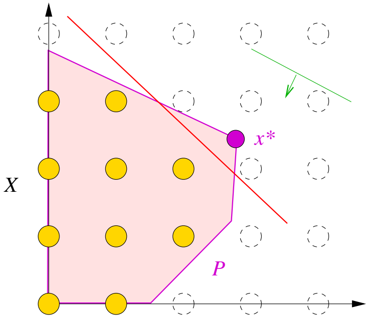
$\Rightarrow$ Aucune inégalité valide de $\mathcal{F}$ n'est violée par x^*

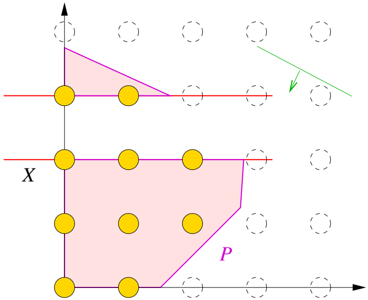
Pour achever la résolution à l'optimum :

- Utiliser un algorithme de *Branch & Bound* standard sur la formulation obtenue

Comparaison avec le *Branch & Bound*

- Algorithme de Plan Sécant : raffine la description du polyèdre autour de l'optimal
- Algorithme de *Branch & Bound* : découpe le polyèdre en morceaux





Branch & Cut

Les algorithmes de plan sécant peuvent échouer

- à séparer une solution fractionnaire
- ou, trop d'inégalités sont nécessaires

Un algorithme de *Branch & Bound* doit alors être utilisé.

Branch & Cut

Un *Branch & Cut* consiste à appliquer un algorithme de plan sécant sur chaque nœud avant de brancher

- But : améliorer la formulation de chaque nœud

$\Rightarrow$ Nombre de nœuds explorés $<<$ *Branch & Bound*

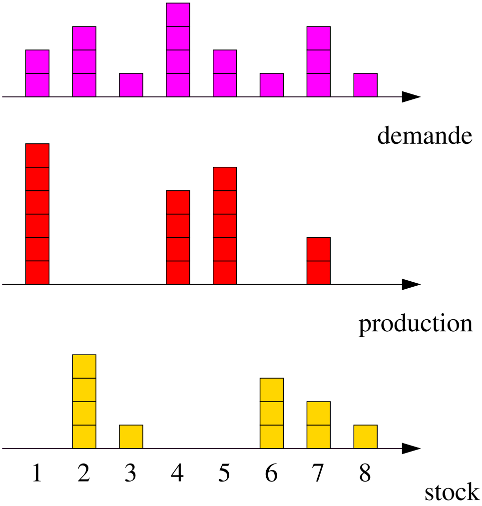
$\Rightarrow$ Calcul de chaque nœud $>>$ *Branch & Bound*

Dimensionnement de lots (DLS)

- Une demande journalière d_t sur un horizon T
- Coût de production $p_t(x) = f_t + a_t x$
- Coût de stockage unitaire h_t (par jour par unité)
- Quel plan de production choisir pour minimiser les coûts ?

1. Comment décrire une solution ?
2. Comment décrire une solution réalisable ?

Dimensionnement de lots (DLS)

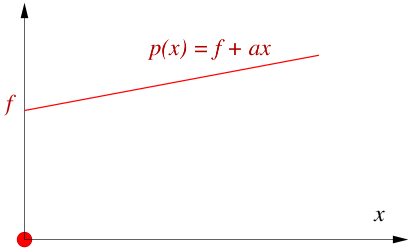


Dimensionnement de lots (DLS)

- Une demande journalière d_t sur un horizon T
- Coût de production $p_t(x) = f_t + a_t x$
- Coût de stockage unitaire h (par jour par unité)
- Quel plan de production choisir pour minimiser les coûts ?

Dimensionnement de lots (DLS)

Modélisation du coût de production, non linéaire



Variables de décision

- $y_t \in \{0, 1\}$ indicatrice des instants de production
  $y_t \equiv 1$ ssi $x_t > 0$, et 0 sinon
- Comment traduire le lien entre y et x ?

