Vocabulaire

Graph : ensemble arc et sommet
Predecesseur : $\Gamma^-(x)$ } = sommet adjacent
Successeur : $\Gamma^+(x)$

ω^+ arc incident $x \to$ extérieur
$d^+ = card(\omega^+) = \#(\Gamma^+) =$ nb d'arc

ω^- arc incident ext $\to x$
$d^- = card(\omega^-)$

$d = d^\circ$ total $= d^+ + d^-$

p-graph : p arc entre x, y
↳ nb max d'arc entre 2 pt

graph partiel : ↓ nb arc
Sous graphe : ↓ nb arc ↓ nb som
Graphe simple : 1-graph ss boucle
Graph symétrique : si arc $(x,y) \in U$ $\Rightarrow (y,x) \in U$
Antisymétrique
Arborescence : 1 seul prédecesseur
Orienté / non orienté
Complet : un arc entre chaque som
Chemin :
Chaîne :
Chemin hamiltonien : passe par tt les sommet 1 seule fois
Cycle : bouclage $x_0 \to x_0$
Connexité : non orienté : tous som sont connectés par 1 chaîne
Fortement connexe : trajet $x \to y$ et $y \to x$
Chemin eulerien : parcourir 1 seule fois les arcs

Arbre = graph connexe ss cycle
n : nb som
m : nb arc
p = nb composant connexe
c = nb de cyclomatique
tout cycle ind

$\boxed{p = n - m + c}$
Arbre $p = 1 / c = 0$

Algorithme de Kruskal

↳ calcul l'arbre maximal de poids minimum

	A	B	C	D	E	F
A	-	5	10	22	15	23
B		-	14	9	20	21
C			-	13	12	9
D				-	7	6
E					-	4
F						-

	1	2	3	4	5
A	0	2	2	2	1
B	0	2	2	2	1
C	0	0	0	1	1
D	0	0	1	1	1
E	1	1	1	1	1
F	1	1	1	1	1
Arc ajouté	EF	AB	FD	FC DE	DB

Algorithme de Bellman

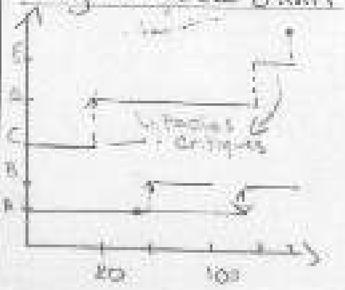
↳ recherche du sommet optimum
$\omega(x_p)$ = poids du chemin min entre $x \to y$
$\omega(x,x) = ct$
$\Pi(x)$ = poids chemin min entre entrée $\to x$

$\boxed{\Pi(x) = \min[\Pi(y) + \omega(y \to x)]}$
$A(x)$ = som optimum

	1	2	3	4
0	1	4	22	32
1		2	14	~~7~~
2			7	13
3				6

$\Pi(1) = 1$ $A(1) = 0$
$\Pi(2) = 3$ $A(2) = 1$
$\Pi(3) = 10$ $A(3) = 2$
$\Pi(4) = 8$ $A(4) = 1$

Diagramme de Gantt



Méthode Potentiel Tâche

- durée minimale
- date au plus tôt : t_i
- date au plus tard : t_i'
- marge : $m_i = t_i' - t_i$
- tâches critiques : $m_i = 0$ fin $(=0)$

Π_i - longueur plus long chemin de début $\to i$
γ_i : longueur plus long chemin $i \to$ fin

$t_i' = t_{fin}' - \gamma_I$
$m_i = t_i' - t_i$
$\Pi_i = \max_{J \in \Gamma^-(I)} (\Pi_J + d(i,j))$
$\gamma_i = \max_{J \in \Gamma^+(I)} (d(I,J) + \gamma_J)$

Programmation dynamique

Hamilton - Jacobi - Bellman

- Bellman

$V_t(X)$: longueur trajectoire plus court de debut $\to$ som (t, x)
Initialisat : $V_0(X) = 0$ si $X \in$ G sinon ∞

U : commande
Equat° d'état : $X_{t+1} = f_t(X_t, U_t)$
Itérat° pr t = 0 ... T-1
$V_{t+1}(X) = \min[V_t(Y) + ct(Y,U)]$
avec $ct(Y,U)$ = cout $((t,Y) \to (t+1,X))$

- Bellman inverse

$W_t(X) = \min[W_{t+1}(Z) + ct(X,U)]$
avec $ct(X,U)$ = cout $((t,X) \to (t+1,Z))$

