

Plan du cours

1. Introduction
 - Contenu du cours
2. Logique mathématique
 - Calcul propositionnel
 - Calcul des prédicats
 - Logique floue et aide à la décision
3. Induction
4. Analyse d'algorithmes
 - Comparaison asymptotique de fonctions
 - Complexité
5. Récurrence
6. Mathématique de la gestion
 - Théorie des graphes
 - Optimisation

Programmation linéaire

- Technique utilisée par 85% des entreprises Fortune 500.
- Utilisée par : banques, éducation, foresterie, industrie pétrolière, transport, sidérurgie, compagnies aériennes, ...
- Problème type :
 - Allocation optimale de ressources, disponibles en quantités limitées, à des activités concurrentes.
 - programmation \equiv planification
- Un premier exemple
- D'autres cas réels
- Définitions - Notations
- Solution graphique
- Résultats fondamentaux

Un premier exemple

- Giapetto's Woodcarving, Inc., fabrique deux types de jouets en bois : des soldats et des trains.
- Chaque soldat fabriqué se vend €27 et requiert un dépense de €10 en matières premières. De plus, chaque soldat produit génère €14 de coûts supplémentaires (salaires et frais généraux).
- Un train se vend €21 et coûte €9 en matières premières, plus €10 en coûts supplémentaires.

Un premier exemple

- La fabrication des soldats et des trains demande deux types de main-d'œuvre : menuiserie et finition.
- Un soldat demande 2 heures de travail de finition et 1 heure de menuiserie.
- Un train demande 1 heure de travail de finition et 1 heure de menuiserie.

Un premier exemple

- Hebdomadairement, Giapetto peut disposer de toutes les matières premières nécessaires à la fabrication mais l'entreprise ne dispose que de 100 heures de finition et 80 heures de menuiserie.
- La demande pour les trains est illimitée, mais un maximum de 40 soldats peuvent être vendus chaque semaine.
- Giapetto souhaite maximiser son profit hebdomadaire (revenus - coûts).

Formulation (1)

- Variables de décision :
 - x_1 = nombre de soldats produits chaque sem.
 - x_2 = nombre de trains produits chaque sem.
- Fonction économique (objectif) :
 - Revenus hebdomadaires = $27x_1 + 21x_2$
 - Coûts hebdomadaires = $10x_1 + 9x_2$
 - Autres coûts variables = $14x_1 + 10x_2$
 - Fonction économique à maximiser =

$$z = 3x_1 + 2x_2$$

Formulation (2)

- Contraintes :

- Finition (100h) :

$$2x_1 + x_2 \leq 100$$

- Menuiserie (80h):

$$x_1 + x_2 \leq 80$$

- Soldats (max 40):

$$x_1 \leq 40$$

- Restrictions de signe :

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Programme linéaire (PL)

$$\text{Max } z = 3x_1 + 2x_2$$

$$2x_1 + x_2 \leq 100$$

$$x_1 + x_2 \leq 80$$

$$x_1 \leq 40$$

$$x_1 \geq 0 \quad x_2 \geq 0$$

- Optimisation d'une fonction linéaire sous des contraintes linéaires

Hypothèses

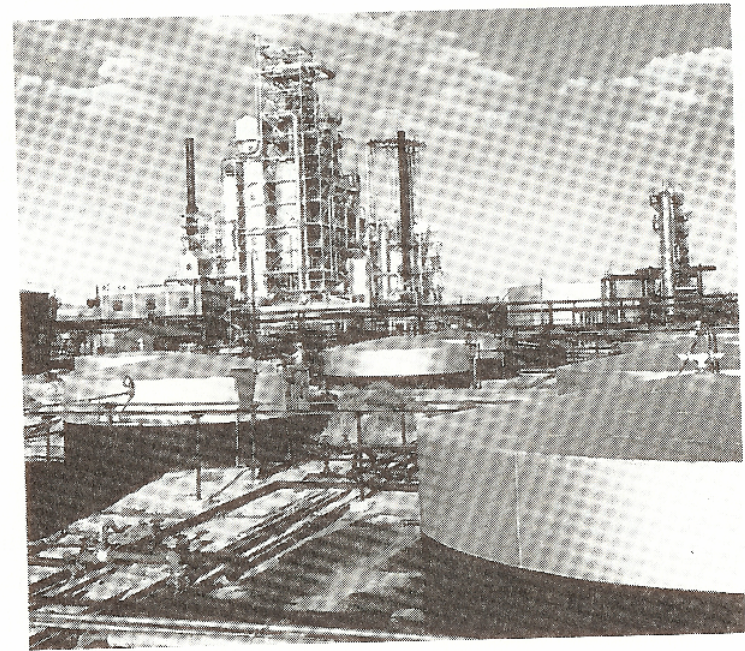
- Proportionnalité: Les contributions de chaque variable à la fonction économique et aux contraintes sont proportionnelles à la valeur prise par cette variable.
- Additivité: Les contributions de chaque variable à la fonction économique et aux contraintes sont indépendantes des valeurs prises par les autres variables.
- Divisibilité: Les variables de décision peuvent prendre des valeurs fractionnaires. (cf. IP)
- Certitude: Chaque paramètre est connu avec certitude.

Quelques exemples réels...

Management Science Application

A Refinery Linear Programming System at Citgo Petroleum

In 1983 Southland Corporation, the parent company of the 7-Eleven convenience store chain, acquired Citgo Petroleum Corporation. Prior to this acquisition, Citgo had lost money for several years; thus, a primary objective of Southland was to improve Citgo's profitability. The Southland Corporation invested in a number of management science applications to achieve this objective, and one of the largest and most important was a refinery linear programming system. The linear programming system allowed for effective management of crude stock acquisition, processing costs, and energy costs, which were almost \$4 billion in 1984. The refinery linear programming system is used routinely to make decisions regarding crude selection and acquisition, refinery run levels, feedstock acquisitions, unit turnaround options, and hydrocracker conversion. The linear programming system is now one of the primary corporate operational planning tools. In 1985 Citgo achieved a pretax profit of over \$70 million, and the linear programming system was cited as a significant contributor to this turnaround.

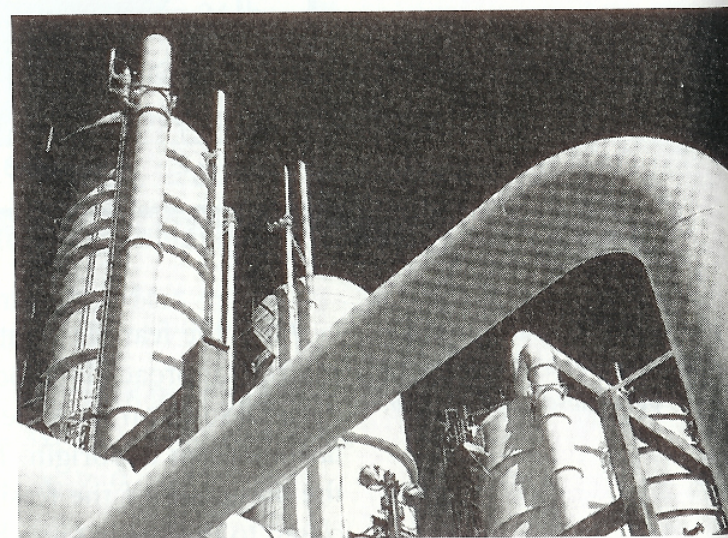


Source: D. Klingman et al., "The Successful Deployment of Management Science Throughout Citgo Petroleum Corporation," *Interfaces* 17, no. 1 (January–February 1987): 4–25.

Management Science Application

Chemical Production at Monsanto

Monsanto produces maleic anhydride (a chemical used in making plastic) at plants in St. Louis and Pensacola. The combined capacity of the two plants is more than 45% of total U.S. output of 359 million pounds per year. Capacity at both plants exceeded demand, resulting in a need to assign production to each plant in an optimal manner. Three linear programming models were developed for this purpose, including a global model to determine the amount produced at each plant and individual plant models to adjust operating plans during a production period. The models, which encompassed over a thousand variables and a dozen or more constraints, minimize cost subject to meeting a production target. Use of the system has resulted in estimated annual savings of between \$1 and \$3 million (depending on plant operating rules).



Source: R. Boykin, "Optimizing Chemical Production at Monsanto," *Interfaces* 15, no. 1 (January–February 1985): 88–95.

Grape Juice Management at Welch's

With annual sales over \$550 million, Welch's is one of the world's largest grape-processing companies. Founded in 1869 by Dr. Thomas B. Welch, it processes raw grapes (nearly 300,000 tons per year) into juice, as well as jellies and frozen concentrates. Welch's is owned by the National Grape Cooperative Association (NGCA), which has a membership of 1,400 growers. Welch's is NGCA's production, distribution, and marketing organization. Welch's operates its grape-processing plants near its growers. Because of the dynamic nature of product demand and customer service, Welch's holds finished goods inventory as a buffer, and maintains a large raw materials inventory stored as grape juice in refrigerated tank farms. Packaging operations at each plant draw juice from the tank farms during the year as needed. The value of the stored grape juice often exceeds \$50 million. Harvest yields and grape quality vary between localities. In order to maintain national quality and consistency in its products, Welch's transfers juices between plants and adjusts product recipes. To do this Welch's uses a spreadsheet-based linear programming model. The juice logistics model (JLM) encompasses 324 decision variables and 361 constraints, that minimizes the combined costs of transporting grape juice between plants and the product recipes at each plant, and the carrying cost of storing grape juice. The model decision variables include the grape juice shipped to customers for different product groups, the grape juice transferred between plants, and inventory levels at each plant. Constraints are for recipe requirements, inventories, and grape juice usage and transfers. During the first year the linear programming model was used, it saved Welch's between \$130 thousand

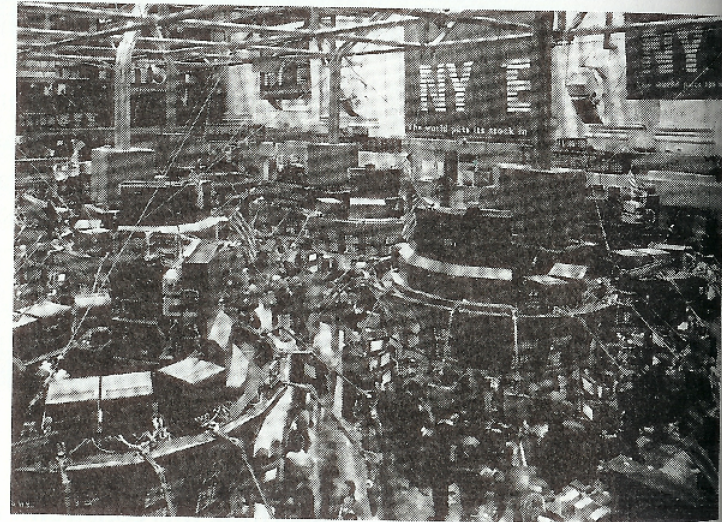


and \$170 thousand in carrying costs alone by showing Welch's it did not need to purchase extra grapes that were then available. The model has enabled Welch's management to make quick decisions regarding inventories, purchasing grapes, and adjusting product recipes when grape harvests are higher or lower than expected, and when demand changes, resulting in significant cost savings.

Source: E. W. Schuster and S. J. Allen, "Raw Material Management at Welch's, Inc.," *Interfaces* 28, no. 5 (September–October 1998): 13–24.

A Linear Programming Model for Optimal Portfolio Selection at Prudential Securities, Inc.

In the secondary mortgage market, government agencies purchase mortgage loans from the original mortgage lenders and pool them to create mortgage-backed securities (MBSs). These securities are traded in capital markets along with other fixed-income securities such as treasury and corporate bonds. The total size of this market for MBSs is well over \$1 trillion. There are a number of types of MBSs. The market for MBSs is maintained by a network of dealers, including Prudential Securities, Inc. These firms and others like it underwrite and issue new market-based securities. This market is somewhat more complex than standard bond investments such as treasury or corporate securities because the principal on mortgages is returned gradually over the life of the security rather than in a lump sum at the end. In addition, cash flows fluctuate because of the homeowner's right to prepay mortgages. In order to deal with these complexities, Prudential Securities has developed and implemented a number of management science models to reduce investment risk and properly value securities for its investors. One such model employs linear programming to design an optimal securities portfolio to meet various investors' criteria under different interest rate environments. The linear programming model determines the amount to invest in different MBSs to meet the client's objectives for portfolio performance. Constraints might include maximum and minimum percentages (of the total portfolio investment) that could be invested in any one or more securities, the duration of the securities, the amount to be invested, and the amount of the different



securities available. This linear programming model and other management science models are used hundreds of times each day at Prudential Securities by traders, salespeople, and clients. The models allow the firm to participate successfully in the mortgage market. Prudential's secondary market trading in MBSs typically exceeds \$5 billion per week.

Source: Y Ben-Dow, L. Hayre, and V. Pica, "Mortgage Valuation Models at Prudential Securities," *Interfaces* 22, no. 1 (January–February 1992): 55–71.

Gasoline Blending at Texaco

The petroleum industry first began using linear programming to solve gasoline blending problems in the 1950s. A single grade of gasoline can be a blend of from three to ten different components. A typical refinery might have as many as 20 different components it blends into four or more grades of gasoline. Each grade of gasoline differs according to octane level, volatility, and area marketing requirements.

At Texaco, for example, the typical gasoline blends are Power and unleaded regular, Plus, and Power Premium. These different grades are blended from a variety of available stocks that are intermediate refinery products such as distilled gasoline, reformate gasoline, and catalytically cracked gasoline. Additives include, among other things, lead and other octane enhancers. As many as 15 stocks can be blended to yield up to eight different blends. The properties or attributes of a blend are determined by a combination of properties that exist in the gasoline stocks and those of any additives. Examples of stock properties (that originally emanated from crude oil) include to some extent vapor pressure, sulfur content, aromatic content, and octane value, among other items. A linear programming model determines the volume of each blend subject to constraints for stock availability, demand, and the property (or attribute) specifications for each blend. A single blend may have up to 14 different characteristics. In a typical blend analysis involving 7 input



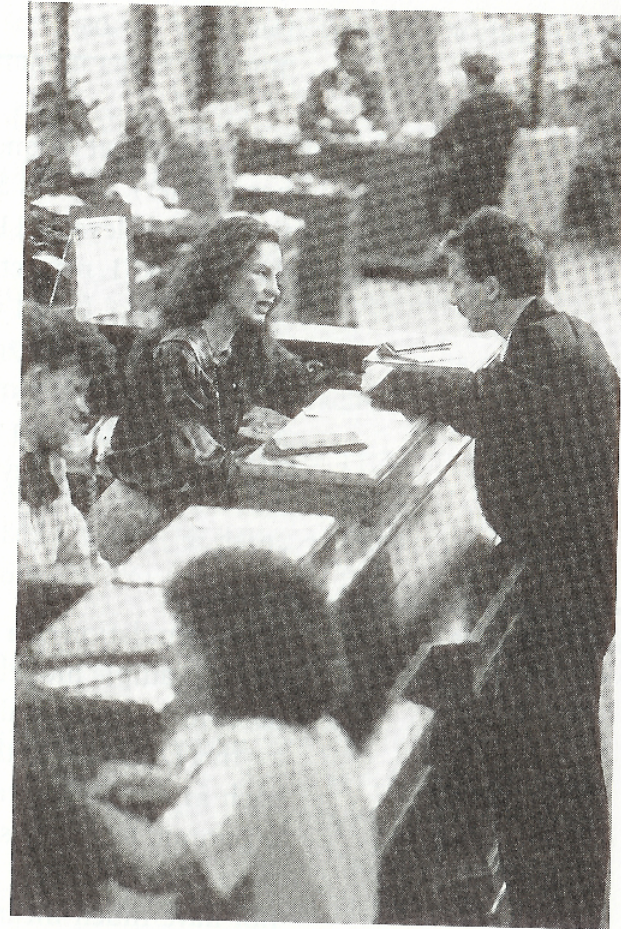
stocks and 4 blends, the problem will include 40 variables and 71 constraints.

Source: C. E. Bodington and T. E. Baker, "A History of Mathematical Programming in the Petroleum Industry," *Interfaces* 20, no. 4 (July–August 1990): 117–27; and, C. W. DeWitt et al., "OMEGA: An Improved Gasoline Blending System for Texaco," *Interfaces* 19, no. 1 (January–February 1989): 85–101.

Analyzing Bank Branch Efficiency with DEA

Data envelopment analysis (DEA) is an application of linear programming used to determine the less productive (i.e., inefficient) service units among a group of similar service units. DEA bases this determination on the inputs (resources) and outputs of the service units.