Formulations d'un PLNE

On obtient la formulation AGG

$$\min_t f_t y_t + hI_t$$

$$\begin{cases} x_t + I_t = d_t + I_{t+1} & t = 1, \ldots, T-1 \\ x_T + I_T = d_T & \\ x_t \leq D_t y_t & t = 1, \ldots, T \\ y_t \in \{0, 1\} & t = 1, \ldots, T \end{cases}$$

Que se passe-t-il si on essaie de la résoudre ?

Limite du *Branch & Bound*

OPL ne parvient pas à résoudre ! Pourtant :

- Le problème est "facile" et l'exemple est petit

$\Rightarrow$ Il existe des algorithmes qui la résolvent instantanément

- La formulation naturelle n'est pas efficace

$\Rightarrow$ Peut-on formuler différemment le problème ?

Formulation UFL

Formulation moins naturelle

Variables de décision

- $y_t \in \{0, 1\}$ indicatrice des instants de production
- x_{uv} fraction de la demande de v produite le jour u
- Contraintes ?

Comparaison des 2 formulations

Formulation AGG

- $\mathcal{O}(T)$ variables binaires et continues
- $\mathcal{O}(T)$ contraintes

Formulation UFL

- $\mathcal{O}(T)$ variables binaires
- $\mathcal{O}(T^2)$ variables continues
- $\mathcal{O}(T^2)$ contraintes

La seconde formulation est beaucoup plus grosse

Est-ce le bon critère de comparaison pour un PLNE ?

Formulation UFL

Avec la formulation UFL

- OPL résout sans faire de *Branch & Bound* !

$\Rightarrow$ la relaxation linéaire donne directement l'optimum entier

Si on active les coupes *Flow cover*

$\Rightarrow$ OPL résout la formulation AGG en explorant seulement 5 nœuds !

Que se passe-t-il ?

Conclusion

- L'algorithme de *Branch & Bound* peut être inefficace
- Il est primordial d'avoir une bonne formulation
  - Reformulation a priori, formulation étendue
  - Algorithme de Plan Sécant
  - Algorithme de *Branch & Bound*
- Heureusement, les logiciels commerciaux font du *Branch & Cut* avec des familles génériques de coupes
- Jouer sur le paramétrage peut être utile.
- Enrichir la formulation initiale en connaissant la structure du problème (symétries,. . .) aussi !

Programmation dynamique

Plan

Plan

Jeux introductifs

Pyramide de nombres

Trouver un chemin de haut en bas qui maximise la somme des nombres traversés

```
        3
      7   4
    2   4   6
  8   5   9   3
6   7   4   2   8
```

```
        3                       3
      7   4                   7   4
    2   4   6               2   4   6
  8   5   9   3           8   5   9   3
6   7   4   2   8       6   7   4   2   8
```

Jeux introductifs

Pyramide de nombres

$$
\begin{array}{ccccccccc}
 & & & & 3^{3} & & & & \\
 & & & 7^{10} & & 4^{7} & & & \\
 & & 2^{12} & & 4^{14} & & 6^{13} & & \\
 & 8^{20} & & 5^{19} & & 9^{23} & & 3^{16} & \\
6^{26} & & 7^{27} & & 4^{27} & & 2^{25} & & 8^{26}
\end{array}
$$

Jeux introductifs

Partager un sac de pièces

Partager les pièces suivantes en deux ensembles égaux
$\{5 \quad 9 \quad 3 \quad 8 \quad 2 \quad 5\}$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	v					v											
2	v					v				v					v		
3	v			v		v			v	v			v		v		
4	v			v		v			v	v		v	v		v		v

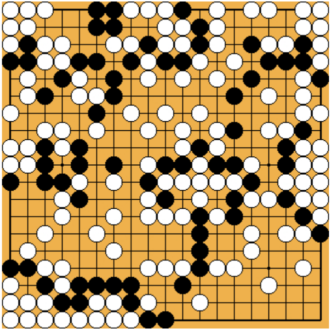
Construire le tableau :

- $m(i,j) = V$ si je peux avoir j avec les i premières pièces
- $m(i,0) = V$ pour $i = 0$ à nb de pièces
- $m(i,j) = m(i-1,j)$ ou $m(i-1, j - pi\grave{e}ce(i))$
  $i = 1$ à nb de pièces $j = pi\grave{e}ce(i)$ à 16.

