Chapitre 7 : Programmation dynamique

J.-F. Scheid

v2.1

Plan du chapitre

Programmation dynamique :

- Programmation dynamique inventée par Bellman ($\sim$ 1954) pour résoudre des pb de chemins optimaux (longueur max. ou min.)

R. Bellman (1920-1984)

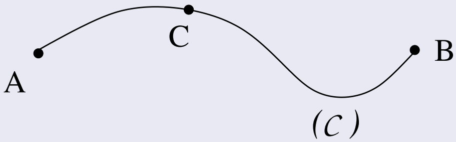
- La programmation dynamique est une méthode de construction d'algorithme très utilisée en optimisation combinatoire ($\rightarrow$ recherche de solution optimale dans un ensemble **fini** de solutions **mais très grand**).

- La programmation dynamique est une méthode de construction d'algorithme très utilisée en optimisation combinatoire ($\rightarrow$ recherche de solution optimale dans un ensemble **fini** de solutions **mais très grand**).

- Il s'agit d'une méthode d'énumération implicite (idem PSE) : on retient ou rejette des sous-ensembles de solutions mais on ne construit pas toutes les solutions. On rejette certaines solutions sans les avoir contruites explicitement si elles appartiennent à un sous-ensemble qui n'est pas intéressant.

Principe d'optimalité de Bellman

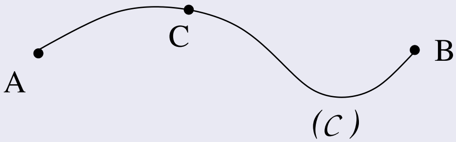
Un chemin optimal est formé de sous-chemins optimaux : Si $(\mathcal{C})$ est un chemin *optimal* allant de A à B et si C appartient à $(\mathcal{C})$ alors les sous-chemins de $(\mathcal{C})$ allant de A à C et de C à B sont *optimaux*.



Démonstration par l'absurde.

Principe d'optimalité de Bellman

Un chemin optimal est formé de sous-chemins optimaux : Si $(\mathcal{C})$ est un chemin *optimal* allant de A à B et si C appartient à $(\mathcal{C})$ alors les sous-chemins de $(\mathcal{C})$ allant de A à C et de C à B sont *optimaux*.

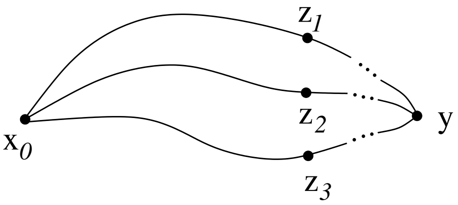


Démonstration par l'absurde.

Ce principe appliqué de façon *séquentielle* fournit des **formules récursives** pour la recherche de chemins optimaux.

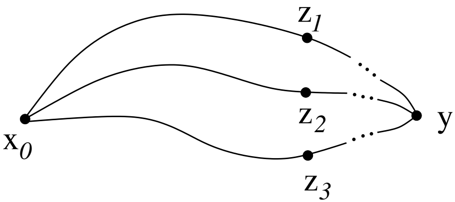
Un exemple.

On cherche le chemin le plus court allant de x_0 fixé à y quelconque dans un graphe valué.



Un exemple.

On cherche le chemin le plus court allant de x_0 fixé à y quelconque dans un graphe valué.



On note $F(y)$ la longueur *minimale* de tous les chemins allant de x_0 à y.

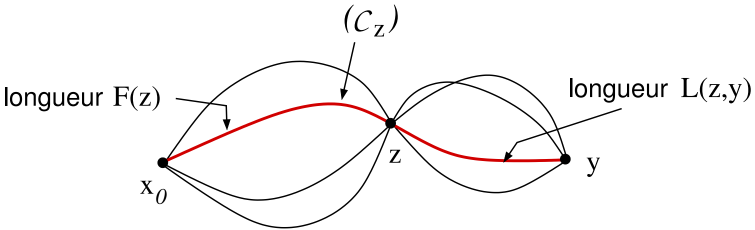
Formule récursive

$$F(y) = \min_{z \neq y} (F(z) + L(z, y)) \tag{1}$$

où $L(z, y)$ est la longueur *minimale* de tous les chemins allant de z à y.

Démonstration de la formule récursive (1).

Soit un point z fixé dans le graphe. On considère un chemin minimal $(\mathcal{C}_z)$ allant de x_0 à y et passant par z. Par le principe d'optimalité de Bellman, le sous-chemin allant de x_0 à z est minimal et a pour longueur $F(z)$. De même, le sous-chemin allant de z à y est minimal et a pour longueur $L(z, y)$.



Donc la longueur du chemin $(\mathcal{C}_z)$ est $F(z) + L(z, y)$. On conclut en prenant le minimum sur z $\Rightarrow F(y) = \min_z (F(z) + L(z, y))$ $\square$

- La programmation dynamique appliquée à un problème donné, consiste à trouver une formulation récursive du problème.
- En procédant ensuite à un découpage étape par étape, on obtient une formule de récurrence.