A major northeastern bank, with 33 branches, used DEA to identify ways to improve its productivity. Bank management did not know how to identify the more efficient branches. The bank identified five resource inputs available to all branches—customer service (tellers), sales service, management, expenses, and office space. Five groups of branch outputs were also identified—deposits, withdrawals, and checking; bank and traveler's checks and bonds; night deposits; loans; and new accounts. The DEA model found that only 10 branches were operating efficiently, whereas 23 branches were using excess resources to provide their volume and mix of services and were, thus, inefficient. The DEA model results provided the basis for reviewing and evaluating branch operations, which revealed operating differences between inefficient branches and best-practice branches. The bank was able to substantially improve its branch productivity and profits, reducing its total branch staff by approximately 20% within one year of the completion of the DEA analysis, and implementing changes in branch operations that resulted in annual savings of over \$6 million.



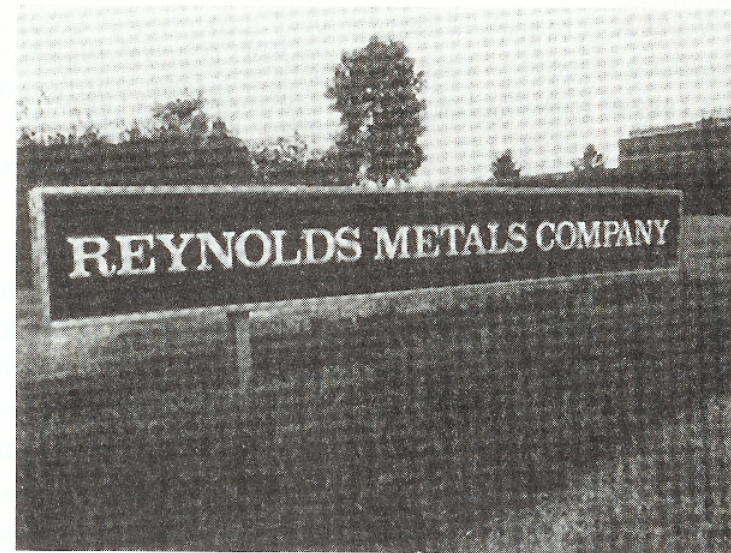
Source: H. D. Sherman and G. Ladino, "Managing Bank Productivity Using Data Envelopment Analysis (DEA)," *Interfaces* 25, no. 2 (March–April 1995): 60–73.

Selecting Freight Carriers at Reynolds Metals Company

In 1991, Reynolds Metals Company spent over \$0.25 billion to deliver its products and receive raw materials through a transportation network that included truck, rail, ocean, and air shipments from over 120 shipping locations to over 5,000 receiving destinations. Truck shipments accounted for over half the company's freight costs, and interstate truck shipments alone cost over \$80 million.

Reynolds Metals encompasses 12 decentralized operating divisions; and consistent with their decentralized operating philosophy, each division and plant traditionally was responsible for negotiating with and selecting its own carriers and arranging for shipments.

However, because of concerns about variability in service quality and high costs, in 1987 the company developed a new central dispatch system located in Richmond, Virginia, to select all (independent) truck carriers and dispatch them centrally. A vital component of this new central dispatch was a mixed integer programming model that selected the number and type of carrier to be further evaluated for final selection with a simulation model. The objective of the model was to optimize central dispatch freight costs. The model optimally selected a specific number of truck carriers and assigned them to shipping locations. Constraints included a limit on the number of carriers to be selected, carrier limits on the number of trucks they would provide Reynolds, and limits on the number of carriers allowed to service individual locations. A typical model had over 5,000 constraints, 200 integer variables, and 9,000 total variables. Based on the



model results, the number of truck carriers used by Reynolds was reduced from over 200 to 14. Savings in freight costs using the entire central dispatch system is over \$7 million annually, and on-time delivery service was increased from 80% to 95%.

Source: E. W. Moore Jr., "The Indispensable Role of Management Science in Centralizing Freight Operations at Reynolds Metals Company," *Interfaces* 21, no. 1 (January–February 1991): 107–29.

Minimizing Color Photographic Paper Waste at Kodak

Kodak (Australasia) Pty. Ltd., a division of Eastman Kodak Company, produces rolls of photographic color paper used to produce color photographs. In producing these rolls, the company loses some of the color paper as waste or “trim loss” in the cutting process, which results in a significant cost. The paper is originally purchased in large bulk rolls from 42 to 52.5 inches wide and up to 8,750 feet in length. The paper also comes in three different blends (i.e., different chemical coatings on the paper that react differently to light). Customers, such as one-hour photo-processing shops, order rolls from 3.5 to 11 inches wide and in lengths from 275 to 1,150 feet. To produce the customer rolls from its bulk rolls, the company must cut an entire bulk roll according to a specific design for length and width, which is referred to as a cutting plan. In order to minimize the amount of waste created when these bulk rolls are cut, Kodak developed a system of mixed integer and 0–1 programming models to determine the cutting plans for customer orders. The variables for the mixed integer model included the number of customer order lots and the number of bulk rolls used for a cutting plan, while the 0–1 integer model variables were for the paper blend for the cutting plan used. General benefits of the models included a reduction in trim waste, an increase in productivity, a reduction in the planning effort for diagramming cutting plans, and a reduced planning



horizon; in addition, the company was able to better match production to customer orders. The savings in the reduction of waste alone was over \$2 million in the first year of operation since waste levels were reduced by 50%.

Source: A. A. Farley, “Planning the Cutting of Photographic Color Paper Rolls for Kodak (Australasia) Pty. Ltd.,” *Interfaces* 21, no. 1 (January–February 1991): 92–106.

Optimal Assignment of Gymnasts to Events

An integer programming model was developed to assign members of a women's gymnastics team to the four events conducted at a typical NCAA meet—vault, uneven bars, balance beam, and floor exercises. Each team can enter up to six gymnasts in each event, and the top five scores among these entrants contribute to the team score. At least four of the entrants must participate in all four events. These conditions formed the model constraints; the objective was to maximize the team's overall expected score. The model was tested at Utah State University and allowed officials to analyze the effects of changing conditions, such as improved performance or injuries, on the team score; to indicate to a team member why she was not selected for a particular event; and to eliminate the time the coach spent manually evaluating different team combinations.

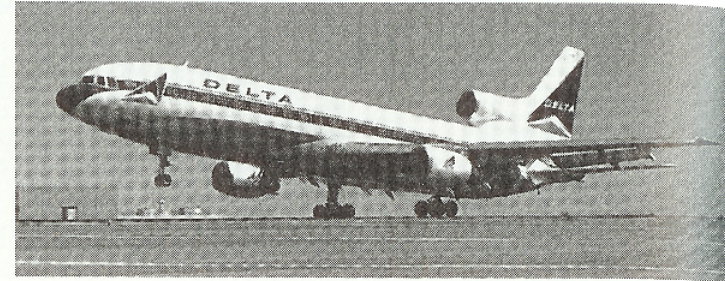


Source: P. Ellis and R. Corn, "Using Bivalent Integer Programming to Select Teams for Intercollegiate Women's Gymnastics Competition," *Interfaces* 14, no. 3 (May–June 1984): 41–46.

Fleeting the Schedule at Delta Airlines

Each day, Delta Airlines flies over 2,500 flight legs in the United States, Canada, and Mexico. Delta uses about 450 different aircraft for these flights, arranged into 10 groups or fleets. The pattern of routes that these aircraft fly through the airline's system is the schedule, which is a crucial aspect of an airline's profitability. The schedule must be designed to maximize revenue potential by eliminating empty seats while simultaneously minimizing operating costs, which have historically been high throughout the aircraft industry. A small change in the daily system flight schedule can result in millions of dollars in revenues or losses. As a result, an airline like Delta is continuously refining and attempting to improve its schedule.

Each Delta fleet is made up of different types of aircraft, and the assignment of a particular set of aircraft to specific markets is called *fleeting the schedule*. In 1991 Delta began the Coldstart project to address the problem of fleeting its schedule. The primary goal of the fleeting process is to have the right plane at the right place at the right time; otherwise, it either will not be able to accommodate customer demand or will fly with empty seats. The results of the Coldstart project was a mixed integer linear programming model that assigns fleet types to flight legs to minimize operating and spill costs, where spill is the number of passengers not carried because of insufficient aircraft capacity.



Model operational constraints include the number of aircraft available in each fleet and scheduling requirements. The model is run daily. It normally contains about 40,000 constraints as well as 60,000 variables, about 40,000 of which are integer. The Coldstart model was implemented in September 1992, and by the summer of 1993 it was saving Delta approximately \$220,000 per day. The model was expected to save Delta approximately \$200 million during its first three years.

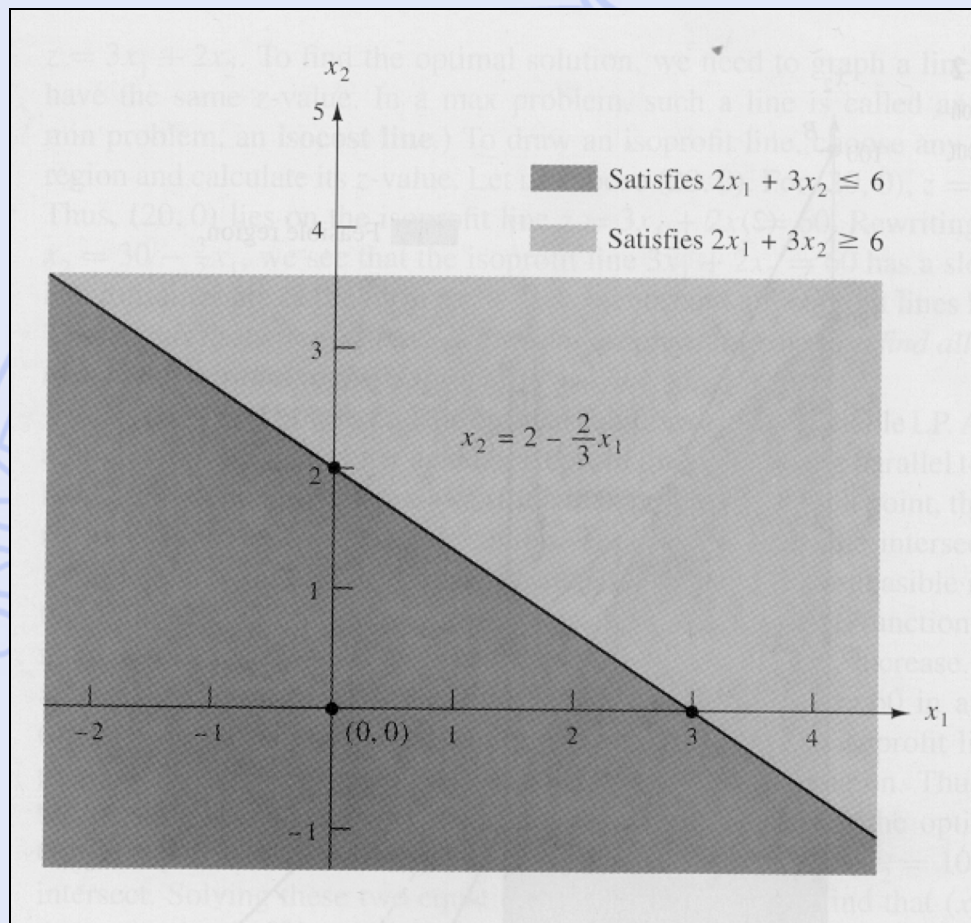
Source: R. Subramanian, et al., "Coldstart: Fleet Assignment at Delta Airlines," *Interfaces* 24, no. 1 (January–February 1994), 104–20.

Définitions

- **Domaine réalisable :**
 - Ensemble de tous les jeux de valeurs des variables de décision satisfaisant toutes les contraintes et restrictions de signe du PL (solutions réalisables ou solutions admissibles).
- **Solution optimale :**
 - Solution réalisable qui optimise (max ou min) la fonction économique.
 - Existence ?
 - Unicité ?