Plan

Combinatoire

- Structure discrète
- Très grand nombre de possibilités



Problèmes combinatoires

Définition

Un problème d'optimisation se définit par

- INSTANCE : décrit les données d'entrée
- SOLUTIONS REALISABLES : décrit l'ensemble $\mathcal{F}$ des solutions admissibles
- CRITERE à optimiser. Mesure c sur les solutions réalisables

- Définition générique : une infinité d'instances
- On recherche une méthode (algorithme) capable de fournir pour chaque instance I :
  - une solution optimale S^*
  - ou la valeur $OPT(I)$ du critère à l'optimum

$$OPT(I) = c(S^*) = \max\{c(S) | S \in \mathcal{F}\}$$

Problèmes combinatoires

Un problème d'optimisation combinatoire typique

- INSTANCE : Un ensemble d'objets $1, \ldots, n$, avec des poids c_i
- SOLUTIONS REALISABLES : Un ensemble $\mathcal{F}$ de parties de $\{1, \ldots, n\}$
- CRITERE maximiser

$$c(S) = \sum_{i \in S} c_i$$

- L'ensemble $\mathcal{F}$ est en général défini par des contraintes.
- Son cardinal peut être très grand (ici potentiellement 2^n)

Le sac à dos

Un randonneur veut remplir son sac de capacité 4kg avec les objets les plus utiles

objets	utilité	poids (g)
carte	10	200
gourde	7	1500
2ème gourde	3	1500
pull	6	1200
Kway	2	500
tomme	4	800
fruits secs	5	700

Le Sac à dos

Problème d'optimisation classique

- Utiliser au mieux une capacité
- Choix d'un portefeuille d'investissement
- Apparaît dans des problèmes plus complexes

Modélisation

- INSTANCE :
- SOLUTIONS REALISABLES :
- CRITERE :

Méthodes énumératives

- Nombre fini de solutions

$$\mathcal{F} = \{S_1, S_2, \ldots, S_N\}$$

```
- Parcourir toutes les solutions
- Pour chaque S ∈ F, évaluer c(S)
- Retenir la meilleure solution
```

Problème

Le nombre de solutions potentielles est fini mais gigantesque

Espérance de vie du soleil $\simeq$ 5 milliards d'années $< 2^{58}$ secondes

Challenge de l'optimisation combinatoire

Comment trouver la meilleure solution sans parcourir toutes les solutions ?

- Utiliser la structure du problème
- Enumération implicite : éliminer *a priori* des solutions
  Détecter que des solutions sont "mauvaises" ou irréalisables sans les évaluer explicitement.
- Programmation dynamique : réduire l'espace de recherche à des sous-solutions optimales.

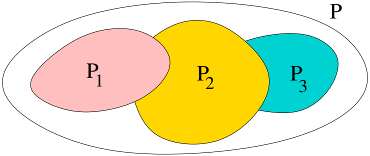
Plan

Principe de sous-optimalité

On veut résoudre un problème P sur une instance I

Structure spécifique de P

Les "morceaux" d'une solution optimale sont optimaux



Le problème P se décompose en sous-problèmes $P_1, \ldots, P_k$. L'optimum sur P s'obtient à partir des optimaux des sous-problèmes.

Principe de sous-optimalité

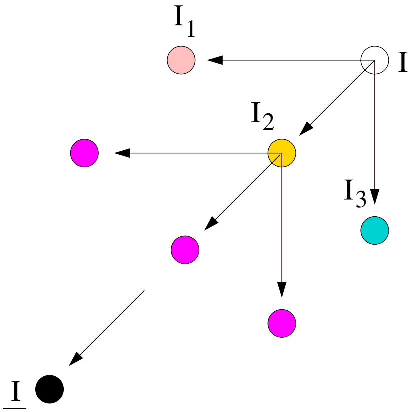
Principe de sous-optimalité

L'optimum sur une instance I peut se construire à partir de solutions optimales sur des instances plus "simples" $I_1, \ldots, I_k$

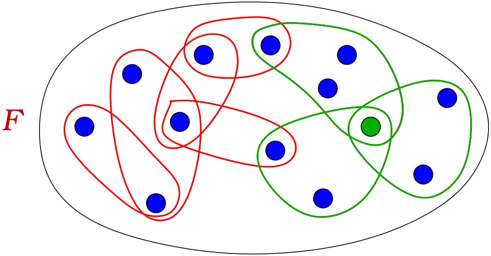
$$OPT(I) = f(OPT(I_1), \ldots, OPT(I_k))$$