Un exemple de Programmation linéaire en nombres entiers.

Pb avec coefficients positifs dans les contraintes :

$$(PLNE) \left\{ \begin{array}{l} \max\limits_{\mathbf{x}} \left[F(\mathbf{x}) = 6x_1 + 9x_2 + 8x_3 + 4x_4 \right] \\ 4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 8 \\ 3x_1 + 6x_2 + 4x_3 + 3x_4 \leq 7 \\ x_i \ \textbf{entiers}, \ \forall i \end{array} \right.$$

Un exemple de Programmation linéaire en nombres entiers.

Pb avec coefficients positifs dans les contraintes :

$$(PLNE) \quad \begin{cases} \max_{\mathbf{x}} \left[F(\mathbf{x}) = 6x_1 + 9x_2 + 8x_3 + 4x_4\right] \\ 4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 8 \\ 3x_1 + 6x_2 + 4x_3 + 3x_4 \leq 7 \\ x_i \ \textbf{entiers}, \ \forall i \end{cases}$$

- Pour $1 \leq k \leq n = 4$, on cherche la solution de $(PLNE)$ pour un second membre $\mathbf{d}$ quelconque ($\mathbf{d} \leq (8,7)^\top$) et en supposant que les variables au delà de la k-ième sont **nulles**.

Un exemple de Programmation linéaire en nombres entiers.

Pb avec coefficients positifs dans les contraintes :

$$(PLNE) \left\{ \begin{array}{l} \max\limits_{\mathbf{x}} [F(\mathbf{x}) = 6x_1 + 9x_2 + 8x_3 + 4x_4] \\ 4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 8 \\ 3x_1 + 6x_2 + 4x_3 + 3x_4 \leq 7 \\ x_i \ \textbf{entiers}, \ \forall i \end{array} \right.$$

- Pour $1 \leq k \leq n = 4$, on cherche la solution de $(PLNE)$ pour un second membre $\mathbf{d}$ quelconque ($\mathbf{d} \leq (8,7)^\top$) et en supposant que les variables au delà de la k-ième sont **nulles**.

  $\rightsquigarrow$ On note $F_k(\mathbf{d})$ la valeur maximale de la fonction objectif pour ce nouveau problème :

$$F^*_{\max} = F_4(8,7) = \max_{\begin{cases} A\mathbf{x} \leq \binom{8}{7} \\ \mathbf{x} \in \mathbb{N}^4 \end{cases}} F(\mathbf{x})$$

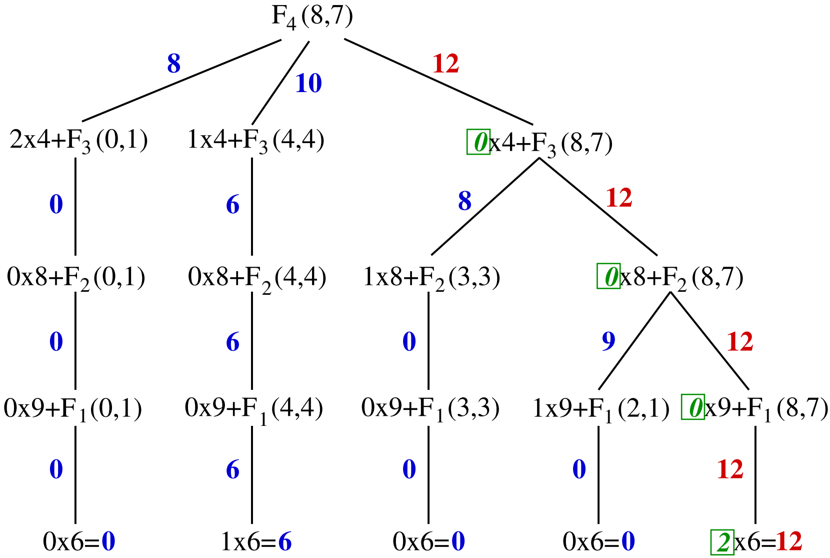
- Par le principe d'optimalité de Bellman, on peut alors écrire

$$F^*_{\max} = F_4(8,7) = \max_{\begin{cases} 8-4x_4 \geq 0 \\ 7-3x_4 \geq 0 \\ x_4 \text{ entier} \geq 0 \end{cases}} [4x_4 + F_3(8-4x_4, 7-3x_4)]$$

où

$$F_3(8-4x_4, 7-3x_4) = \max_{\begin{cases} 4x_1+6x_2+5x_3 \leq 8-4x_4 \\ 3x_1+6x_2+4x_3 \leq 7-3x_4 \\ x_1, x_2, x_3 \text{ entiers} \geq 0 \end{cases}} \left[F' = 6x_1 + 9x_2 + 8x_3\right]$$

$\Rightarrow$ formule de récurrence



Calculs par *remontée*.