Solution graphique - cas de 2 variables

- Domaine réalisable = région dans le plan.
- Contrainte : équation = droite



Exemple de Giapetto

$$\text{Max } z = 3x_1 + 2x_2$$

$$2x_1 + x_2 \leq 100$$

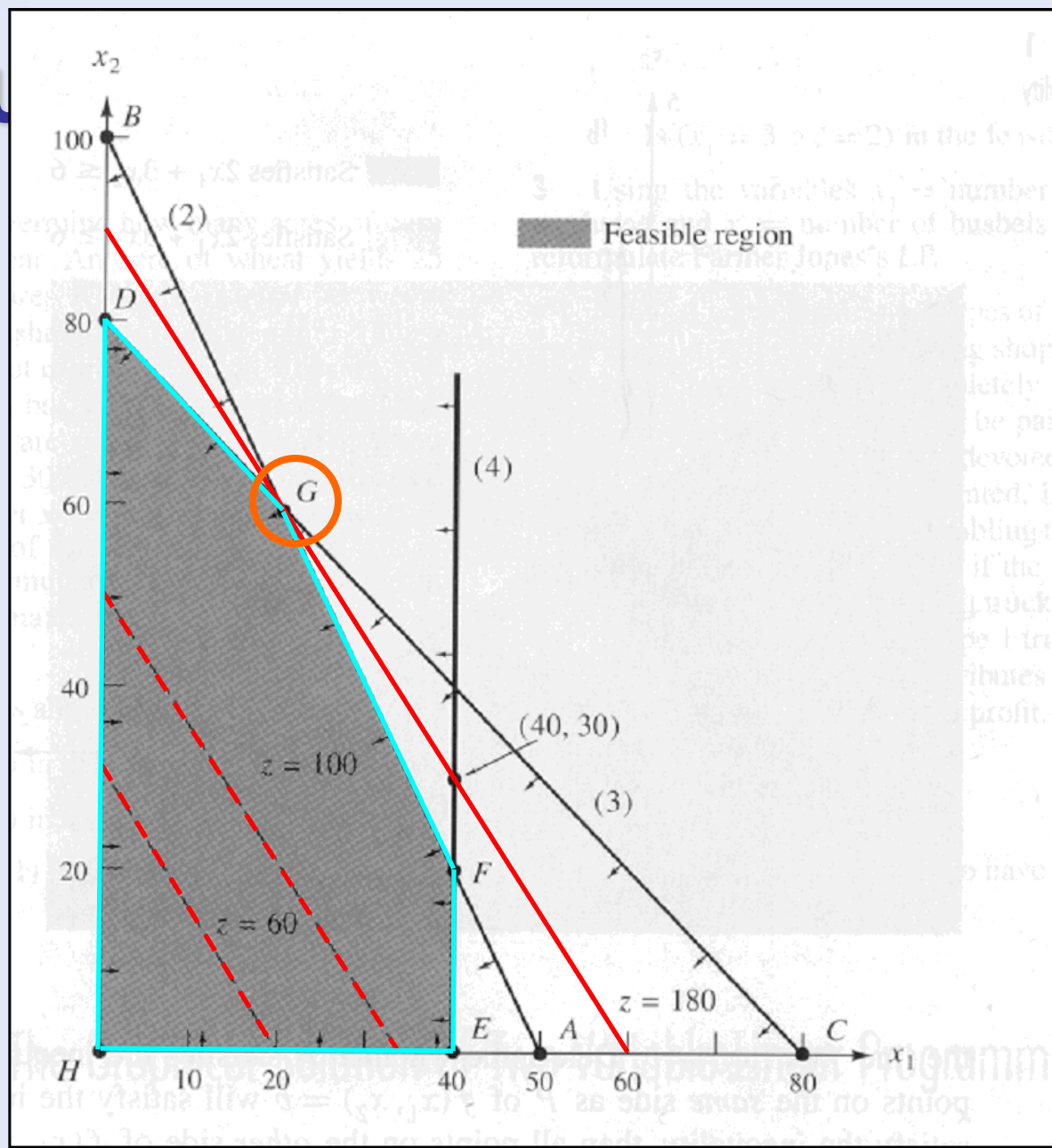
$$x_1 + x_2 \leq 80$$

$$x_1 \leq 40$$

$$x_1 \geq 0 \quad x_2 \geq 0$$

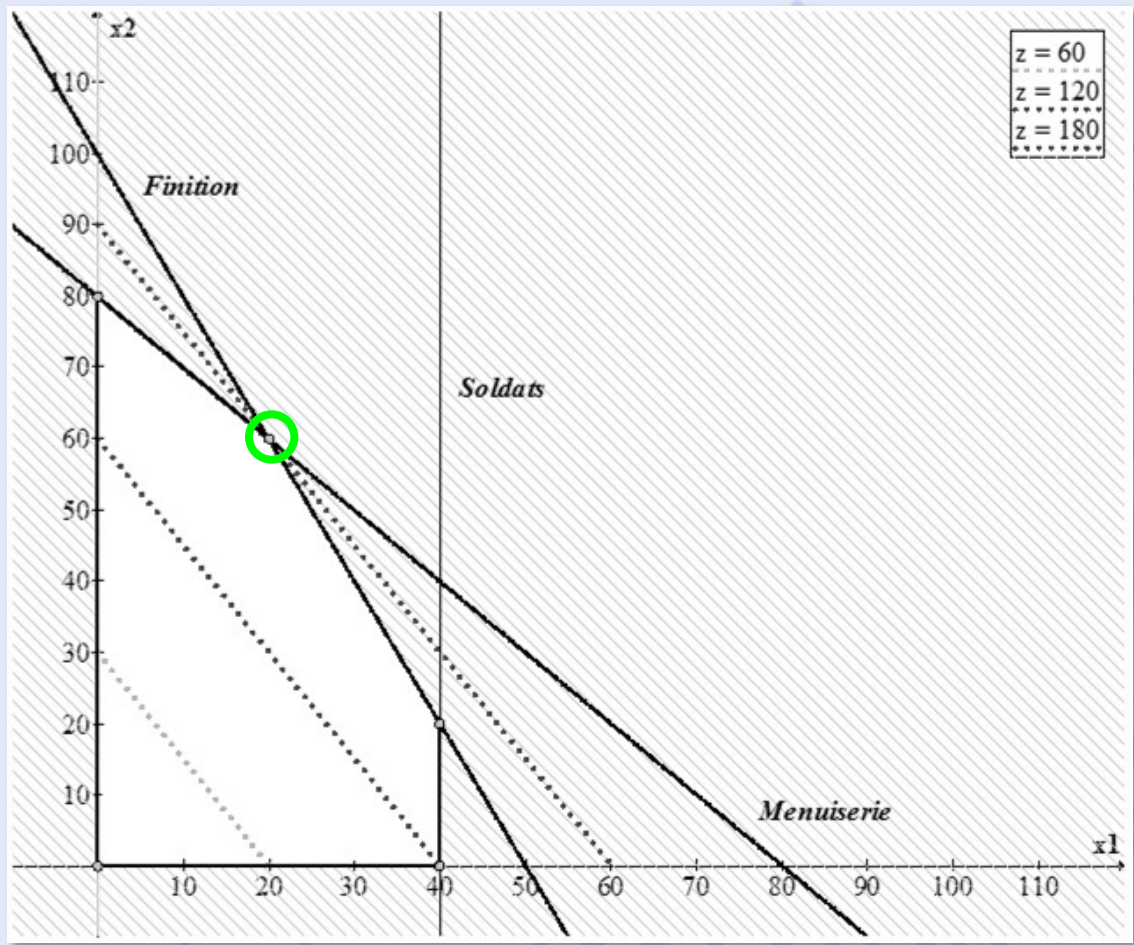
Solu

- Contraintes
- Courbes iso-profit
- Solution optimale



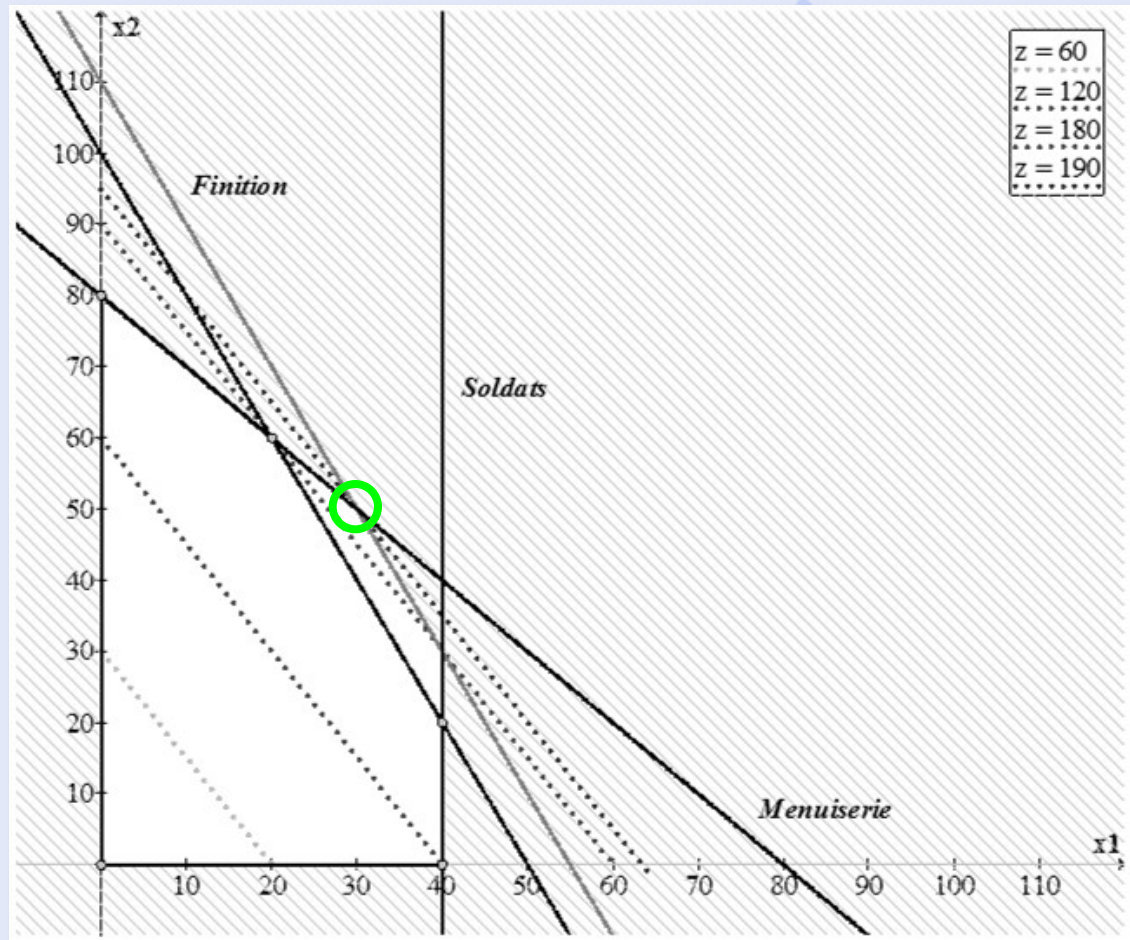
Contrainte active - Finition

100 heures



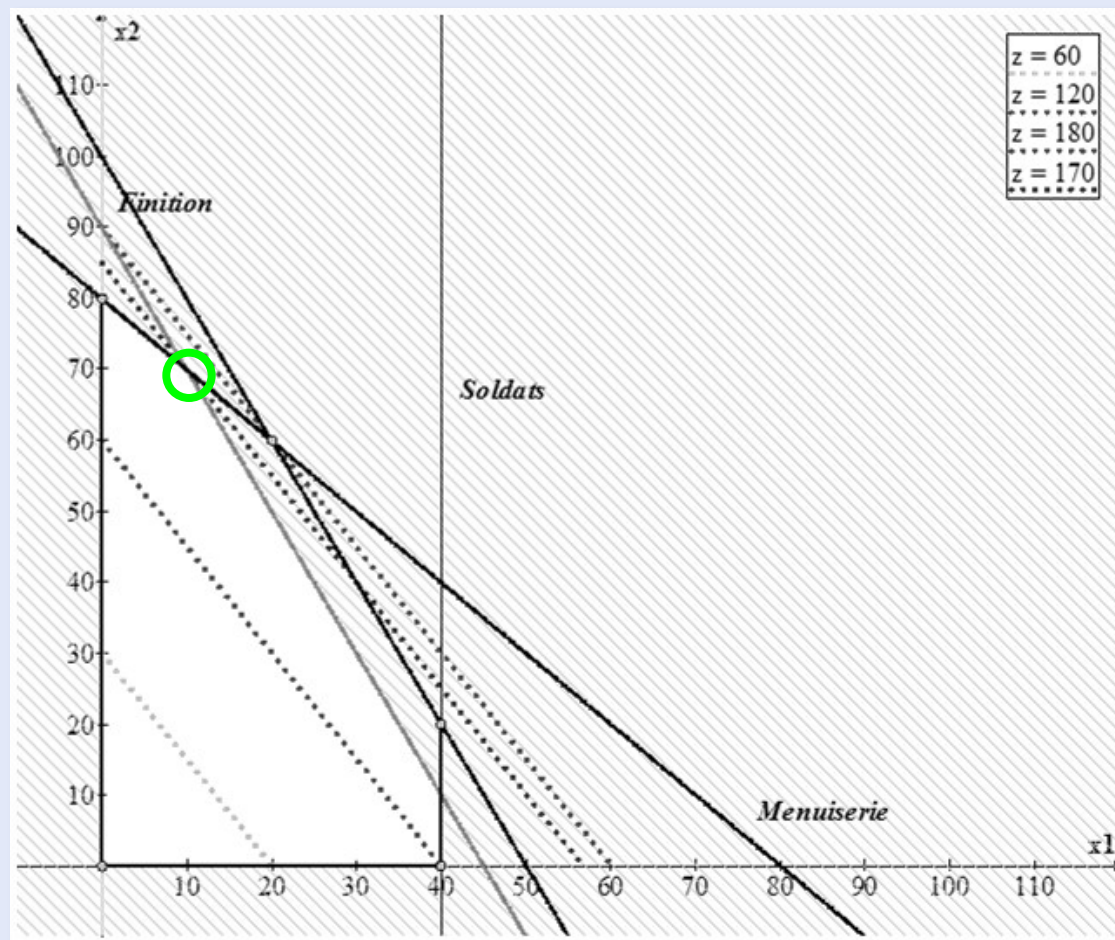
Contrainte active - Finition

110 heures



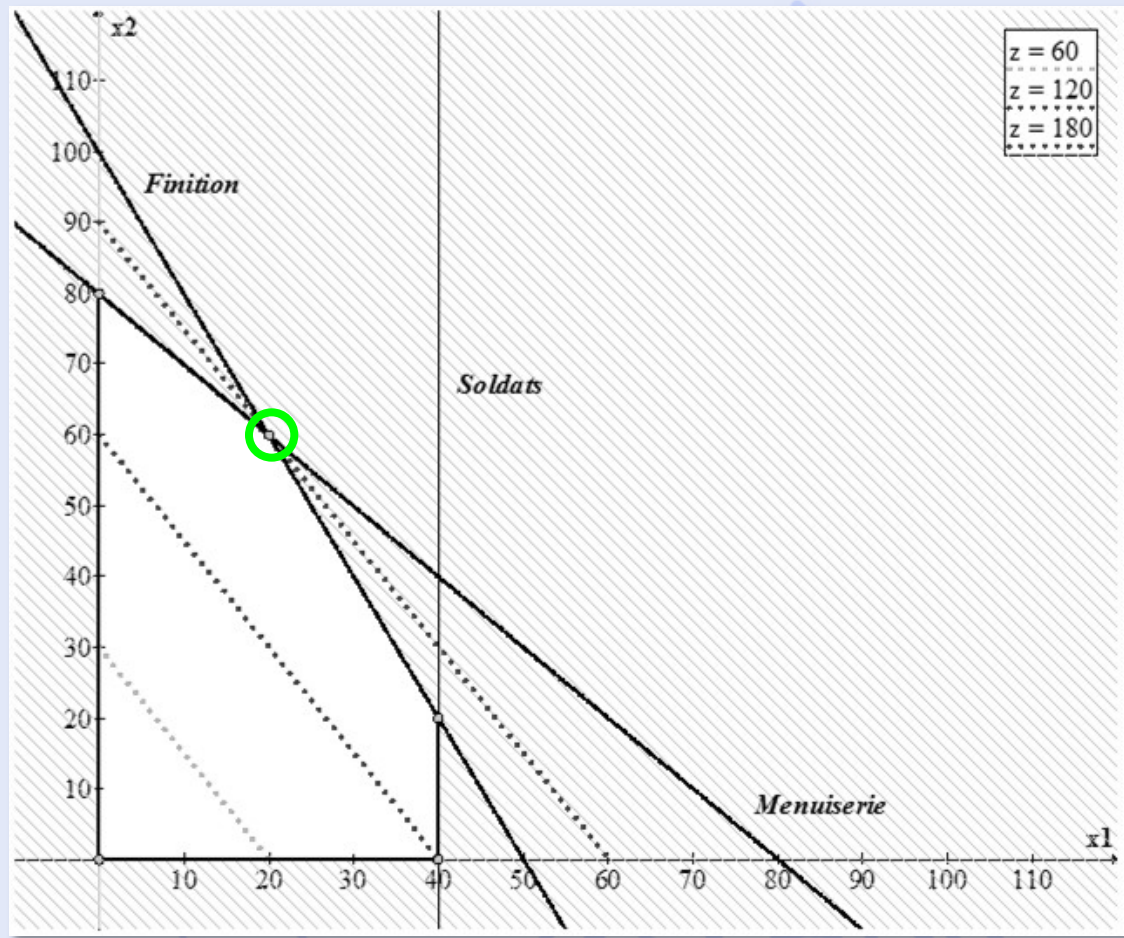
Contrainte active - Finition

90 heures



Contrainte non active - Soldats

40 soldats



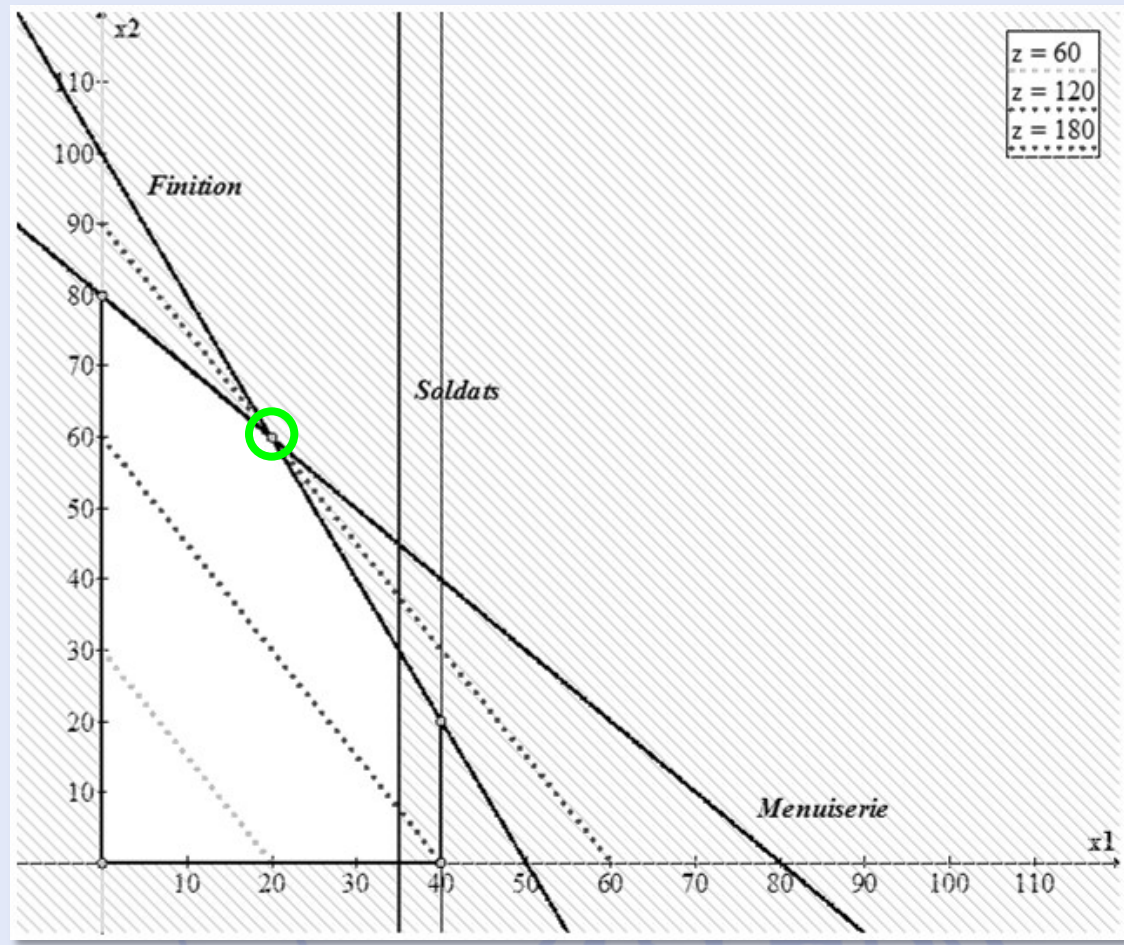
Contrainte non active - Soldats

45 soldats



Contrainte non active - Soldats

35 soldats



Solutions optimales multiples

Une entreprise du secteur automobile fabrique des voitures et des camions. Chaque véhicule doit être traité dans l'atelier de peinture et dans l'atelier de carrosserie. La capacité de l'atelier de peinture permet de traiter 40 camions par jour (si l'on ne peint que des camions), ou 60 voitures par jour (si l'on ne peint que des voitures). De la même façon la capacité de l'atelier de carrosserie est limitée à 50 camions par jour et à 50 voitures par jour. Chaque camion produit rapporte €300, et chaque voiture €200. Déterminer un plan de production quotidien qui permette de maximiser le profit de l'entreprise.