- On a une formulation récursive de $OPT(I)$
- Il suffit de calculer l'optimum pour $OPT(I_1), \ldots, OPT(I_k)$ puis d'appliquer f
- Chaque $OPT(I_j)$ s'exprime à son tour en fonction d'instances plus simples
- Jusqu'à obtenir une instance de base $\underline{I}$ directement calculable

Calcul récursif de l'optimum

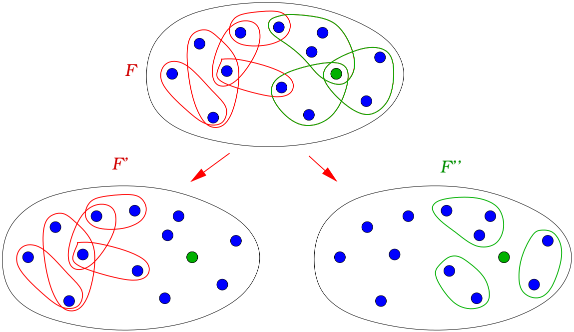


Décomposition en sous-problèmes



- Instance I à résoudre
- Partition des solutions selon l'objet n
  $\mathcal{F}' = \{S \in \mathcal{F} | n \notin S\}$ ne contenant pas n
  $\mathcal{F}'' = \{S \in \mathcal{F} | n \in S\}$ contenant n
- On a $OPT(I) = \max\{c(S'^*), c(S''^*)\}$

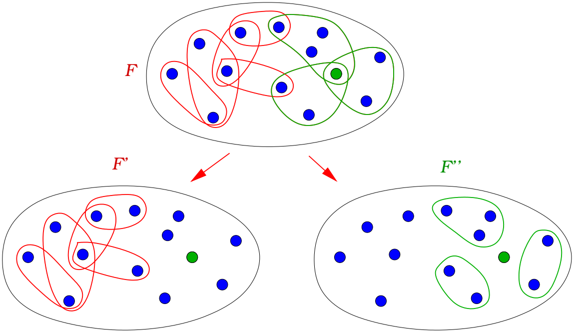
Décomposition en sous-problèmes



Deux sous-problèmes à résoudre

- Sur $\mathcal{F}'$: problème P restreint aux $n-1$ premiers objets
- Sur $\mathcal{F}''$: également restreint aux $n-1$ premiers objets
  mais structure des solutions réalisables ?

Décomposition en sous-problèmes



- Décrire $\mathcal{F}''$ comme $\{S \in \mathcal{F} | n \in S\}$ est inefficace

$\Rightarrow$ énumération explicite de toutes les solutions

- $\mathcal{F}''$ doit pouvoir être décrit comme un sous-problème de P

Sac à dos

Sac à dos

- Instance: n objets de poids w_i et d'utilité u_i, un sac de taille W.
- Solution: sous-ensemble S d'objets tel que $w(S) \leq W$.
- Critere: l'utilité totale $u(S)$ des objets

- Quel est l'optimum de $OPT(I)$ par rapport à l'objet n ?
- Comment écrire le principe de sous-optimalité ?

Paramétrisation

Principe de sous-optimalité : les problèmes qui apparaissent dans la décomposition correspondent au problème initial sur des instances plus simples

- Instance I' pour un sous-problème

$\Rightarrow$ I' diffère de I par certains paramètres (entiers) $p_1, \ldots, p_l$

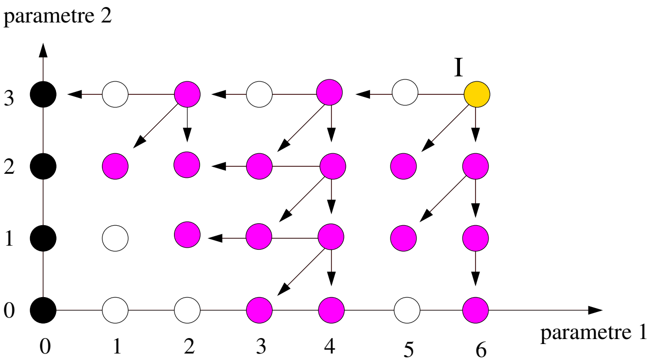
- Pour le Sac à dos : les objets considérés et la taille du sac
- On décrit I' par la valeur de ses paramètres $(x'_1, \ldots, x'_l)$

Définition

On appelle état le vecteur de paramètres $(x_1, \ldots, x_l)$ décrivant une sous-instance.