$Y_{i+1} = y_i + x_{i+1}$ } Éq d'état
Variable d'état — commande

Critère : $\max \sum y_i$

Forme standard :
Variables = cst

Programmation linéaire

$Ax = b$
↳ A matrice / bloc

Algorithme du simplex

$\max(cx) \to$ critère
$\begin{cases} Ax \le b \\ x \ge 0 \end{cases}$

$\Leftrightarrow \begin{cases} Ax + y = b \\ y \ge 0 \\ x \ge 0 \end{cases}$ ← variable d'écart

et $Ax \ge b \Leftrightarrow \begin{cases} Ax - y = b \\ y \ge 0 \end{cases}$

et $Ax = b \Leftrightarrow \begin{cases} Ax \le b \\ -Ax \le -b \end{cases}$

$\min cx = \max(-cx)$

Representat° du pb en 2D par zone en fct des égalité

Algorithme Simplex primal

$Ax = b$ A_x : $\to$ variable x_i
B : base du syst
N : hors base
$Bx_B + Nx_N = b$ (1)
Solut° $\begin{cases} x_B = B^{-1}b \\ x_N = 0 \end{cases}$

(1) $\Leftrightarrow x_B = B^{-1}b - B^{-1}Nx_N$

Sol $\beta = B^{-1}b = x_B$

Tableau du simplex

x_B	β	$B^{-1}A$
	b	x

$= T$

$Tx = \beta \Leftrightarrow B^{-1}Ax = B^{-1}b$

Pivotage
$\to T' = $ | β' | $B'^{-1}A$ |

Avec $T' = (B'^{-1}B)T$

variable sortante $x(s)$
variable entrante x_e

B	$x_B(s)$	T_{se}
b	x	x_e

- Ligne s de T' : ligne s de T / T_{se}
- Ligne i de T' : Ligne i de T $- \frac{T_{ie}}{T_{se}} \times$ Ligne s de T

Choix des lignes :
- Sortante : $T_{se} > 0$
  - s réalise le minimum de β_i / T_{ie}

Tableau du simplex complet

z	$c_B\beta$	γ
x_B	β	$T = B^{-1}A$
	b	x

$= T$

$z = c_B B^{-1}b = c_B\beta$: valeur du critère pr solut base B
β = solut de base B avec $x_B = \beta$ [illegible]
$\gamma = c_B B^{-1}A - c$: cout reduit
c : critère
- Choix ligne entrante pr avoir que des zeros sur γ [illegible]

$\to \boxed{\gamma' = \gamma - \gamma_E \times \text{Ligne s de } T'}$

Cas solut° ∞
e colonne entrante < 0
identificat° variable de base
↳ partie matrice identité

Règles à suivre si colonne tout négatif :
$x(i) = \beta(i) = \beta_i - t T_{ie}$
$x_e(t) = t$
$x_j = 0$ $\forall j \in N$, $j \ne e$
$t \ge 0$

Méthode

1 - mise sous forme standard
2 - introduire variable d'écart

Ex. $\begin{cases} \max z = 3x_1 + 2x_2 \\ x_1 - 2x_2 + y_1 = 8 \\ x_1 - 4x_2 + y_2 = 4 \end{cases}$

3 - Calcul de z_0 et y_0
$z_0 = \max 3x_1 + 2x_2 = 0$
$y_1 = \ldots$

z	0	-3	-2	0	0
y_1	8	1	-2	1	0
y_2	4	1	-4	0	1
	b	x_1	x_2	y_1	y_2

4 - Choix des colonne entrante
↳ le plus petit en c_u

5 - Calcul et colonne sortante :
$b_1 / T_{1e} = 8/1$
$b_2 / T_{2e} = 4/1$
↳ prend le + petit mais > 0
↳ nouvelle base (y_1, x_1)

Variable artificielle
ajoute z_i si solut° avec $x = 0$ donne des $y_i < 0$

Règle de Bland
↳ plus long / si $b = (0)$
↳ on choisit som colonne entrante la $1^{er} < 0$

Variable d'écart artifi.
Si inegalité $\ge$ (si syst augmenté)
$x_1 + x_{e2} + z_2 = 6 \Rightarrow x_1 + z_2$
On pose $W = \max(-z_1 - z_2 - \ldots)$

ligne W = ligne W - ligne 1 - ligne 2 - ligne 3 ...
$-(\underbrace{0\ 0\ 0}_{x}, \underbrace{-1\ -1\ -1}_{z})$

↳ colonne entrante avec W petit petit
On s'arrête quand on a une base sans Z puis reprendre calcul z → x2