Solution optimale entière : $x_1^* = 2$, $x_2^* = x_3^* = x_4^* = 0$ avec $\max F = 12$

Généralisation

PL en nombres entiers avec des coefficients positifs $A \geq 0$

$$(PLNE) \begin{cases} \max_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} \\ A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \\ \mathbf{x} \textbf{ entiers} \end{cases}$$

Principe : on cherche la solution de $(PLNE)$ pour un second membre $\mathbf{d} \leq \mathbf{b}$ quelconque en supposant que les variables au delà de la k-ième sont **nulles**. On note $F_k^*(\mathbf{d})$ la valeur maximale de la fonction objectif de ce nouveau problème :

$$F_k^*(\mathbf{d}) = \max_{\begin{cases} A\bar{\mathbf{x}}_k \leq \mathbf{d} \\ \bar{\mathbf{x}}_k \in \mathbb{N}^n \end{cases}} F(\bar{\mathbf{x}}_k) \quad \text{avec} \quad \bar{\mathbf{x}}_k = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Formule de récurrence

$$F_k^*(\mathbf{d}) = \max_{\begin{cases} x_k \ \textbf{entier} \geq 0 \\ A^k x_k \leq \mathbf{d} \end{cases}} \left[c_k x_k + F_{k-1}^*(\mathbf{d} - A^k x_k) \right]$$

où A^k est le vecteur de la k-ième colonne de la matrice A i.e. $(A^k)_i = a_{ik}$

Pour obtenir la solution optimale en utilisant les k premières variables, on examine toutes les valeurs possibles pour la k-ième variable x_k.

III. Problèmes de cheminement dans un graphe

1) Optimisation "**passé** $\rightarrow$ **futur**"

On cherche les plus courts chemins de x_0 fixé (racine du graphe) à y quelconque.

1) Optimisation "**passé** $\rightarrow$ **futur**"

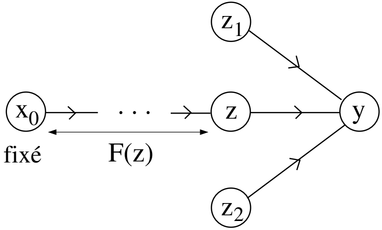
On cherche les plus courts chemins de x_0 fixé (racine du graphe) à y quelconque.

On note $F(y)$ la longueur de ces chemins **minimaux**. D'après le principe de Bellman,

$$F(y) = \min_{z \in P(y)} \left(F(z) + L(z, y) \right)$$

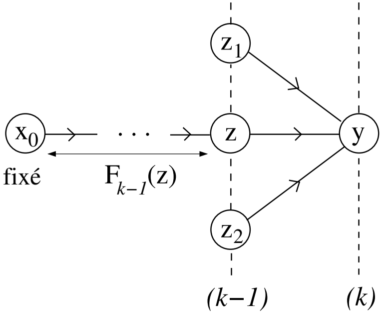
où $L(z, y)$ est la longueur de l'arête allant de z à y

$P(y)$ est l'ensemble des *sommets prédécesseurs* de y.



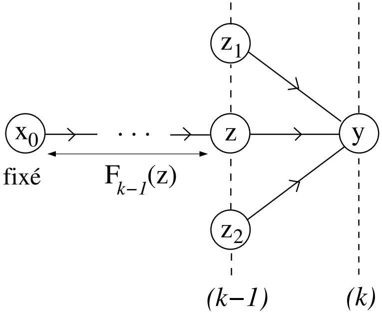
Formule de récurrence (passé→futur)

- $F_0(x_0) = 0, \quad F_0(y) = +\infty, \; y \neq x_0$
- $F_k(y) = \min_{z \in P(y)} \left(F_{k-1}(z) + L(z,y)\right)$



Formule de récurrence (passé→futur)

- $F_0(x_0) = 0, \quad F_0(y) = +\infty,\; y \neq x_0$
- $F_k(y) = \min\limits_{z \in P(y)} \left(F_{k-1}(z) + L(z, y)\right)$



Interprétation.

- $F_k(y) = +\infty$ s'il n'y a pas de chemin entre x_0 et y avec au plus k arêtes.
- $F_k(y)$ représente la longueur *minimale* entre x_0 et y, de tous les chemins qui vont de x_0 à y et ayant au plus k arêtes.

Chemin minimal. Pour déterminer le *chemin minimal*, on considère l'ensemble des sommets

$$P_k(y) = \{z \in P(y) \text{ tel que } F_k(y) = F_{k-1}(z) + L(z, y)\}$$

C'est l'ensemble des prédécesseurs de y qui réalisent le minimum.