Solutions optimales multiples

x_1 = number of trucks produced daily
 x_2 = number of cars produced daily

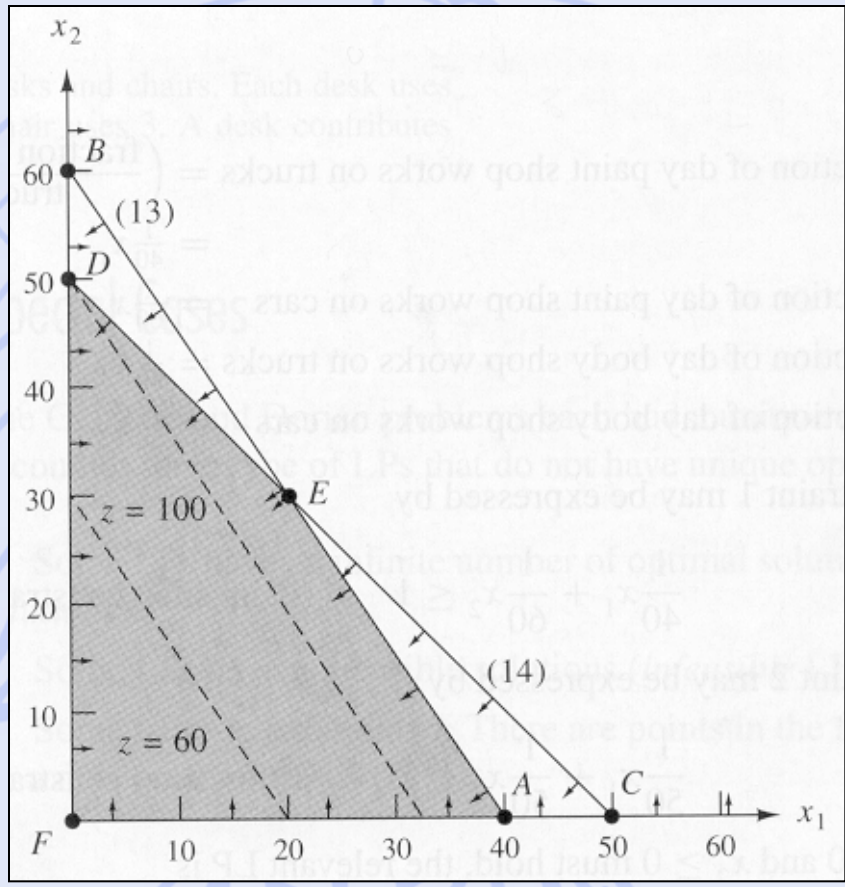
$$\max z = 300x_1 + 200x_2$$

s.t.

$$\frac{1}{40}x_1 + \frac{1}{60}x_2 \leq 1$$

$$\frac{1}{50}x_1 + \frac{1}{50}x_2 \leq 1$$

$$x_1 \geq 0 \quad x_2 \geq 0$$

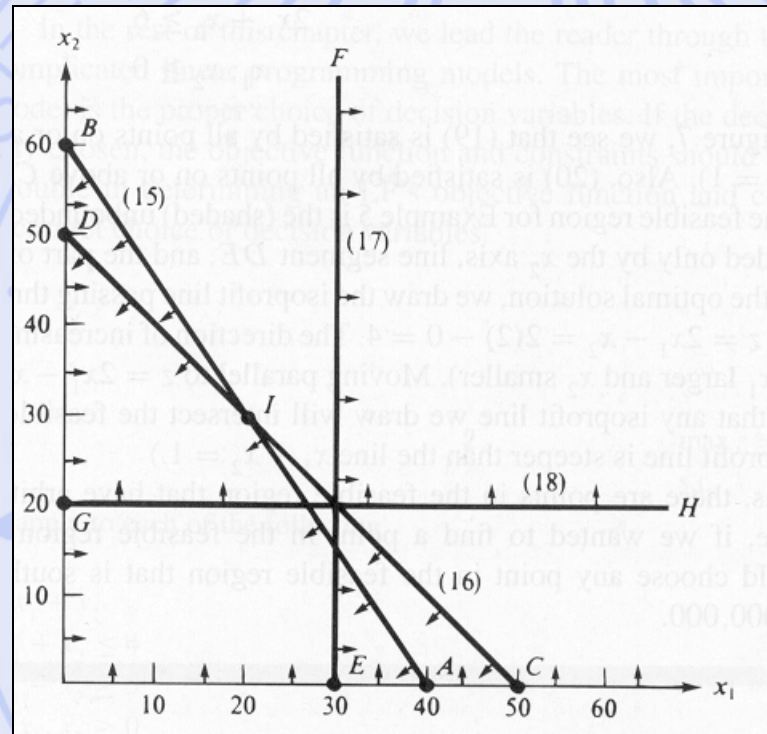


Exemple de PL contradictoire

Suppose now that the auto company is required to product at least 30 trucks and 20 cars per day.

→ 2 additional constraints:

$$x_1 \geq 30 \quad x_2 \geq 20$$



Exemple de PL non borné

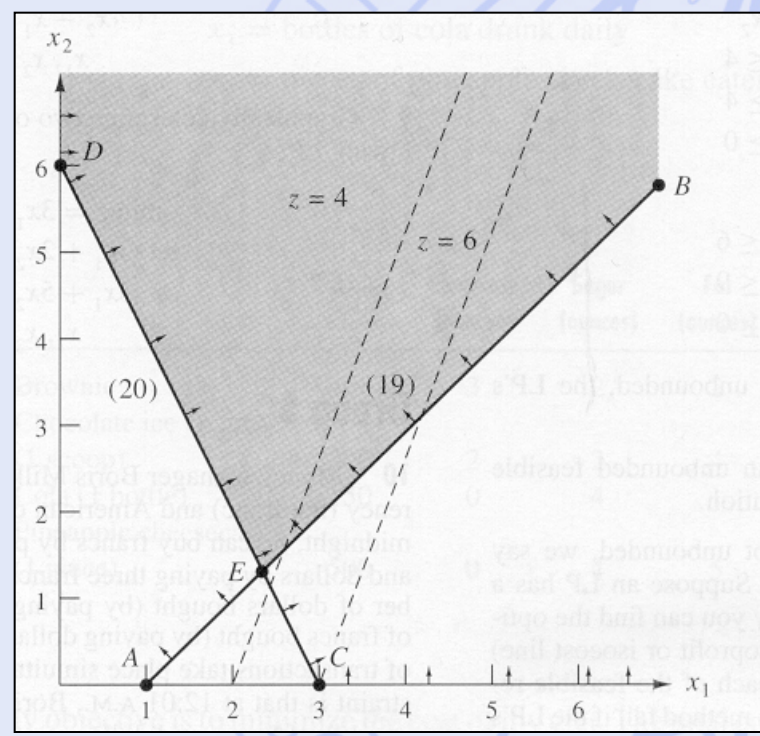
$$\max z = 2x_1 - x_2$$

s.t.

$$x_1 - x_2 \leq 1$$

$$2x_1 + x_2 \geq 6$$

$$x_1 \geq 0 \quad x_2 \geq 0$$



Solution optimale

- Peut être :
 - Unique → sommet du domaine réalisable,
 - Multiple → côté du domaine réalisable,
 - Infinie (contraintes manquantes ?),
 - Impossible (contraintes incompatibles !).
- Certaines contraintes sont actives (binding) :
LHS = RHS
- Certaines contraintes sont non-actives (nonbinding):
LHS \neq RHS (différence (écart) = slack)

Formulation PL

Minimiser
$$\sum_i \left\{ e_i A_i + \sum_{j,l} p_{il} X_{ijl} \right\} + \sum_j \left\{ f_j B_j + v_j \sum_{k,l} C_{jk} d_{kl} \right\}$$

$$+ \sum_{i,j,l} c_{ijl} X_{ijl} + \sum_{j,k,l} h_{jkl} C_{jk}$$

SOUS :

$$\sum_{j,l} X_{ijl} \leq A_i P_i \quad \forall i$$

$$\sum_{i,l} X_{ijl} \leq B_j W_j \quad \forall j$$

$$\sum_{i,j} X_{ijl} = \sum_{j,k} C_{jk} d_{kl} \quad \forall l$$

$$\sum_j C_{jk} = 1 \quad \forall k$$

$$X_{ijl} \geq 0, \quad A_i, B_j, C_{jk} = 0 \text{ or } 1.$$

→ 1427 variables et 77 contraintes.

PL typiques :

Régime alimentaire (diet)

Four foods are available for consumption: brownies, chocolate ice cream, cola and pineapple cheesecake. One brownie costs $\text{€}50$, one scoop of chocolate ice cream costs $\text{€}20$, one bottle of cola costs $\text{€}30$, and one piece of cheesecake costs $\text{€}80$. Each day, you must ingest at least 500 calories, 6 oz of chocolate, 10 oz of sugar and 8 oz of fat. Formulate a LP to satisfy these requirements at minimum cost.

Per unit:	Calories	Chocolate	Sugar	Fat
Brownie	400	3	2	2
Chocolate ice cream	200	2	2	4
Cola	150	0	4	1
Cheesecake	500	0	4	5

PL typiques : Régime alimentaire

x_1	number of brownies
x_2	number of scoops of chocolate ice cream
x_3	bottles of cola
x_4	pieces of pineapple cheesecake

• LP formulation:

$$\begin{aligned} \min z &= 50x_1 + 20x_2 + 30x_3 + 80x_4 \\ 400x_1 + 200x_2 + 150x_3 + 500x_4 &\geq 500 \\ 3x_1 + 2x_2 &\geq 6 \\ 2x_1 + 2x_2 + 4x_3 + 4x_4 &\geq 10 \\ 2x_1 + 4x_2 + x_3 + 5x_4 &\geq 8 \\ x_i &\geq 0 \quad (i=1,2,3,4) \end{aligned}$$

• LP solution:

$$x_1 = 0 \quad x_2 = 3 \quad x_3 = 1 \quad x_4 = 0 \quad z = 90$$

• Slacks:

$$t_1 = 250 \quad t_2 = 0 \quad t_3 = 0 \quad t_4 = 5$$

PL typiques :

Horaire de travail

- Un fast-food a besoin d'un nombre différent d'employés plein-temps chaque jour de la semaine :

Jour	Lu	Ma	Me	Je	Ve	Sa	Di
Nb.	17	13	15	19	14	16	11
Coût	100	100	100	100	100	150	200

- Chaque employé travaille 5 jours de suite et a ensuite 2 jours de congé.
- Il y a donc 7 horaires de travail possibles :
Lu-Ve, Ma-Sa, Me-Di, Je-Lu, Ve-Ma, Sa-Me, Di-Je

PL typiques : Horaire de travail

Horaires	Lu-Ve	Ma-Sa	Me-Di	Je-Lu	Ve-Ma	Sa-Me	Di-Je				
Salaire hebd.	500	550	650	650	650	650	600				
	x1	x2	x3	x4	x5	x6	x7				z
Nb. Tps-plein	6,33333333	5	0,33333333	7,33333333	0	3,33333333	0		Nb. Total		13066,6667
Contraintes								LHS	<=, >= or =	RHS	Ecart
Lundi	1			1	1	1	1	17	>=	17	0
Mardi	1	1			1	1	1	14,6666667	>=	13	1,66666667
Mercredi	1	1	1			1	1	15	>=	15	0
Jeudi	1	1	1	1			1	19	>=	19	0
Vendredi	1	1	1	1	1			19	>=	14	5
Samedi		1	1	1	1	1		16	>=	16	0
Dimanche			1	1	1	1	1	11	>=	11	0

- Solution optimale non entière !
- Arrondir ?
- Vers le haut.

PL typiques : Horaire de travail

Horaires	Lu-Ve	Ma-Sa	Me-Di	Je-Lu	Ve-Ma	Sa-Me	Di-Je				
Salaire hebd.	500	550	650	650	650	650	600				
	x1	x2	x3	x4	x5	x6	x7				z
Nb. Tps-plein	7	5	1	8	0	4	0		Nb. Total		14700
Contraintes								LHS	<=, >= or =	RHS	Ecart
Lundi	1			1	1	1	1	19	>=	17	2
Mardi	1	1			1	1	1	16	>=	13	3
Mercredi	1	1	1			1	1	17	>=	15	2
Jeudi	1	1	1	1			1	21	>=	19	2
Vendredi	1	1	1	1	1			21	>=	14	7
Samedi		1	1	1	1	1		18	>=	16	2
Dimanche			1	1	1	1	1	13	>=	11	2

- $z = 14700$ au lieu de $13066,66... !$
- Solution optimale entière ?
- Passage de LP à IP (Integer Programming).

PL typiques : Horaire de travail

Horaires	Lu-Ve	Ma-Sa	Me-Di	Je-Lu	Ve-Ma	Sa-Me	Di-Je				
Salaire hebd.	500	550	650	650	650	650	600				
	x1	x2	x3	x4	x5	x6	x7				z
Nb. Tps-plein	6	6	0	8	0	2	1		Nb. Total		13400
Contraintes								LHS	<=, >= or =	RHS	Ecart
Lundi	1			1	1	1	1	17	>=	17	0
Mardi	1	1			1	1	1	15	>=	13	2
Mercredi	1	1	1			1	1	15	>=	15	0
Jeudi	1	1	1	1			1	21	>=	19	2
Vendredi	1	1	1	1	1			20	>=	14	6
Samedi		1	1	1	1	1		16	>=	16	0
Dimanche			1	1	1	1	1	11	>=	11	0

- $z = 13400$ au lieu de 14700 !
- Supérieur à $13066,66\dots$

Multiperiod financial models

Finco Investment Corp. must determine investment strategy for the firm during the next three years. At present time (time 0), \$100,000 is available for investment. Investments A, B, C, D and E are available. The cash flow associated with investing \$1 in each investment are:

	0	1	2	3
A	-\$1	+\$0.50	+\$1	\$0
B	\$0	-\$1	+\$0.50	+\$1
C	-\$1	+\$1.2	\$0	\$0
D	-\$1	\$0	\$0	+\$1.9
E	\$0	\$0	-\$1	+\$1.5

To ensure that the company's portfolio is diversified, Finco requires that at most \$75,000 be placed in a single investment. In addition to investments A-E, Finco can earn interest at 8% per year by keeping uninvested cash in money market funds. Returns from investments may be immediately reinvested. Finco cannot borrow funds. Formulate an LP to maximize cash on hand at time 3.

Multiperiod financial models

A = dollars invested in A

B = dollars invested in B

C = dollars invested in C

D = dollars invested in D

E = dollars invested in E

S_t = dollars invested in MM funds at time t

- LP formulation:

$$\max z = B + 1.9D + 1.5E + 1.08S_2$$

$$A + C + D + S_0 = 100,000$$

$$0.5A + 1.2C + 1.08S_0 = B + S_1$$

$$A + 0.5B + 1.08S_1 = E + S_2$$

$$A \leq 75,000 \quad B \leq 75,000 \quad C \leq 75,000 \quad D \leq 75,000 \quad E \leq 75,000$$

$$A, B, C, D, E, S_1, S_2 \geq 0$$

Définitions

- Programme linéaire sous forme canonique :

$$\text{Min } z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (1)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & \leq & b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n & \leq & b_2 \\ & \vdots & \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n & \leq & b_i \\ & \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n & \leq & b_m \end{cases} \quad (2)$$

$$x_j \geq 0 \quad j = 1, 2, \dots, n \quad (3)$$

$$\text{Min } z = \sum_{j=1}^n c_j x_j$$

$$\begin{cases} \sum_{j=1}^n a_{ij}x_j \leq b_i, & i = 1, 2, \dots, m \\ x_j \geq 0, & j = 1, 2, \dots, n \end{cases}$$

où $c_1, c_2, \dots, c_n, a_{11}, a_{12}, \dots, a_{mn}, b_1, b_2, \dots, b_m \in \mathbb{R}$

Définitions

- Fonction économique à max ou min :

$$\text{Min } z = - \text{Max } (- z)$$

- Contraintes : \leq ou \geq ou $=$

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i$$



$$-a_{i1}x_1 - a_{i2}x_2 - \dots - a_{in}x_n \leq -b_i$$

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i$$



$$\left\{ \begin{array}{l} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \end{array} \right.$$

$$\left\{ \begin{array}{l} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \end{array} \right.$$

Notation matricielle

$$\text{Min } \{ CX \mid AX \leq b, X \geq 0 \}$$

$$C' = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Variables d'écart

- Pour passer d'inéquations à des équations :

$$\begin{array}{c}
 a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \\
 \hline
 \updownarrow \\
 a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + t_i = b_i \\
 \text{avec } t_i \geq 0
 \end{array}$$

$$\begin{array}{c}
 a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \\
 \hline
 \updownarrow \\
 a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - t_i = b_i \\
 \text{avec } t_i \geq 0
 \end{array}$$

- Permet de ramener toutes les contraintes à des équations, au prix de l'ajout des variables d'écart → 2^{ème} forme canonique pour un PL.