Graphe d'Etat

- Vecteur de paramètres $(x_1, \ldots, x_l)$: état
- Dépendance entre les instances (calcul de f)



Plan

Programmation Dynamique

Sac à dos

- Instance: n objets de poids w_i et d'utilité u_i, un sac de taille W.
- Solution: sous-ensemble S d'objets tel que $w(S) \leq W$.
- Critere: l'utilité totale $u(S)$ des objets

- Dessinez le graphe d'état pour 4 objets de poids 1 et un sac de capacité 3.
- Que remarque-t-on ?

Programmation Dynamique

- Un état peut être calculé un très grand nombre de fois
- Idée : on dérécursive
- On mémorise les états au lieu de les recalculer
- Il suffit de parcourir les états dans un ordre topologique inverse du graphe d'état

- Evaluer les états de base $OPT[0, \ldots, 0]$.
- Parcourir les états jusqu'à $\bar{X}$
  - Pour chaque état X, dépendant de $X_1, \ldots, X_k$ déjà évalués, mémoriser

$$OPT[X] = f(OPT[X_1], \ldots, OPT[X_k])$$

- Retourner $OPT[\bar{X}]$

Sac à dos

- Sac à dos de taille 7, avec 4 objets
- valeurs des objets

2	4	5	6

- poids des objets

2	3	4	5

- Calculer le tableau OPT

Efficacité

- Quel est le temps de résolution ?
- Dépend
  - du nombre d'états
  - du temps t pour évaluer la fonction f en chaque état.
- Le temps de résolution est alors

$$\sum_{(x_1,\ldots,x_l)\in \text{Etats}} t(x_1,\ldots,x_l)$$

- Souvent on a une borne uniforme sur $t(x_1,\ldots,x_l) \leq T$
- Le temps de résolution est majoré par

$$T \times \#\text{Etats}$$

Sac à dos

Temps de résolution du sac à dos

- Quel est le temps pour évaluer un état (i, w) ?
- Quel est le nombre d'états ?

Calcul d'une solution optimale

La programmation dynamique fournit $OPT(I)$
Comment obtenir une solution S^* ?

- Conserver des pointeurs dans le tableau : chemin dans le graphe d'état
- Méthode de *Backtracking*

Les 2 méthodes consistent à remonter le calcul de $OPT(I)$

Donner une solution optimale pour le sac à dos à partir du tableau OPT de la programmation dynamique

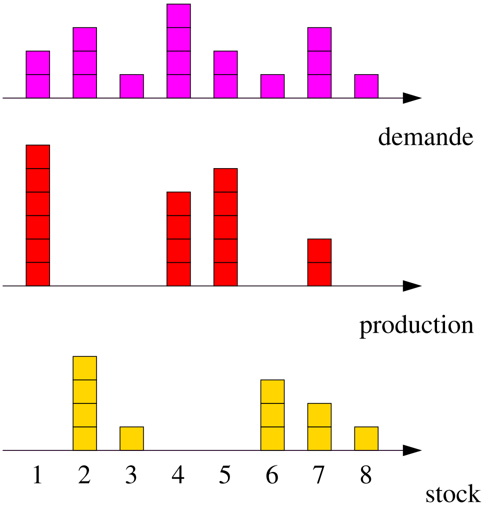
Plan

Dimensionnement de lots

- Une demande journalière d_t sur un horizon T
- Coût de production $p_t(x) = f_t + a_t x$
- Coût de stockage unitaire h_t (par jour par unité)
- Quel plan de production choisir pour minimiser les coûts ?

Comment décrire une solution ?

Dimensionnement de lots



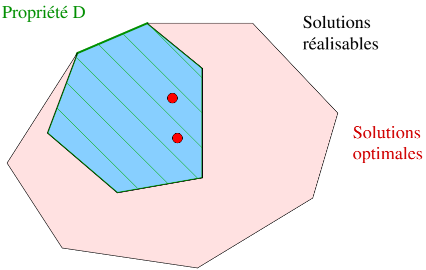
Principe de sous-optimalité

Comment exprimer un principe de sous-optimalité ?
Quels paramètres sont nécessaires ?
Quel est le temps de résolution ?