Chemin minimal. Pour déterminer le *chemin minimal*, on considère l'ensemble des sommets

$$P_k(y) = \{z \in P(y) \text{ tel que } F_k(y) = F_{k-1}(z) + L(z, y)\}$$

C'est l'ensemble des prédécesseurs de y qui réalisent le minimum.

Chemin minimal

$$x_N = y;$$

$$x_k \in P_{k+1}(x_{k+1}) \quad \text{(jusqu'à la racine } x_0\text{)}$$

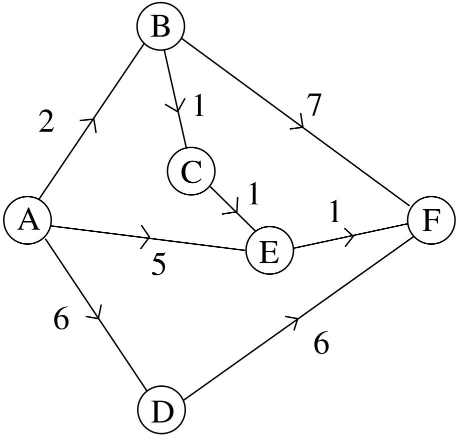
Algorithme de Bellman

$P(y)$ désigne l'ensemble des prédécesseurs du sommet y **au sens large** i.e. $y \in P(y) \Rightarrow F_k$ est toujours décroissante.

```
N : nb de sommets
x0 : racine du graphe
P : ens. des prédécesseurs (sens large)
l(z,y) : longueur de l'arête (z,y)

F0(x0)=0, F0(y) = ∞ pour y ≠ x0
Pour k de 1 à N-1
  |  Pour tout sommet y
  |    F1(y)= min(F0(z) + l(z,y); z ∈ P(y))
  |  Fin pour
  |  F0=F1
Fin pour
```

Exemple. Par programmation dynamique, déterminer dans le graphe suivant le plus court chemin de A à F.



Calcul des F_k.

	A	B	C	D	E	F
F_0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
F_1	0	**2**	$+\infty$	**6**	**5**	$+\infty$
F_2	0	2	**3**	6	5	**6**
F_3	0	2	3	6	**4**	6
F_4	0	2	3	6	4	**5**

Calcul des F_k.

	A	B	C	D	E	F
F_0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
F_1	0	**2**	$+\infty$	**6**	**5**	$+\infty$
F_2	0	2	**3**	6	5	**6**
F_3	0	2	3	6	**4**	6
F_4	0	2	3	6	4	**5**

- $F_1(B) = \mathbf{2}$; $P_1(B) = \{A\}$;
  $F_1(D) = \mathbf{6}$; $P_1(D) = \{A\}$;
  $F_1(E) = \mathbf{5}$; $P_1(E) = \{A\}$;

Calcul des F_k.

	A	B	C	D	E	F
F_0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
F_1	0	**2**	$+\infty$	**6**	**5**	$+\infty$
F_2	0	2	**3**	6	5	**6**
F_3	0	2	3	6	**4**	6
F_4	0	2	3	6	4	**5**

- $F_1(B) = \mathbf{2}$; $\quad P_1(B) = \{A\}$;
  $F_1(D) = \mathbf{6}$; $\quad P_1(D) = \{A\}$;
  $F_1(E) = \mathbf{5}$; $\quad P_1(E) = \{A\}$;
- $F_2(C) = F_1(B) + 1 = \mathbf{3}$; $\quad P_2(C) = \{B\}$;

  $$\begin{aligned} F_2(F) &= \min\left(F_1(F),\ F_1(B)+7,\ F_1(D)+6,\ F_1(E)+1\right) \\ &= \min(+\infty, 9, 12, \mathbf{6}) = \mathbf{6}; \quad P_2(F) = \{E\}; \end{aligned}$$

Calcul des F_k.

	A	B	C	D	E	F
F_0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
F_1	0	**2**	$+\infty$	**6**	**5**	$+\infty$
F_2	0	2	**3**	6	5	**6**
F_3	0	2	3	6	**4**	6
F_4	0	2	3	6	4	**5**

- $F_1(B) = \mathbf{2}$; $P_1(B) = \{A\}$;
  $F_1(D) = \mathbf{6}$; $P_1(D) = \{A\}$;
  $F_1(E) = \mathbf{5}$; $P_1(E) = \{A\}$;
- $F_2(C) = F_1(B) + 1 = \mathbf{3}$; $P_2(C) = \{B\}$;

  $$\begin{aligned} F_2(F) &= \min\left(F_1(F),\ F_1(B) + 7,\ F_1(D) + 6,\ F_1(E) + 1\right) \\ &= \min(+\infty, 9, 12, \mathbf{6}) = \mathbf{6}; \quad P_2(F) = \{E\}; \end{aligned}$$

- $$\begin{aligned} F_3(E) &= \min\left(F_2(E),\ F_2(A) + 5,\ F_2(C) + 1\right) \\ &= \min(5, 5, \mathbf{4}) = \mathbf{4}; \quad P_3(E) = \{C\}; \end{aligned}$$

Calcul des F_k.