Exemple 2

$$\text{Max } z = 40x_1 + 30x_2$$

$$x_1 \leq 8$$

$$x_2 \leq 6$$

$$x_1 + 2x_2 \leq 15$$

$$2x_1 + x_2 \leq 18$$

$$x_1 \geq 0 \quad x_2 \geq 0$$

⇕

$$\text{Max } z = 40x_1 + 30x_2$$

$$x_1 + t_1 = 8$$

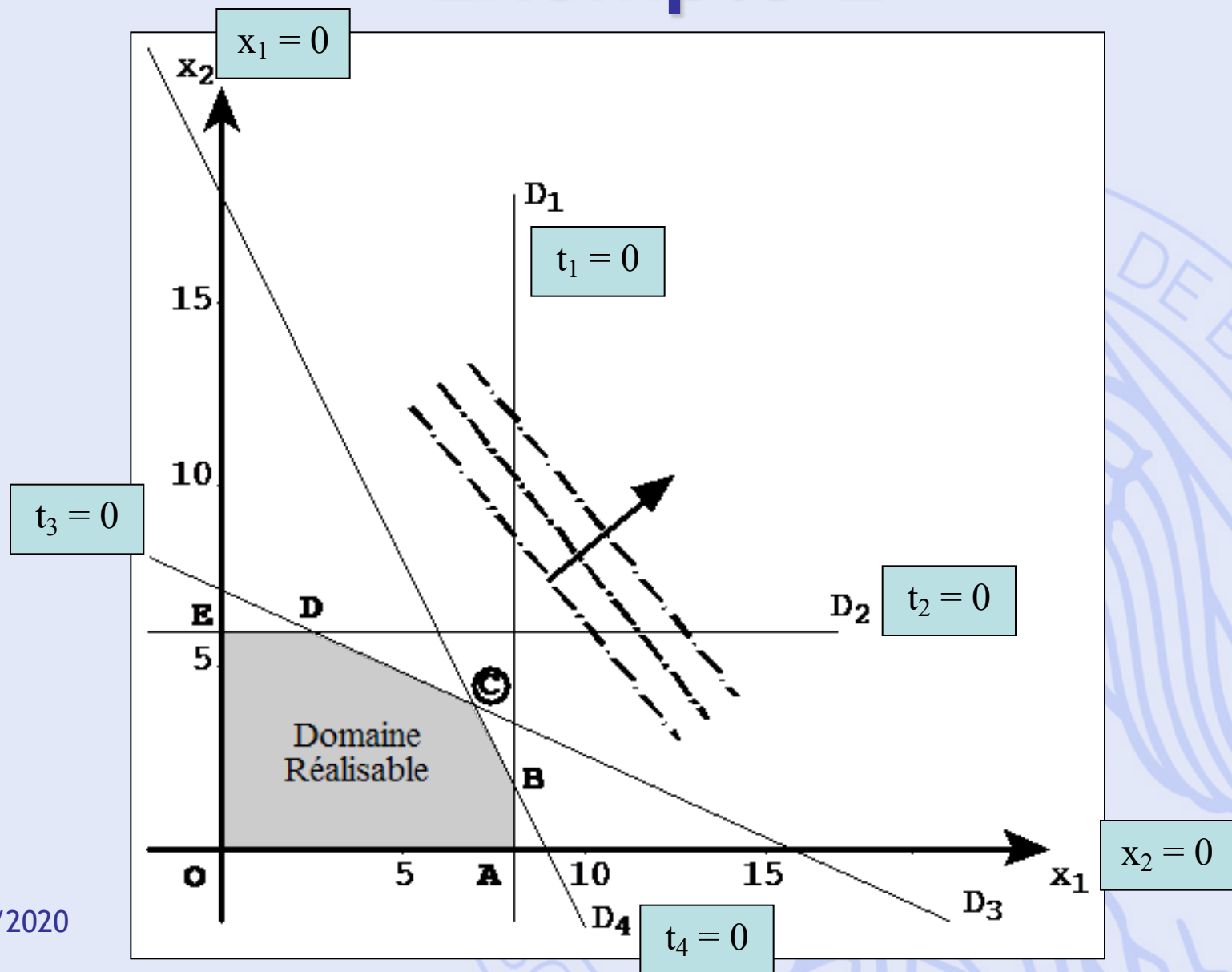
$$x_2 + t_2 = 6$$

$$x_1 + 2x_2 + t_3 = 15$$

$$2x_1 + x_2 + t_4 = 18$$

$$x_1 \geq 0 \quad x_2 \geq 0 \quad t_1 \geq 0 \quad t_2 \geq 0 \quad t_3 \geq 0 \quad t_4 \geq 0$$

Exemple 2



Exemple 2 - Remarques

- Le domaine réalisable est un polygone convexe.
- Contrainte = côté du polygone = une variable égale à 0.
- Sommet = intersection de deux côtés = deux variables égales à 0.
- $n = 6$, $m = 4$

Base

- Sous-matrice non singulière B , $m \times m$, de A .
- Colonnes de B : vecteurs de base :
 - Soit $B = \{ P_{j_1}, P_{j_2}, \dots, P_{j_m} \}$.
 - $I(B) = \{ j_1, j_2, \dots, j_m \} =$ ensemble des indices de base.
 - $x_{j_1}, x_{j_2}, \dots, x_{j_m}$: **variables de base.**
 - $J(B) =$ ensemble des indices hors base \rightarrow **variables hors base.**

Solution de base

- Solution obtenue à partir d'une base B , en annulant les $n-m$ variables hors base et en résolvant le système de Cramer associé aux m variables de base :

$$AX = b \Rightarrow BX_B = b \Rightarrow X_B = B^{-1} b$$

- Cas particuliers :
 - Solution de base réalisable (s.b.r.) : $X_B \geq 0$
 - Solution de base explicitée : si $B = I$,
 $\Rightarrow X_B = b$

Exemple 2 : bases

$$\begin{aligned} \text{Max } z &= 40x_1 + 30x_2 \\ x_1 &\leq 8 \\ x_2 &\leq 6 \\ x_1 + 2x_2 &\leq 15 \\ 2x_1 + x_2 &\leq 18 \\ x_1 \geq 0 \quad x_2 &\geq 0 \end{aligned}$$

⇕

$$\begin{aligned} \text{Max } z &= 40x_1 + 30x_2 \\ x_1 + x_2 + t_1 &= 8 \\ x_2 + t_2 &= 6 \\ x_1 + 2x_2 + t_3 &= 15 \\ 2x_1 + x_2 + t_4 &= 18 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad t_1 \geq 0 \quad t_2 \geq 0 \quad t_3 \geq 0 \quad t_4 \geq 0 \end{aligned}$$

Exemple 2 - Remarques

- Le domaine réalisable est un polygone convexe.
- Contrainte = côté du polygone = une variable égale à 0.
- Sommet = intersection de deux côtés = deux variables égales à 0.
- $n = 6, m = 4$
- Solution de base : $n - m = 2$ variables hors base (égales à 0) !

s.b.r. de l'exemple 2

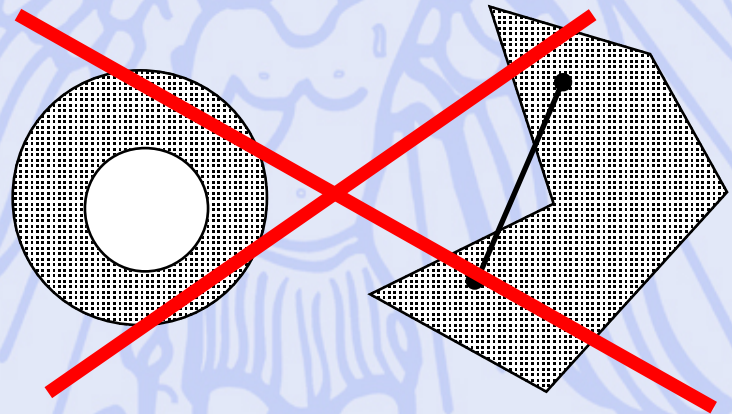
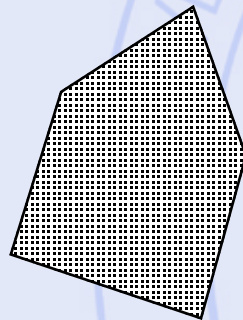
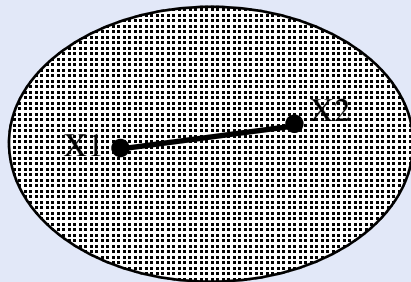
Enumération des Solutions de Base								
Sommet	x_1	x_2	t_1	t_2	t_3	t_4	z	Solution
O	0	0	8	6	15	18	0	s.b.r.
	0		0				-	-
E	0	6	8	0	3	12	180	s.b.r.
	0	7,5	8	-1,5	0	10,5	-	s.b.
	0	18	8	-12	-21	0	-	s.b.
A	8	0	0	6	7	2	320	s.b.r.
		0		0			-	-
	15	0	-7	6	0	-12	-	s.b.
	9	0	-1	6	6	0	-	s.b.
	8	6	0	0	-5	-4	-	s.b.
	8	3,5	0	2,5	0	-1,5	-	s.b.
B	8	2	0	4	3	0	380	s.b.r.
D	3	6	5	0	0	6	300	s.b.r.
	6	6	2	0	-3	0	-	s.b.
<u>C</u>	7	4	1	2	0	0	<u>400</u>	<u>s.b.r.≡s.o.</u>

Résultats fondamentaux

- Ensembles convexes :

$E \subset \mathbb{R}^n$ est dit convexe si pour tout X_1 ,
 $X_2 \in E$ on a :

$X = \alpha X_1 + (1-\alpha) X_2 \in E$, pour tout $0 \leq \alpha \leq 1$

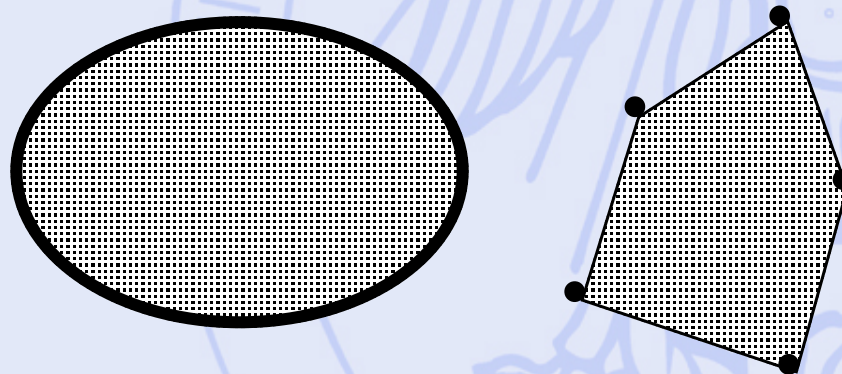


(toute combinaison linéaire convexe de 2 points de E est encore un point de E)

Ensembles convexes

- Sommet d'un ensemble convexe :
 $X \in E$ est appelé point extrême ou sommet de E si :

$$X = \alpha X_1 + (1-\alpha) X_2 \text{ avec } X_1, X_2 \in E \text{ et } 0 < \alpha < 1 \\ \Rightarrow X = X_1 = X_2$$



Ensembles convexes

- Polyèdre convexe :
Ensemble convexe borné comportant un nombre fini de sommets.
- Propriété :
Tout point d'un polyèdre convexe peut s'exprimer comme combinaison linéaire convexe des sommets du polyèdre.

Théorème 1

- L'ensemble $P = \{ X \mid AX \leq b, X \geq 0 \}$ est convexe.
- Démonstration :

Soient $X_1, X_2 \in P$ et $0 \leq \alpha \leq 1$:

Il faut montrer que $X = \alpha X_1 + (1-\alpha) X_2 \in P$

Or :

$$\textcircled{1} \quad X_1 \geq 0, X_2 \geq 0 \text{ et } 0 \leq \alpha \leq 1 \Rightarrow \alpha X_1 + (1-\alpha) X_2 \geq 0$$

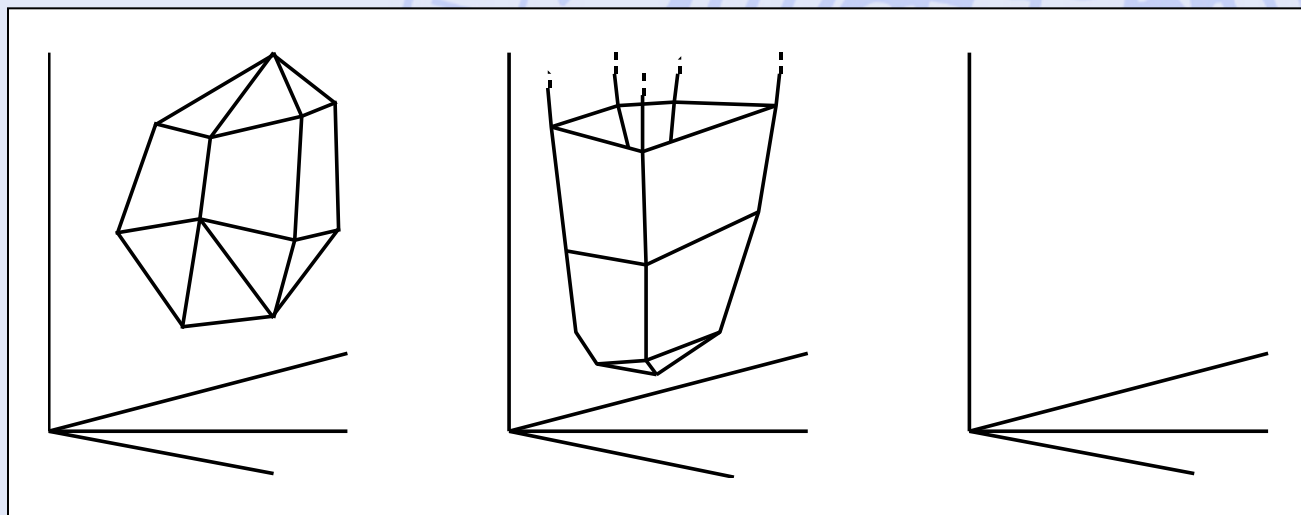
$$\textcircled{2} \quad AX_1 \leq b \text{ et } AX_2 \leq b \Rightarrow A(\alpha X_1 + (1-\alpha) X_2) \leq b$$

Théorème 2

- **Théorème de Weyl :**

L'ensemble $P = \{ X \mid AX \leq b, X \geq 0 \}$ est :

- soit vide,
- soit un polyèdre convexe,
- soit un ensemble polyédrique convexe non borné.



Théorème 3

- Dans le cas où $P = \{ X \mid AX \leq b, X \geq 0 \}$ est un polyèdre convexe, l'ensemble des s.o. du PL contient au moins un sommet de P .
- Démonstration :

Soit X une s.r. du PL $\text{opt } \{ CX \mid AX \leq b, X \geq 0 \}$:

$$X = \alpha_1 S_1 + \alpha_2 S_2 + \dots + \alpha_k S_k$$

où S_1, S_2, \dots, S_k sont les sommets de P , et :

$$\alpha_i \geq 0 \ (i=1,2,\dots,k) \text{ avec } \sum_{i=1}^k \alpha_i = 1$$

Théorème 3 (suite)

$$CX = \alpha_1 CS_1 + \alpha_2 CS_2 + \dots + \alpha_k CS_k$$

Soit :

$$CS_m = \min \{ CS_j ; j=1,2,\dots,k \}$$

$$CS_M = \max \{ CS_j ; j=1,2,\dots,k \}$$

Il vient :

$$CS_m \leq CX \leq CS_M$$

Conclusion :

Pour un PL à maximum : S_M est une s.o.