Dominance

Definition (Dominance)

Une dominance est une propriété $\mathcal{D}$ vérifiée par au moins une solution optimale.



Dimensionnement de lots

Politiques ZIO

Une politique ZIO consiste à ne produire que si le stock est vide

$$\text{si } I_t > 0, \text{ alors } x_t = 0$$

Si pour chaque instant $a_t + h_t \geq a_{t+1}$, alors les politiques ZIO sont dominantes

Argument d'échange

- On considère un planning (optimal) qui ne vérifie pas la dominance
- On montre qu'on peut le modifier en préservant l'objectif

Algorithme de Wagner & Within

- Exprimer un principe de sous-optimalité en utilisant la dominance
- Quel est maintenant le temps de résolution ?

Bilan de la programmation dynamique

- Paradigme pouvant être très efficace
- Pas de condition sur la forme de la fonction objectif...
- ...mais la propriété de sous-optimalité doit être vérifiée
- Gourmand en mémoire
- Devient inopérant si l'espace des états est grand
- Nécessité de trouver des dominances pour le réduire

Méthodologie et études de cas

Plan

Plan

Méthodologie

Face à un problème pratique de décision :

- Comprendre le problème
- En dégager les aspects mathématiques
- Reconnaître un type de problème classique
  - informs `http://www2.informs.org/Resources/`
  - wikipedia (portail RO fait et corrigé par des chercheurs)

Méthodologie

- Analyser la complexité
  - que peut-on espérer pour le temps de résolution imparti ?

  $\Rightarrow$ solution exacte, approchée, avec performance...
  - problèmes NP-complets
    - `http//www.nada.kth.se/~viggo/problemlist/`
  - ordonnancement
    - `http//www.mathematik.uni-osnabrueck.de/research/OR/class/`

Méthodologie

- Proposer une formulation
  - graphes, programmation linéaire, PPC...
- Implémenter une solution
  - solveurs, librairies, algorithmes connus, heuristiques, métaheuristiques, programmation dynamique, programme ad hoc
- Analyser et interpréter les résultats
- Valider par rapport à la demande initiale
- Itérer avec le demandeur si nécessaire

Plan

Découpe

- Rouleaux de papier de longueur standard 180 cm
- Couteaux de découpe (nombre et position arbitraires)
- Couper des rouleaux de même diamètre
- Liste des commandes pour la prochaine période

longueur	nombre de rouleaux
80	200
45	120
27	130

Trouver les schémas de découpe qui minimisent la perte

Découpe

Étapes de la résolution

- Solution manuelle
- Borne inférieure
- Schémas de découpe
- Variables et contraintes
- Fonction objectif 1, résolution et analyse
- Fonction objectif 2, interprétation et résolution
- . . . et la contrainte d'intégralité ?

Plan

Fabrication de charbon

On mélange des charbons dans un haut fourneau où ensuite, une réaction à haute température produit le coke. Il y a 8 charbons disponibles. Ces charbons sont entrés par des bandes porteuses qui sont au nombre de 4 (au maximum 4 charbons différents dans le mélange). Si un charbon est dans le mélange, il doit l'être à hauteur de minimum 5%. On exige que la teneur du mélange en Silicium soit d'au plus 1,8 %. Le tableau suivant reprend les prix et teneur en Si des charbons.

Charbon	Prix	Teneur Si
Charbon 1	12	2 %
Charbon 2	14	2,5 %
Charbon 3	17	1 %
Charbon 4	10	5 %

Charbon	Prix	Teneur Si
Charbon 5	13	1 %
Charbon 6	9	5 %
Charbon 7	15	2 %
Charbon 8	11	1,5 %

On veut déterminer un mélange qui est de coût minimum.

(exo de D. de Wolf)

Plan

Approvisionnement des stations service

Une compagnie pétrolière souhaite déterminer les emplacements possibles pour ses dépôts (destinés à fournir ses stations service). Les stations service sont au nombre de n et on a m dépôts. On a un seul produit.