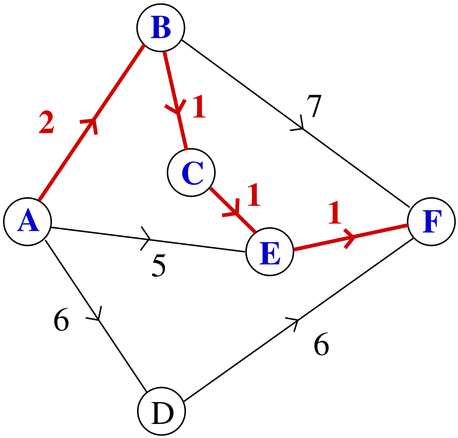
	A	B	C	D	E	F
F_0	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
F_1	0	**2**	$+\infty$	**6**	**5**	$+\infty$
F_2	0	2	**3**	6	5	**6**
F_3	0	2	3	6	**4**	6
F_4	0	2	3	6	4	**5**

- $F_1(B) = \mathbf{2}$; $\quad P_1(B) = \{A\}$;
  $F_1(D) = \mathbf{6}$; $\quad P_1(D) = \{A\}$;
  $F_1(E) = \mathbf{5}$; $\quad P_1(E) = \{A\}$;
- $F_2(C) = F_1(B) + 1 = \mathbf{3}$; $\quad P_2(C) = \{B\}$;

  $$\begin{aligned} F_2(F) &= \min\left(F_1(F),\ F_1(B)+7,\ F_1(D)+6,\ F_1(E)+1\right) \\ &= \min(+\infty, 9, 12, \mathbf{6}) = \mathbf{6}; \quad P_2(F) = \{E\}; \end{aligned}$$
- $$\begin{aligned} F_3(E) &= \min\left(F_2(E),\ F_2(A)+5,\ F_2(C)+1\right) \\ &= \min(5, 5, \mathbf{4}) = \mathbf{4}; \quad P_3(E) = \{C\}; \end{aligned}$$
- $$\begin{aligned} F_4(F) &= \min\left(F_3(F),\ F_3(B)+7,\ F_3(D)+6,\ F_3(E)+1\right) \\ &= \min(6, 9, 12, \mathbf{5}) = \mathbf{5}; \quad P_4(F) = \{E\}; \end{aligned}$$

Chemin optimal.

$$\left.\begin{aligned} x_4 &= F \\ x_3 &= P_4(F) = E \\ x_2 &= P_3(E) = C \\ x_1 &= P_2(C) = B \\ x_0 &= P_1(B) = A \end{aligned}\right. \quad \Rightarrow \quad \begin{array}{l} \text{chemin optimal } (A, B, C, E, F) \\ \text{de longueur minimale } 5 \end{array}$$



2) Optimisation "**futur $\to$ passé**"

On cherche les plus courts chemins de x quelconque à y_0 fixé (antiracine du graphe).

2) Optimisation "**futur** $\rightarrow$ **passé**"

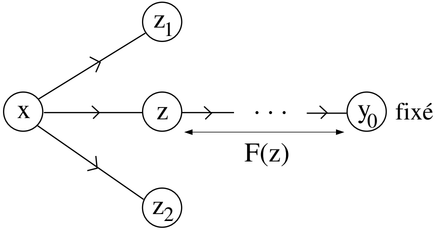
On cherche les plus courts chemins de x quelconque à y_0 fixé (antiracine du graphe).

On note $F(x)$ la longueur de ces chemins **minimaux**. D'après le principe de Bellman,

$$F(x) = \min_{z \in S(x)} \left(F(z) + L(x,z) \right)$$

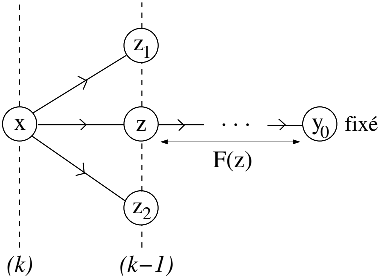
où $L(x,z)$ est la longueur de l'arête allant de x à z

$S(x)$ est l'ensemble des *sommets successeurs* de x.



Formule de récurrence (futur→passé)

- $F_0(y_0) = 0, \quad F_0(x) = +\infty, \; x \neq y_0$
- $F_k(x) = \min_{z \in S(x)} \left(F_{k-1}(z) + L(x,z) \right)$



Interprétation.

- $F_k(y) = +\infty$ s'il n'y a pas de chemin allant de x à y_0 avec au plus k arêtes.
- $F_k(y)$ représente la longueur *minimale* entre x et y_0, de tous les chemins qui vont de x à y_0 et ayant au plus k arêtes.