Pour un PL à minimum : S_m est une s.o.

Théorème 4

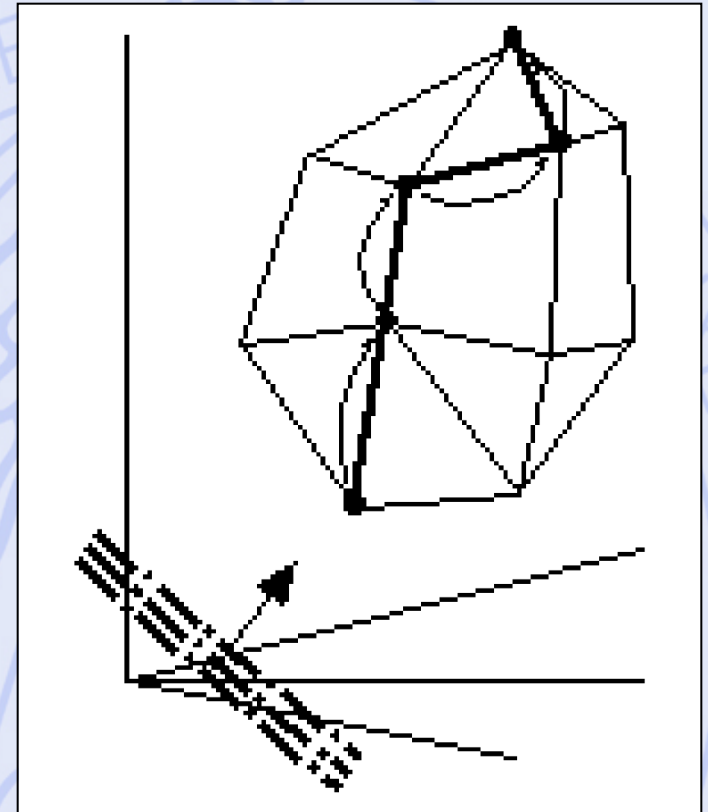
- Soit le PL $\min \{ CX \mid AX = b, X \geq 0 \}$.
- Si A est de rang m , tout sommet de l'ensemble $P = \{ X \mid AX = b, X \geq 0 \}$ correspond à une s.b.r. du PL, et réciproquement.
- Conclusion : On peut se limiter à l'étude des sommets du polyèdre P , c'est-à-dire à celle des s.b.r.

Dénombrement des bases

- Principe :
Calculer l'ensemble des s.b.r. du P.L. et comparer les valeurs prises par la fonction économique.
- Inconvénients :
 - Grand nombre de s.b. à calculer (cf. nombre de combinaisons de n variables m à m).
 - Ne permet pas de détecter une s.o. infinie.

Algorithme du simplexe

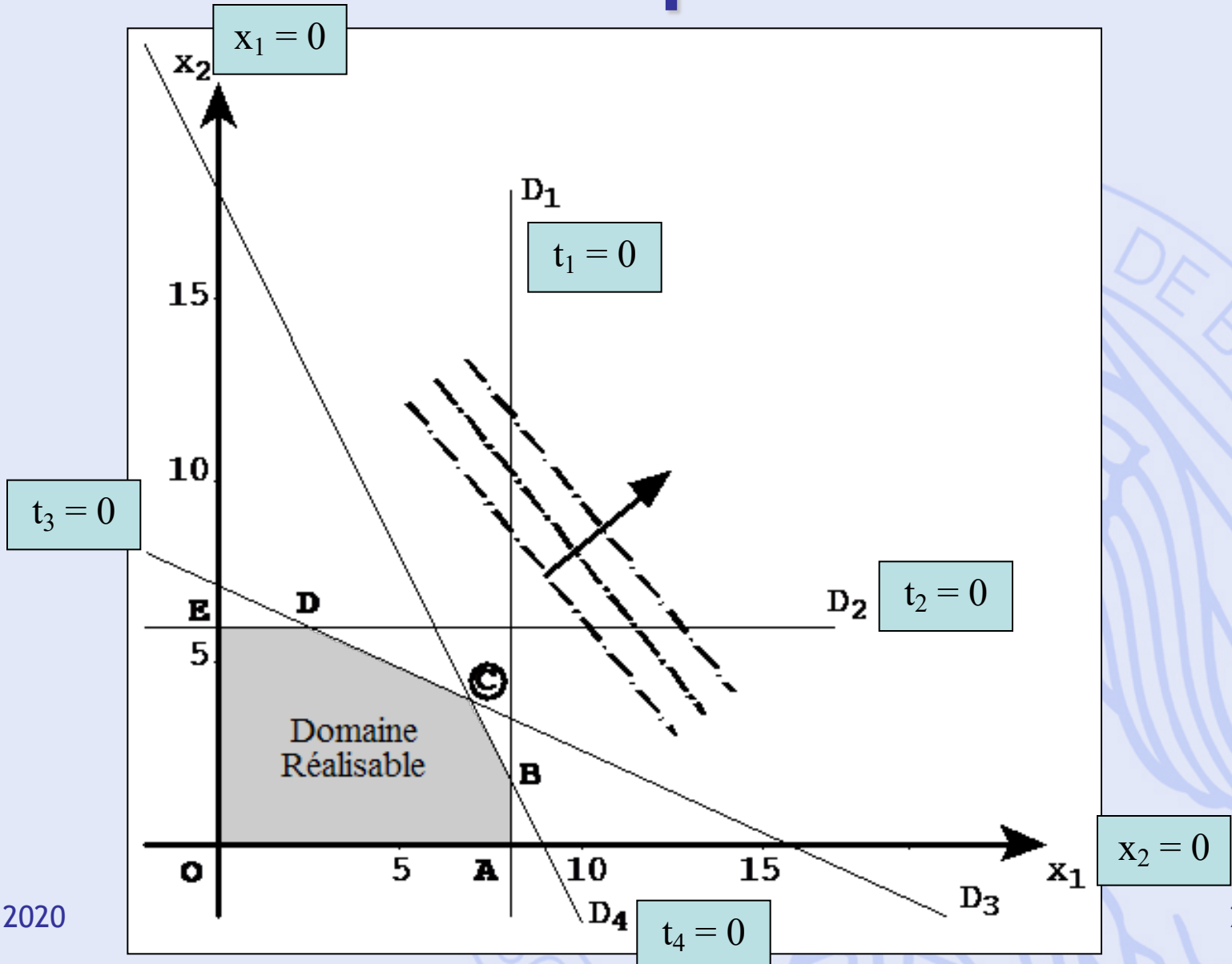
- Principe :
Cheminer de sommet (s.b.r.) en sommet adjacent en améliorant à chaque étape la fonction économique z , jusqu'à atteindre une solution optimale ou à montrer que la solution optimale est infinie.



Algorithme du simplexe

- Passer d'un sommet à un sommet adjacent revient à permuter une variable de base et une variable hors base :
 - faire entrer en base une variable hors base,
 - faire sortir de base une variable de base.
- Conditions d'application :
 - Contraintes sous formes d'équations (\Rightarrow variables d'écart éventuellement).
 - 1^{ère} solution de base réalisable explicitée (sinon méthode M ou méthode en deux phases) (\Rightarrow càd A contient une sous-matrice unité et $b \geq 0$).

Exemple 2



Mise en œuvre

- Formules de changement de base.
- Variation de z lors d'un changement de base.
- Critères de choix d'un changement de base.
- Critère d'optimalité.
- Détection d'une solution optimale infinie.
- Méthode de la base artificielle.

Formules de changement de base

- $\min \{ CX \mid AX = b, X \geq 0 \}$.
- s.b.r. explicitée (base B) :
 - A contient une sous-matrice unité ($m \times m$) et $b \geq 0$.
 - soit $I(B) = \{ 1, 2, \dots, m \}$: ensemble des indices de base.
 - soit $J(B) = \{ m+1, \dots, n \}$: ensemble des indices hors base.
- s.b.r. correspondante :

$$\begin{cases} x_i = b_i & \forall i \in I(B) \\ x_j = 0 & \forall j \in J(B) \end{cases}$$

$$z_0 = \sum_{i \in I(B)} c_i b_i$$

Changement de base

- Passer à une nouvelle s.b.r. explicitée (base B'):
 - Càd. réécrire le système des contraintes sous une forme équivalente $A'X = b'$ où $b' \geq 0$ et A' contient une sous-matrice unité B' .

- Nouvelle s.b.r. :

$$\begin{cases} x_i = b_i' & \forall i \in I(B') \\ x_j = 0 & \forall j \in J(B') \end{cases}$$

$$z_0' = \sum_{i \in I(B')} c_i b_i'$$

- x_r ($r \in I(B)$) est remplacée par x_k ($k \in J(B)$)
 → nouvelle base B' : $I(B') = (I(B) \cup \{k\}) \setminus \{r\}$
 à expliciter.

Changement de base

- P_1, P_2, \dots, P_m : base de l'espace des contraintes :

$$P_j = \sum_{i \in I(B)} a_{ij} P_i, \quad j=1,2,\dots,n$$

$$P_0 = \sum_{i \in I(B)} b_i P_i$$

- Dans la nouvelle base, si $a_{rk} \neq 0$:

$$\begin{aligned} P_k &= \sum_{i \in I(B)} a_{ik} P_i \\ &= \sum_{\substack{i \in I(B) \\ i \neq r}} a_{ik} P_i + a_{rk} P_r \end{aligned}$$

$$P_r = \frac{1}{a_{rk}} P_k - \sum_{\substack{i \in I(B) \\ i \neq r}} \frac{a_{ik}}{a_{rk}} P_i$$

Changement de base

- Donc :

$$\begin{aligned} P_j &= \sum_{\substack{i \in I(B) \\ i \neq r}} a_{ij} P_i + a_{rj} P_r \\ &= \sum_{\substack{i \in I(B) \\ i \neq r}} \left(a_{ij} - \frac{a_{rj} a_{ik}}{a_{rk}} \right) P_i + \frac{a_{rj}}{a_{rk}} P_k \end{aligned}$$

$$\begin{aligned} P_0 &= \sum_{\substack{i \in I(B) \\ i \neq r}} b_i P_i + b_r P_r \\ &= \sum_{\substack{i \in I(B) \\ i \neq r}} \left(b_i - \frac{b_r a_{ik}}{a_{rk}} \right) P_i + \frac{b_r}{a_{rk}} P_k \end{aligned}$$

Changement de base

• Par ailleurs :
$$P_j = \sum_{i \in I(B')} a'_{ij} P_i \quad P_0 = \sum_{i \in I(B')} b'_i P_i$$

→ Formules de changement de base :

$$\left\{ \begin{array}{l} b'_i = b_i - \frac{b_r a_{ik}}{a_{rk}}, \quad \forall i \in I(B'), i \neq k \\ b'_k = \frac{b_r}{a_{rk}} \end{array} \right.$$

$$\left\{ \begin{array}{l} a'_{ij} = a_{ij} - \frac{a_{rj} a_{ik}}{a_{rk}}, \quad \forall i \in I(B'), i \neq k \\ a'_{kj} = \frac{a_{rj}}{a_{rk}} \end{array} \right.$$

Changement de base

- Lorsque B est explicitée :
 - a_{ij} : coeff. de x_j dans la contrainte où la variable en base est x_i
 - b_i : second membre de la contrainte (valeur de x_i)
- Lorsque B' est explicitée :
 - a'_{ij} : coeff. de x_j dans la contrainte où la variable en base est x_i
 - b'_i : second membre de la contrainte (valeur de x_i)
- a_{rk} est le pivot du changement de base
→ doit être positif pour que B' soit réalisable !

Variation de z

- Soit :

$$z_j = \sum_{i \in I(B)} c_i a_{ij} \qquad z_0 = \sum_{i \in I(B)} c_i b_i$$

- En B' :

$$\begin{aligned} z'_0 &= \sum_{i \in I(B')} c_i b'_i \\ &= \sum_{i \in I(B)} c_i b'_i - c_r b'_r + c_k b'_k \\ &= \sum_{i \in I(B)} c_i \left(b_i - \frac{b_r a_{ik}}{a_{rk}} \right) + c_k \frac{b_r}{a_{rk}} \\ &= z_0 - \frac{b_r}{a_{rk}} (z_k - c_k) \end{aligned}$$

Variation de z

- Pour un PL à maximum : intérêt à avoir $z_k - c_k < 0$
- Pour un PL à minimum : intérêt à avoir $z_k - c_k > 0$
- Pour $i \in I(B)$: $z_i - c_i = 0$
- Calcul des $z_j - c_j$ dans la nouvelle base :

$$\begin{aligned}
 z'_j - c_j &= \sum_{i \in I(B')} c_i a'_{ij} - c_j \\
 &= \sum_{i \in I(B)} c_i \left(a_{ij} - \frac{a_{rj} a_{ik}}{a_{rk}} \right) + c_k \frac{a_{rj}}{a_{rk}} - c_r a'_{rj} - c_j \\
 \hline
 z'_j - c_j &= (z_j - c_j) - \frac{a_{rj}}{a_{rk}} (z_k - c_k)
 \end{aligned}$$

Variation de z

- Expression de z en une s.r. quelconque Q en fonction de z_0 :

$$\begin{aligned} z(Q) &= \sum_{j=1}^n c_j x_j \\ &= \sum_{i \in I(B)} c_i x_i + \sum_{j \in J(B)} c_j x_j \end{aligned}$$

- Or : $x_i + \sum_{j \in J(B)} a_{ij} x_j = b_i, \quad \forall i \in I(B)$

- D'où :
$$\begin{aligned} z(Q) &= \sum_{i \in I(B)} c_i b_i - \sum_{j \in J(B)} (z_j - c_j) x_j \\ &= z_0 - \sum_{j \in J(B)} (z_j - c_j) x_j \end{aligned}$$

Critère d'optimalité

- Pour la s.b.r. correspondant à la base B :
 - Pour un PL à max : $z_j - c_j \geq 0 \quad \forall j \in J(B)$
 - Pour un PL à min : $z_j - c_j \leq 0 \quad \forall j \in J(B)$
- Démonstration :
Résulte directement de la propriété précédente :
 - Pour un PL à max : $z(Q) \leq z_0 \quad \forall \text{ s.r. } Q$
 - Pour un PL à min : $z(Q) \geq z_0 \quad \forall \text{ s.r. } Q$

Solution optimale infinie

- Pour un PL à minimum, si $\exists k \in J(B)$ tel que :

$$z_k - c_k > 0 \quad \text{et} \quad a_{ik} \leq 0, \quad \forall i$$

Alors la s.o. est infinie (cas non borné).