- c_{ij} : coût unitaire de transport entre un dépôt i et la station service j
- f_i : coût fixe d'ouverture du dépôt i
- s_i : capacité du dépôt i
- d_j : demande de la station service j (peut être satisfaite par plusieurs dépôts)

Déterminer les emplacements des stations services qui permettent de minimiser les coûts pour les données suivantes.

Approvisionnement des stations service

6 dépôts possibles, 7 stations services

dépôt	coût ouverture	capacité
A	7	70
B	8	70
C	4	40
D	28	110
E	20	50
F	10	50

station	demande
1	30
2	30
3	30
4	10
5	20
6	10
7	10

Approvisionnement des stations service

Coûts de transport

	A	B	C	D	E	F
1	10	10	30	35	35	100
2	10	10	25	30	30	95
3	20	10	10	10	30	50
4	100	50	10	10	20	30
5	100	80	30	10	10	10
6	60	60	60	20	10	10
7	30	40	60	20	10	20

Plan

Planification d'expériences

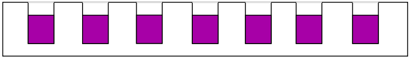
- Dans une industrie chimique, une phase amont teste différents produits de synthèse pour déterminer les meilleures compositions.
- Les réactions se font à température élevée dans un four de cuisson

Le process :

Remplissage	$\rightarrow$	Cuisson	$\rightarrow$	Filtrage
$1/2$ journée		de 3 à 14 jours		2 jours

Cuisson

- Un robot a été acheté pour automatiser la cuisson
- Chaque expérience est chargée dans une barre de cuisson



- On dispose de 8 barres de cuisson
- Le robot peut traiter les 8 barres simultanément
- La température et la durée de chaque barre est programmable.

Remplissage

Cette étape correspond

- A la préparation d'une barre de cuisson
- Au mélange des différents constituants

Pour la réaliser, 3 postes de travail ont été installés, chacun pouvant traiter une barre.

$\Rightarrow$ Un opérateur est requis pour surveiller le déroulement des opérations.

Filtrage

Cette étape correspond

- A l'analyse des résultats de l'expérience

Elle est réalisée de manière semi-automatique

- Un opérateur doit surveiller le déroulement des analyses
- Les 8 barres de cuisson peuvent être analysées simultanément

Opérateur

La présence d'un chimiste qualifié est requise

- Pendant le remplissage
- Pendant le filtrage
- Au démarrage de la cuisson (programmation du robot)
- A la fin de la cuisson

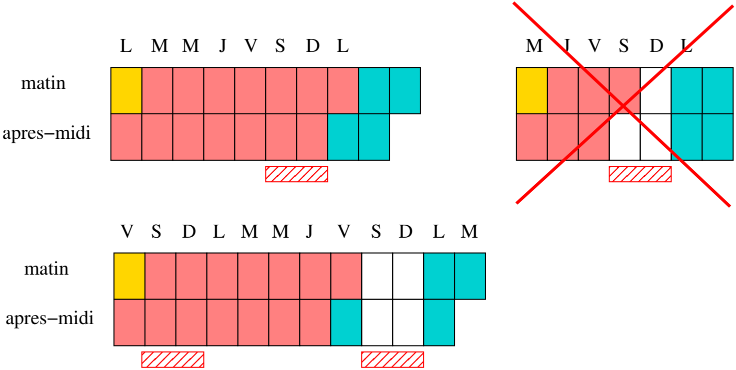
$\Rightarrow$ lancer le filtrage pour arrêter la réaction

$\Rightarrow$ le filtrage peut ensuite être interrompu

Seule la cuisson peut être réalisée sans la présence du chimiste

Disponibilités

Le planning des absences du chimiste est connu à l'avance (week-end, congés, autres obligations)



Les buts de l'industriels

Planifier les expériences à effectuer sur un horizon de l'ordre de 1 mois afin de

- Maximiser l'utilisation du robot (investissement important)
- Finir au plus tôt pour obtenir les résultats des tests

De nouvelles expériences sont à planifier chaque mois

Jeu de données

Vous devez planifier 17 expériences

- 6 avec un temps de cuisson de 14 jours
- 8 avec un temps de cuisson de 7 jours
- 3 avec un temps de cuisson de 3 jours

Le planning des disponibilités de l'opérateur