3) Remarques sur les formules de récurrence

• Dans les formules de récurrence précédentes, les valuations peuvent être de signes **quelconques**.

3) Remarques sur les formules de récurrence

- Dans les formules de récurrence précédentes, les valuations peuvent être de signes **quelconques**.

- ℹ Attention aux graphes *avec circuit*

Circuit dans un graphe

Un **circuit** dans un graphe est un chemin dont les extrémités sont confondues. La **valeur du circuit** est alors définie comme la somme de toutes les valuations des arêtes du circuit.

3) Remarques sur les formules de récurrence

• Dans les formules de récurrence précédentes, les valuations peuvent être de signes **quelconques**.

• ℹ Attention aux graphes *avec circuit*

Circuit dans un graphe

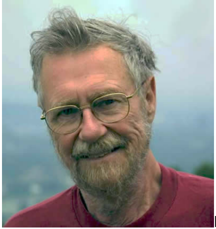
Un **circuit** dans un graphe est un chemin dont les extrémités sont confondues. La **valeur du circuit** est alors définie comme la somme de toutes les valuations des arêtes du circuit.

- Si un circuit a une valeur *strictement positive*, on ne peut pas boucler sur le circuit car sinon on ne fait qu'augmenter la longueur du chemin.
- Si un circuit a une valeur *strictement négative*, alors la longueur du chemin n'est plus minorée et on boucle...

Quelques algorithmes de programmation dynamique

Les différents algorithmes de programmation dynamique utilisant les formules de récurrences précédentes pour la recherche de chemins *minimaux* sont valables pour des graphes **sans circuit de valeur strictement négative**.

1. Algorithme de Bellman. Valuations de signes **quelconques** sur un graphe sans circuit de valeur négative. Nécessite une numérotation topologique des sommets : arêtes (i, j) avec $i < j$. Complexité en $\mathcal{O}(nm)$ avec n sommets, m arêtes.
2. Algorithme de Dijkstra. Valuations **positives** sur un graphe qui peut comporter des circuits (de valeurs positives). Complexité en $\mathcal{O}((n+m)\log(n))$.



E. Dijkstra (1930-2002)

3. ...

Tous ces algorithmes diffèrent essentiellement sur la façon de parcourir les sommets.

Par exemple,

- Bellman : on choisit un sommet dont tous les prédécesseurs ont déjà été traités.
- Dijkstra : on choisit le sommet avec la plus petite distance.

Algorithme de Dijkstra (Rappel)

```
E : ens. des sommets du graphe
x0 : racine du graphe
S : ens. des successeurs
l(z,y) : longueur de l'arête (z,y)

F0(x0)=0, F0(y) = ∞ pour y ≠ x0
T=E-{x0}
y=x0
Tant que T ≠ ∅
  | Pour tout sommet z ∈ S(y)∩T
  |   F0(z)= min(F0(z), F0(y) + l(y,z))
  | Fin pour
  | y est le sommet de T tq F0(y)= min(F0(z); z∈ T)
  | T=T-{y}
Fin tant que
```

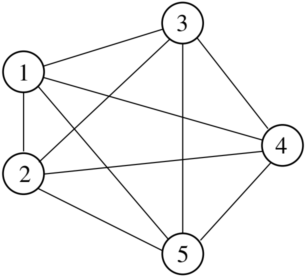
IV. Problème du voyageur de commerce

Un voyageur de commerce doit passer une et une seule fois par n villes et revenir à son point de départ en ayant parcouru la distance la plus petite possible.

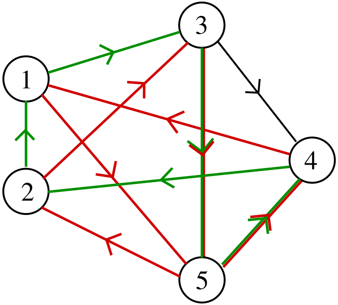
Un voyageur de commerce doit passer une et une seule fois par n villes et revenir à son point de départ en ayant parcouru la distance la plus petite possible.

Modélisation par un graphe.

- Graphe $G = (E, \Gamma)$ **complet** : $\forall (x, y) \in E \times E,\ x \neq y$, on a $(x, y) \in \Gamma$



- Chemin **hamiltonien** : chemin dans un graphe qui passe une et une seule fois par tous les sommets



(1,3,5,4,2,1) circuit hamiltonien

(1,5,2,3,5,4,1) circuit non–hamiltonien

Problème du voyageur de commerce

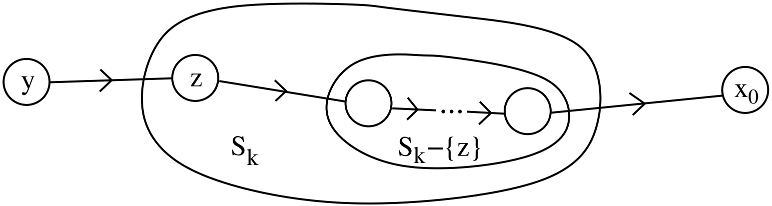
Le problème du voyageur de commerce consiste à chercher un *circuit hamiltonien* de valeur *minimale* dans un graphe valué *complet* à n sommets.