- Démonstration :

En prenant $x_k > 0$ ($x_k = 0$ en B), on obtient une s.r. Q avec :

$$x_i = b_i - a_{ik} x_k \geq 0 \quad \forall i \in I(B)$$

et

$$z(Q) = z_0 - (z_k - c_k) x_k < z_0$$

Choix du changement de base

- Pour un PL à minimum :
 - Si B ne correspond pas à une s.o. et si la s.o. n'est pas infinie :

$$\exists j \in J(B) : z_j - c_j > 0 \text{ et } \exists i : a_{ij} > 0$$

- Choix de la variable qui entre en base :

$$x_k \text{ avec } z_k - c_k = \max_{z_j - c_j > 0} (z_j - c_j)$$

- Choix de la variable qui sort de base :

$$x_r \text{ avec } a_{rk} > 0 \text{ et } \frac{b_r}{a_{rk}} = \min_{a_{ik} > 0} \frac{b_i}{a_{ik}}$$

Choix du changement de base

- Nouvelle s.b.r. (meilleure) :

$$\begin{cases} b'_i = b_i - \frac{b_r a_{ik}}{a_{rk}} \geq 0, & \forall i \neq k \\ b'_k = \frac{b_r}{a_{rk}} \geq 0 \end{cases}$$

- Pour un PL à maximum :

$$x_k \text{ avec } z_k \ominus c_k = \min_{z_j - c_j < 0} (z_j - c_j)$$

Exemple 2

$$\begin{aligned} \text{Max } z &= 40x_1 + 30x_2 \\ x_1 &\leq 8 \\ x_2 &\leq 6 \\ x_1 + 2x_2 &\leq 15 \\ 2x_1 + x_2 &\leq 18 \\ x_1 \geq 0 \quad x_2 &\geq 0 \end{aligned}$$

⇕

$$\begin{aligned} \text{Max } z &= 40x_1 + 30x_2 \\ x_1 + t_1 &= 8 \\ x_2 + t_2 &= 6 \\ x_1 + 2x_2 + t_3 &= 15 \\ 2x_1 + x_2 + t_4 &= 18 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad t_1 \geq 0 \quad t_2 \geq 0 \quad t_3 \geq 0 \quad t_4 \geq 0 \end{aligned}$$

Tableau simplexe

b_i : seconds membres des contraintes

c_j : coefficients de la fonction économique

	max		40	30	0	0	0	0
B	C_B	P_0	x_1	x_2	t_1	t_2	t_3	t_4
t_1	0	8	1	0	1	0	0	0
t_2	0	6	0	1	0	1	0	0
t_3	0	15	1	2	0	0	1	0
t_4	0	18	2	1	0	0	0	1
		0	-40	-30	0	0	0	0

Z_0

$Z_j - c_j$

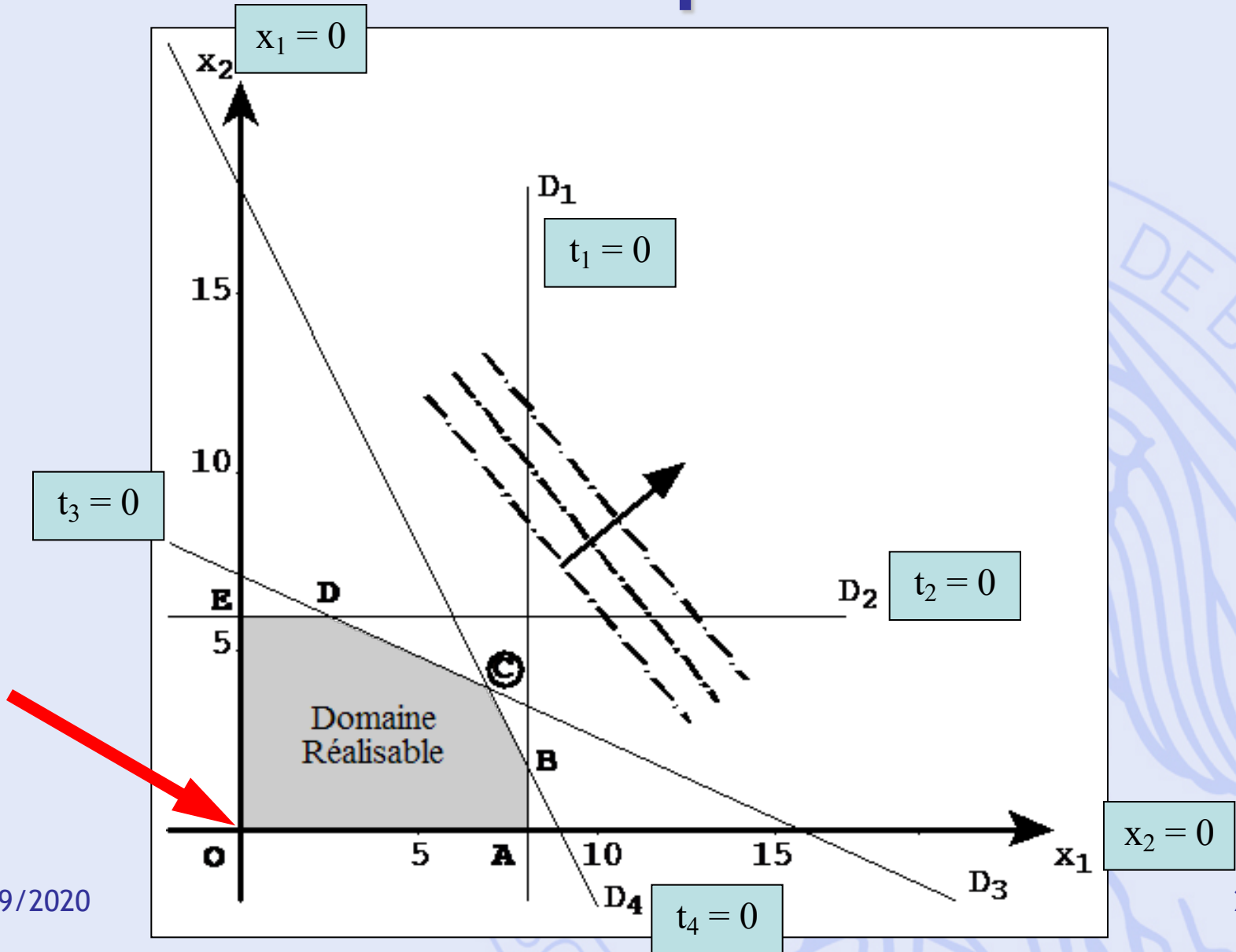
variables de base (col. matrice unité)

Tableau simplexe

- 1^{ère} s.b.r. explicitée :
 - Variables de base : $t_1=8, t_2=6, t_3=15, t_4=18$
 - Variables hors base : $x_1=0, x_2=0$
 - Valeur de la fonction économique : $z_0=0$

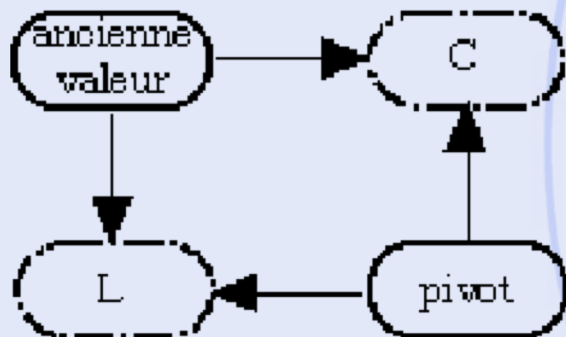
	max		40	30	0	0	0	0
B	C_B	P_0	x_1	x_2	t_1	t_2	t_3	t_4
t_1	0	8	1	0	1	0	0	0
t_2	0	6	0	1	0	1	0	0
t_3	0	15	1	2	0	0	1	0
t_4	0	18	2	1	0	0	0	1
		0	-40	-30	0	0	0	0

Exemple 2



Règles de changement de base

- La ligne du pivot est divisée par la valeur du pivot (a_{rk}).
- Les autres lignes du tableau simplexe sont transformées par la règle du rectangle :



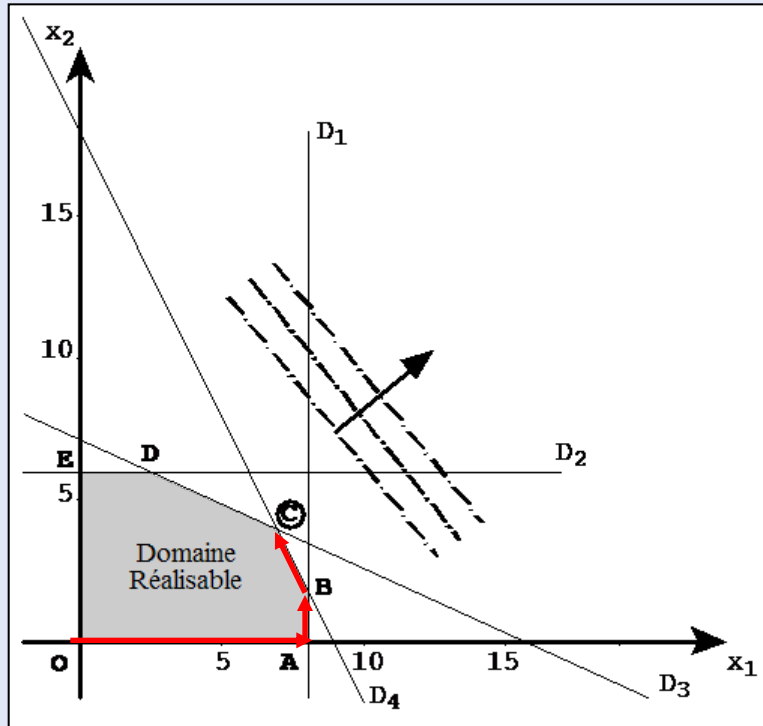
$$\text{nouveau} = \text{ancien} - \frac{L \times C}{\text{pivot}}$$

Exemple 2

1^{er} changement de base

max			40	30	0	0	0	0
B	C _B	P ₀	x ₁	x ₂	t ₁	t ₂	t ₃	t ₄
t ₁	0	8	1	0	1	0	0	0
t ₂	0	6	0	1	0	1	0	0
t ₃	0	15	1	2	0	0	1	0
t ₄	0	18	2	1	0	0	0	1
		0	-40	-30	0	0	0	0
x ₁	40	8	1	0	1	0	0	0
t ₂	0	6	0	1	0	1	0	0
t ₃	0	7	0	2	-1	0	1	0
t ₄	0	2	0	1	-2	0	0	1
		320	0	-30	40	0	0	0

Exemple 2

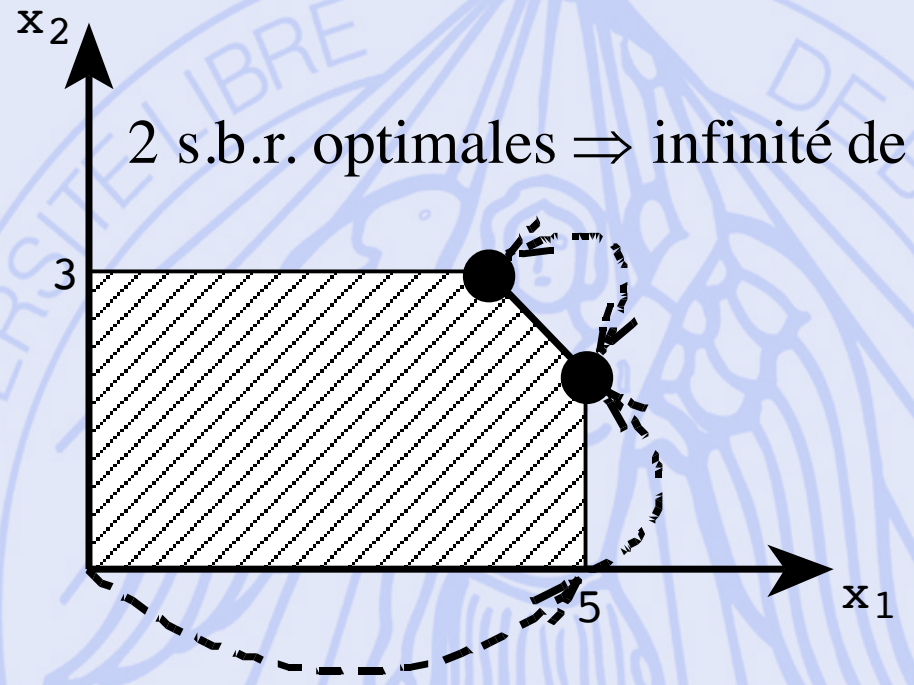


$O \rightarrow A \rightarrow B \rightarrow C$

	max		40	30	0	0	0	0
B	C_B	P_0	x_1	x_2	t_1	t_2	t_3	t_4
t_1	0	8	1	0	1	0	0	0
t_2	0	6	0	1	0	1	0	0
t_3	0	15	1	2	0	0	1	0
t_4	0	18	2	1	0	0	0	1
		0	-40	-30	0	0	0	0
x_1	40	8	1	0	1	0	0	0
t_2	0	6	0	1	0	1	0	0
t_3	0	7	0	2	-1	0	1	0
t_4	0	2	0	1	-2	0	0	1
		320	0	-30	40	0	0	0
x_1	40	8	1	0	1	0	0	0
t_2	0	4	0	0	2	1	0	-1
t_3	0	3	0	0	3	0	1	-2
x_2	30	2	0	1	-2	0	0	1
		380	0	0	-20	0	0	30
x_1	40	7	1	0	0	0	-1/3	2/3
t_2	0	2	0	0	0	1	-2/3	1/3
t_1	0	1	0	0	1	0	1/3	-2/3
x_2	30	4	0	1	0	0	2/3	-1/3
		400	0	0	0	0	20/3	50/3

Exemple 3

$$\begin{cases} \text{Max } z = x_1 + x_2 \\ x_1 \leq 5 \\ \quad x_2 \leq 3 \\ x_1 + x_2 \leq 7 \\ x_1, x_2 \geq 0 \end{cases}$$



Exemple 3

Max			1	1	0	0	0
B	C _B	P ₀	x ₁	x ₂	t ₁	t ₂	t ₃
t ₁	0	5	1	0	1	0	0
t ₂	0	3	0	1	0	1	0
t ₃	0	7	1	1	0	0	1
		0	œl	œl	0	0	0
x ₁	1	5	1	0	1	0	0
t ₂	0	3	0	1	0	1	0
t ₃	0	2	0	1	œl	0	1
		5	0	œl	1	0	0
x ₁	1	5	1	0	1	0	0
t ₂	0	1	0	0	1	1	œl
x ₂	1	2	0	1	œl	0	œl
		7	0	0	0	0	1
x ₁	1	4	1	0	0	œl	1
t ₁	0	1	0	0	1	1	œl
x ₂	1	3	0	1	0	1	0
		7	0	0	0	0	1

Méthode de la base artificielle

- Quand il n'y a pas de s.b.r. explicitée !
- Exemple 4 :

$$\begin{cases} \text{Min } z = x_1 + 2x_2 \\ 2x_1 + x_2 \geq 8 \\ x_1 + x_2 \geq 6 \\ x_1 + 5x_2 \geq 10 \\ x_1, x_2 \geq 0 \end{cases}$$

$$\begin{cases} \text{Min } z = x_1 + 2x_2 \\ 2x_1 + x_2 - t_1 = 8 \\ x_1 + x_2 - t_2 = 6 \\ x_1 + 5x_2 - t_3 = 10 \\ x_1, x_2, t_1, t_2, t_3 \geq 0 \end{cases}$$

Méthode de la base artificielle

- Principe : ajouter des variables artificielles pour construire une s.b.r. explicitée dans le PL.
- Remplacer le PL par un PL « artificiel » dont la s.o. (obtenue par l'algorithme du simplexe) correspondra à celle du PL de départ (si pas contradictoire).

Méthode de la base artificielle

- Soit le PL(1) à résoudre :

$$\begin{cases} \text{Min } z = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m \quad (b_i \geq 0) \\ x_j \geq 0, \quad j = 1, \dots, n \end{cases}$$

Méthode de la base artificielle

- On remplace le PL(1) par le PL(2) suivant :

$$\begin{cases} \text{Min } z = \sum_{j=1}^n c_j x_j - M \sum_{i=1}^m v_i \\ \sum_{j=1}^n a_{ij} x_j + v_i = b_i, \quad i = 1, \dots, m \\ x_j \geq 0 \quad (j = 1, \dots, n), \quad v_i \geq 0 \quad (i = 1, \dots, m) \end{cases}$$

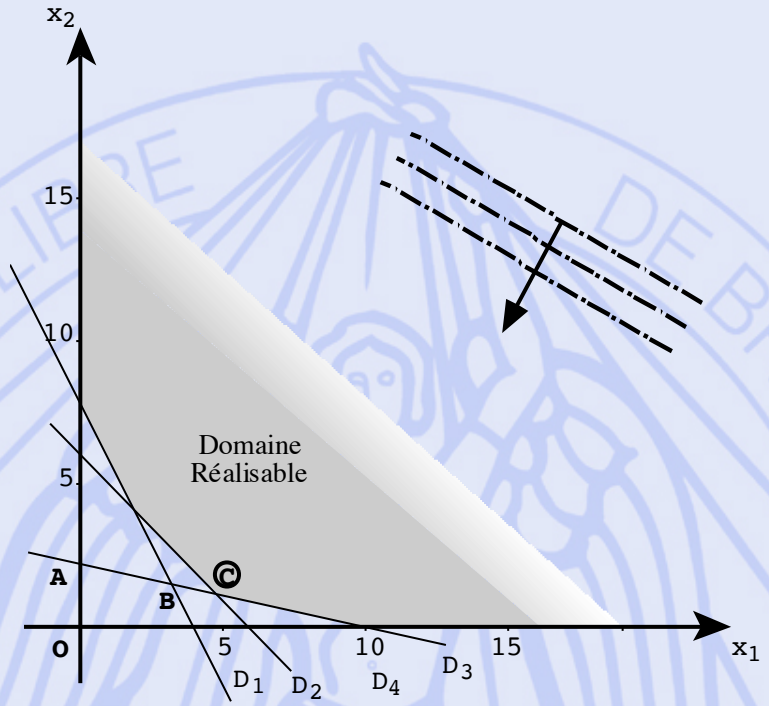
où les v_i sont des variables artificielles et M est un nombre positif arbitrairement grand.

Propriétés

- $PL(2) \Leftrightarrow PL(1)$ ssi $v_i = 0 \forall i=1, \dots, m$
- s.o. finie pour $PL(1) \Rightarrow$ s.o. de $PL(2)$ aussi.
- Conclusion :
 - $PL(2)$ admet une s.o. avec $v_i = 0 \forall i \Rightarrow$ s.o. de $PL(1)$.
 - $PL(2)$ admet une s.o. avec $\exists v_i > 0 \Rightarrow PL(1)$ contradictoire.

Exemple 4

$$\begin{cases} \text{Min } z = x_1 + 2x_2 \\ 2x_1 + x_2 \geq 8 \\ x_1 + x_2 \geq 6 \\ x_1 + 5x_2 \geq 10 \\ x_1, x_2 \geq 0 \end{cases}$$



$$\begin{cases} \text{Min } z = x_1 + 2x_2 + Mv_1 + Mv_2 + Mv_3 \\ 2x_1 + x_2 - t_1 + v_1 = 8 \\ x_1 + x_2 - t_2 + v_2 = 6 \\ x_1 + 5x_2 - t_3 + v_3 = 10 \\ x_1, x_2, t_1, t_2, t_3 \geq 0 \end{cases}$$

Exemple 4

B	C _B	P ₀	1 x ₁	2 x ₂	0 t ₁	0 t ₂	0 t ₃	M v ₁	M v ₂	M v ₃
v ₁	M	8	2	1	œ1	0	0	1	0	0
v ₂	M	6	1	1	0	œ1	0	0	1	0
v ₃	M	10	1	5	0	0	œ1	0	0	1
		0	œ1	œ2	0	0	0	0	0	0
		24M	4M	7M	œM	œM	œM	0	0	0
v ₁	M	6	9/5	0	œ1	0	1/5	1	0	
v ₂	M	4	4/5	0	0	œ1	1/5	0	1	
x ₂	2	2	1/5	1	0	0	œ1/5	0	0	
		4	œ3/5	0	0	0	œ2/5	0	0	
		10M	13/5M	0	œM	œM	2/5M	0	0	
x ₁	1	10/3	1	0	œ5/9	0	1/9		0	
v ₂	M	4/3	0	0	4/9	œ1	1/9		1	
x ₂	2	4/3	0	1	1/9	0	œ2/9		0	
		6	0	0	œ1/3	0	œ1/3		0	
		4/3M	0	0	4/9M	œM	1/9M		0	
x ₁	1	5	1	0	0	œ5/4	1/4			
t ₁	0	3	0	0	1	œ9/4	1/4			
x ₂	2	1	0	1	0	1/4	œ1/4			
		7	0	0	0	œ3/4	œ1/4			
		0	0	0	0	0	0			

PL en variables 0-1

- Définition (forme canonique) :

$$\min (c_1x_1 + c_2x_2 + \dots + c_nx_n)$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m \end{cases} \quad \begin{cases} \min cX \\ AX \geq b \\ X \in B_2^n \end{cases}$$

$$x_1, x_2, \dots, x_n \in \{0;1\} \text{ et } c_1, c_2, \dots, c_n \geq 0$$

$$\text{avec } B_2 = \{0;1\}$$

Terminologie

- Solution : $X \in B_2^n$
- Solution admissible (solution réalisable) :

Solution X avec : $t = b - AX \leq 0$

- NB: Nombre de s. admissibles fini : $\leq 2^n$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^{10} = 1.024$$

$$2^{20} = 1.048.576$$

$$2^{100} = 1,3 \times 10^{30}$$

Algorithme de résolution

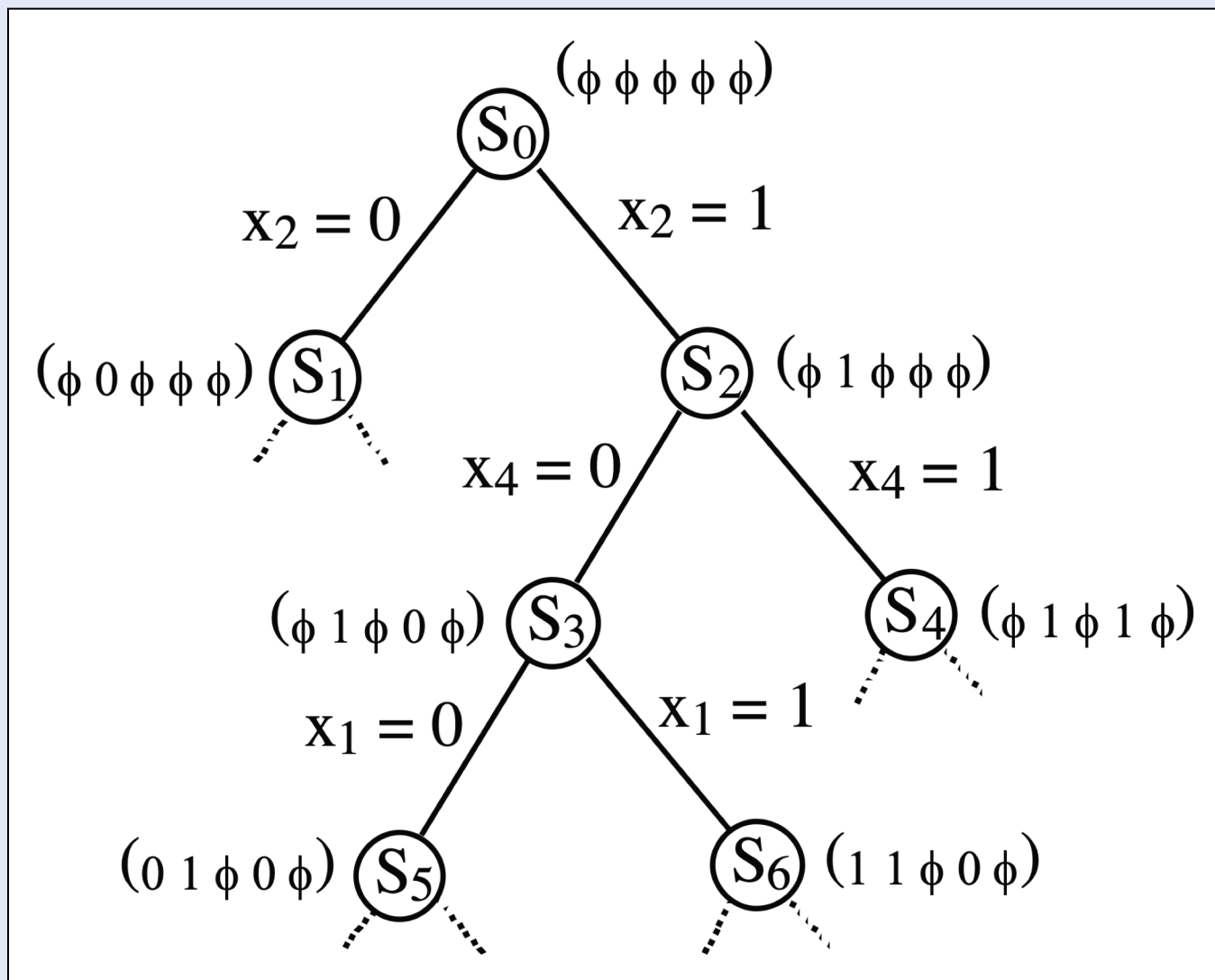
- Procédure de séparation et d'évaluation séquentielle (PSES).
- Construction d'une arborescence :
 - Chaque sommet correspond à un ensemble de solutions.
 - Au départ, toutes les variables sont libres (valeur non fixée).
 - Progressivement, on fixe les valeurs de certaines variables.

PSES

- Principe de séparation :
 - Choisir une variable libre et construire deux sous-ensembles de solutions, suivant que cette variable est portée à 0 ou à 1. Continuer ainsi jusqu'à obtenir des sous-ensembles sondables.
- Utilisation de vecteurs ϕ -booléens :
 - Vecteurs dont les composantes peuvent être égales à 0, 1 ou ϕ (si la variable correspondante est libre) \rightarrow correspondent à des sous-ensembles de solutions.

Exemple

- $n = 5$



Ensemble sondable

- Ensemble qui ne doit plus être séparé parce que :
 1. On connaît la meilleure solution de l'ensemble.
 2. On connaît une solution meilleure que toutes celles de l'ensemble.
 3. On sait que l'ensemble ne contient aucune solution admissible.

Principe de la PSES

- L'arborescence est parcourue séquentiellement, en commençant par les branches de droite (variables portées à 1) jusqu'à rencontrer un ensemble sondable.
- A chaque étape, un **vecteur de régression** (liste), noté R , est mis à jour.

Vecteur de régression

- Au départ : $R = \emptyset$,
- Lorsqu'une variable est portée à 1, on ajoute son indice dans R :
 x_i, x_j, x_k successivement portées à 1 $\rightarrow R = (i j k)$
- Lorsqu'un ensemble sondable est atteint, on remonte d'un niveau dans l'arborescence et on porte à 0 la dernière variable fixée à 1. Dans R , l'indice correspondant est souligné :
 $\rightarrow R = (i j \underline{k})$
- On continue à se déplacer vers la droite jusqu'à obtenir un nouvel ensemble sondable :
 x_1 portée à 1 $\rightarrow R = (i j \underline{k} 1)$

Vecteur de régression (2)

- Si l'ensemble est sondable, on remonte et on fixe x_1 à 0 :
→ $R = (i \ j \ \underline{k} \ \underline{l})$
- Si le sous-ensemble obtenu est sondable, on remonte dans l'arborescence d'autant de niveaux qu'il y a de variables soulignées à la fin du vecteur R , et on porte à 0 la première variable non-soulignée rencontrée :
→ $R = (i \ \underline{j})$
- On continue jusqu'à ce que $R = \emptyset$. Toute l'arborescence a alors été sondée (du moins implicitement) et la meilleure solution trouvée est une solution optimale du programme.

Fonction d'évaluation

- Soit S un sous-ensemble de solutions de l'arborescence :
 - vecteur ϕ -booléen X .
 - on construit le vecteur booléen \underline{X} en remplaçant les ϕ de X par des 0.
 - on pose $B(S) = c \underline{X}$
- $B(S)$ est une borne inférieure :

$$c \underline{X} \leq \min_{X \in S} \sum_{j=1}^n c_j X_j$$

Tests d'optimalité et d'admissibilité

- Soit S un sous-ensemble de solutions à analyser :
 - vecteur ϕ -booléen X
 - X_{opt} - la meilleure solution admissible trouvée
 - V_{opt} - la valeur de la fonction économique en X_{opt}
(initialisation : $V_{opt} = \infty$)

Tests d'optimalité

- T_1 : test d'optimalité
 - Si $c \underline{X} \geq V_{\text{opt}}$: S ne peut contenir de solution meilleure que $X_{\text{opt}} \rightarrow S$ est sondable \rightarrow régression.
 - Si $c \underline{X} < V_{\text{opt}}$: S peut contenir des solutions admissibles meilleures que $X_{\text{opt}} \rightarrow$ test T_2
- T_2 : test d'optimalité conditionnelle
 - $\forall x_j = \phi$: on pose $x_j = 0$ si $c \underline{X} + c_j \geq V_{\text{opt}}$
 \rightarrow permet d'accélérer la procédure.

Tests d'admissibilité

- T_3 : test d'admissibilité
 - Soit $t = b - A \underline{X}$
 - Si $t \leq 0$:
 - \underline{X} est une solution admissible meilleure que X_{opt}
(par T_1)
 - on pose $X_{opt} = \underline{X}$ et $V_{opt} = c \underline{X}$
 - régression.
 - Si t possède au moins une composante positive :
 - \underline{X} n'est pas admissible.
 - test T_4

Tests d'admissibilité

- T_4 : test d'admissibilité conditionnelle
 - Soit $p_i = t_i - \sum_{j: x_j = \phi} \max(0, a_{ij})$
 - Si $\exists p_i > 0$:
S ne peut pas contenir de solution admissible.
→ régression.
 - Si $p_i \leq 0 \forall i$:
S peut contenir une solution admissible.
→ sonder S en le séparant
en deux sous-ensembles.

Principe de séparation

- Porter une des variables libres à 0 (branche de gauche) et à 1 (branche de droite).

- Choix d'une variable libre :

- $\forall x_j = \phi : S(x_j) = \sum_{i: t_i > 0} \max(0, t_i - a_{ij})$

- on choisit x_k telle que : $S(x_k) = \min_{j: x_j = \phi} S(x_j)$

Présentation en tableau

S		$X = (\dots\dots\dots)$
		$\underline{X} = (\dots\dots\dots)$
		$X_{opt} = (\dots\dots\dots)$
$c \underline{X} = \dots$		$V_{opt} = \dots$
T ₂		
T ₃		T ₄
$t = \begin{pmatrix} \\ \\ \end{pmatrix}$		$p = \begin{pmatrix} \\ \\ \end{pmatrix}$
S(x ₁)	...	S(x _n)
R : (\dots\dots\dots)		

Exemple

$$\begin{cases} \min(5x_1 + 7x_2 + 3x_3 + x_4) \\ x_1 - 2x_2 + 3x_3 - 2x_4 \geq 1 \\ -2x_1 + 6x_2 - x_3 + 4x_4 \geq 1 \\ x_1 + x_2 + x_3 + x_4 \geq 2 \\ x_i \in \{0;1\}, \quad i = 1,2,3,4 \end{cases}$$

Résolution (1)

S ₀		$X = (\phi \phi \phi \phi)$	
		$\underline{X} = (0 0 0 0)$	
		$X_{opt} = /$	
$c \underline{X} = 0$		$V_{opt} = \infty$	
T ₂			
T ₃		T ₄	
$t = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$		$p = \begin{pmatrix} \infty \\ \infty \\ \infty \end{pmatrix}$	
$S(x_1) = 4$		$S(x_3) = 3$	
$S(x_2) = 4$		$S(x_4) = 5$	
$R : (3)$			

S ₁		$X = (\phi \phi 1 \phi)$	
		$\underline{X} = (0 0 1 0)$	
		$X_{opt} = /$	
$c \underline{X} = 3$		$V_{opt} = \infty$	
T ₂			
T ₃		T ₄	
$t = \begin{pmatrix} \infty \\ 2 \\ 1 \end{pmatrix}$		$p = \begin{pmatrix} \infty \\ \infty \\ \infty \end{pmatrix}$	
$S(x_1) = 4$			
$S(x_2) = 0$		$S(x_4) = 2$	
$R : (3 2)$			

Résolution (2)

S ₂ X = (φ 1 1 φ)	
<u>X</u> = (0 1 1 0)	
X _{opt} = /	
c <u>X</u> = 10	V _{opt} = ∞
T ₂	
T ₃	T ₄
t = $\begin{pmatrix} 0 \\ \alpha 4 \\ 0 \end{pmatrix}$	
S(x ₁) = 4	S(x ₃) = 3
S(x ₂) = 4	S(x ₄) = 5
R : (3 <u>2</u>)	

S ₃ X = (φ 0 1 φ)	
<u>X</u> = (0 0 1 0)	
X _{opt} = (0 1 1 0)	
c <u>X</u> = 3	V _{opt} = 10
T ₂	
T ₃	T ₄
t = $\begin{pmatrix} \alpha 2 \\ 2 \\ 1 \end{pmatrix}$	p = $\begin{pmatrix} \alpha 3 \\ 2 \\ \alpha 1 \end{pmatrix}$
R : (<u>3</u>)	

Résolution (3)

S ₄ X = (φ φ 0 φ)	
<u>X</u> = (0 0 0 0)	
X _{opt} = (0 1 1 0)	
c <u>X</u> = 0	V _{opt} = 10
T ₂	
T ₃	T ₄
t = $\begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$	p = $\begin{pmatrix} 0 \\ \alpha 5 \\ \alpha 1 \end{pmatrix}$
S(x ₁) = 4	
S(x ₂) = 4	S(x ₄) = 5
R : (<u>3</u> 1)	

S ₅ X = (1 φ 0 φ)	
<u>X</u> = (1 0 0 0)	
X _{opt} = (0 1 1 0)	
c <u>X</u> = 5	V _{opt} = 10
T ₂ x ₂ = 0	
T ₃	T ₄
R : (<u>3</u> 1 <u>2</u>)	

Résolution (4)

S ₆ $X = (1\ 0\ 0\ \phi)$	
<u>X</u> = (1 0 0 0)	
X _{opt} = (0 1 1 0)	
$c \underline{X} = 5$	V _{opt} = 10
T ₂	
T ₃	T ₄
$t = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}$	$p = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix}$
R : (<u>3</u> <u>1</u>)	

S ₇ $X = (0\ \phi\ 0\ \phi)$	
<u>X</u> = (0 0 0 0)	
X _{opt} = (0 1 1 0)	
$c \underline{X} = 0$	V _{opt} = 10
T ₂	
T ₃	T ₄
$t = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$	$p = \begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix}$
R : (∅)	