Remarque. Il y a $(n-1)!$ circuits hamiltoniens possibles.

Formule de récurrence pour le voyageur de commerce

Soit x_0 fixé. On cherche le plus court chemin de y quelconque à x_0 en passant une et une seule fois par les k sommets d'un sous-ensemble (S_k).

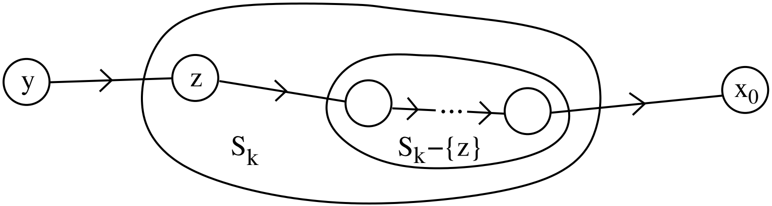
On note $C_k(y, S_k)$ la valeur *minimale* de ce chemin.



Formule de récurrence pour le voyageur de commerce

Soit x_0 fixé. On cherche le plus court chemin de y quelconque à x_0 en passant une et une seule fois par les k sommets d'un sous-ensemble (S_k).

On note $C_k(y, S_k)$ la valeur *minimale* de ce chemin.



Formule de récurrence

- $C_0(y, \emptyset) = L(y, x_0)$ pour tout $y \neq x_0$
- Pour $k = 1, \cdots, n-1$,

  $$C_k(y, S_k) = \min_{z \in S_k} \left(C_{k-1}(z, S_k - \{z\}) + L(y, z) \right) \text{ pour } y, x_0 \notin S_k$$

  avec $y \neq x_0$ sauf pour la dernière étape.

- **Etape** 0

  $C_0(y, \emptyset) = L(y, x_0)$ pour tout $y \neq x_0$

- **Etape** 0

  $C_0(y, \emptyset) = L(y, x_0)$ pour tout $y \neq x_0$

- **Etape** $k = 1, \cdots, n-2$

  $$C_k(y, S_k) = \min_{z \in S_k} \left(C_{k-1}(z, S_k - \{z\}) + L(y, z) \right)$$

  pour tout $y \neq x_0$, pour tout sous-ensemble S_k de k sommets tel que $y, x_0 \notin S_k$.

- **Etape** 0

  $C_0(y, \emptyset) = L(y, x_0)$ pour tout $y \neq x_0$

- **Etape** $k = 1, \cdots, n-2$

  $$C_k(y, S_k) = \min_{z \in S_k} \left(C_{k-1}(z, S_k - \{z\}) + L(y, z) \right)$$

  pour tout $y \neq x_0$, pour tout sous-ensemble S_k de k sommets tel que $y, x_0 \notin S_k$.

- **Etape** $n-1$

  $$C_{n-1}(x_0, \{x_1, \cdots, x_{n-1}\})$$
  $$= \min_{z \in \{x_1, \cdots, x_{n-1}\}} \left(C_{n-2}(z, \{x_1, \cdots, x_{n-1}\} - \{z\}) + L(x_0, z) \right)$$

  $=$ longueur minimale du circuit hamiltonien

- **Etape** 0

  $C_0(y, \emptyset) = L(y, x_0)$ pour tout $y \neq x_0$

- **Etape** $k = 1, \cdots, n-2$

  $$C_k(y, S_k) = \min_{z \in S_k} \left(C_{k-1}(z, S_k - \{z\}) + L(y, z) \right)$$

  pour tout $y \neq x_0$, pour tout sous-ensemble S_k de k sommets tel que $y, x_0 \notin S_k$.

- **Etape** $n-1$

  $$C_{n-1}(x_0, \{x_1, \cdots, x_{n-1}\})$$
  $$= \min_{z \in \{x_1, \cdots, x_{n-1}\}} \left(C_{n-2}(z, \{x_1, \cdots, x_{n-1}\} - \{z\}) + L(x_0, z) \right)$$

  $=$ longueur minimale du circuit hamiltonien

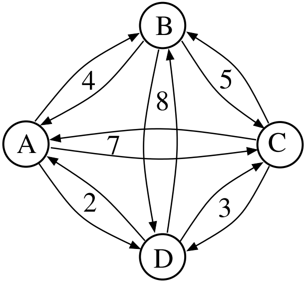
Remontée des calculs. On sélectionne les sommets z où les minima sont atteints :

$$D_k(y, S_k) = \{z \in S_k \text{ qui minimise } C_{k-1}(z, S_k - \{z\}) + L(y, z)\}$$

Exemple.

Tableau des distances entre villes

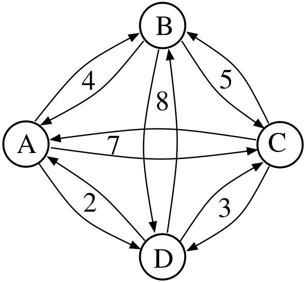
	A	B	C	D
A	0	4	7	2
B	4	0	5	8
C	7	5	0	3
D	2	8	3	0



Exemple.

Tableau des distances entre villes

	A	B	C	D
A	0	4	7	2
B	4	0	5	8
C	7	5	0	3
D	2	8	3	0



On prend $x_0 = A$

Etape 0 : $C_0(B, \emptyset) = L(B, A) = 4$
$C_0(C, \emptyset) = L(C, A) = 7$
$C_0(D, \emptyset) = L(D, A) = 2$

Etape 1 : $C_1(y, S_1) = \min_{z \in S_1} \left(C_0(z, \emptyset) + L(y, z) \right), \quad \forall y \neq A$

$\forall S_1 \in \{\{B\}, \{C\}, \{D\}\}$ tel que $y \notin S_1$.

$$
\begin{array}{ll}
C_1(B, \{C\}) = 7 + 5 = 12; & D_1(B, \{C\}) = C \\
C_1(B, \{D\}) = 2 + 8 = 10; & D_1(B, \{D\}) = D \\
C_1(C, \{B\}) = 4 + 5 = 9; & D_1(C, \{B\}) = B \\
C_1(C, \{D\}) = 2 + 3 = 5; & D_1(C, \{D\}) = D \\
C_1(D, \{B\}) = 4 + 8 = 12; & D_1(D, \{B\}) = B \\
C_1(D, \{C\}) = 7 + 3 = 10; & D_1(D, \{C\}) = C
\end{array}
$$

Etape 2 : $C_2(y, S_2) = \min_{z \in S_2} \left(C_1(z, S_2 - \{z\}) + L(y, z) \right), \quad \forall y \neq A$

$\forall S_2 \in \{\{B, C\}, \{B, D\}, \{C, D\}\}$ tel que $y \notin S_2$.

$C_2(B, \{C, D\}) = \min(5 + 5, 10 + 8) = 10;$ $D_2(B, \{C, D\}) = C$

$C_2(C, \{B, D\}) = \min(10 + 5, 12 + 3) = 15;$ $D_2(C, \{B, D\}) = \{B, D\}$

$C_2(D, \{B, C\}) = \min(12 + 8, 9 + 3) = 12;$ $D_2(D, \{B, C\}) = C$

Etape 2 : $C_2(y, S_2) = \min_{z \in S_2} \left(C_1(z, S_2 - \{z\}) + L(y, z) \right), \quad \forall y \neq A$

$\forall S_2 \in \{\{B, C\}, \{B, D\}, \{C, D\}\}$ tel que $y \notin S_2$.

$C_2(B, \{C, D\}) = \min(5 + 5, 10 + 8) = 10;$ $\quad D_2(B, \{C, D\}) = C$
$C_2(C, \{B, D\}) = \min(10 + 5, 12 + 3) = 15;$ $\quad D_2(C, \{B, D\}) = \{B, D\}$
$C_2(D, \{B, C\}) = \min(12 + 8, 9 + 3) = 12;$ $\quad D_2(D, \{B, C\}) = C$

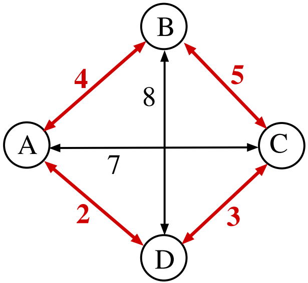
Etape 3 : $C_3(A, S_3) = \min_{z \in S_3} \left(C_2(z, S_3 - \{z\}) + L(A, z) \right)$

avec $S_3 = \{B, C, D\}$

$$\begin{aligned} C_3(A, \{B, C, D\}) &= \min\big(C_2(B, \{C, D\}) + L(A, B), \\ &\qquad\quad C_2(C, \{B, D\}) + L(A, C), \\ &\qquad\quad C_2(D, \{B, C\}) + L(A, D)\big) \\ &= \min(10 + 4, 15 + 7, 12 + 2) = 14 \end{aligned}$$

$$D_3(A, \{B, C, D\}) = \{B, D\}$$

On a obtenu 2 circuits hamiltoniens de longueur minimale 14 :
(A, B, C, D, A) et (A, D, C, B, A).



Remarque : La matrice des longueurs est **symétrique** $\Rightarrow$ on doit toujours trouver un nombre **paire** de solutions car on peut parcourir le circuit dans les deux sens.