Voyageur de Commerce

Algorithme exact avec programmation dynamique

Nicolas Hanusse

Directeur de Recherche CNRS, LaBRI, Université de Bordeaux

Voyageur de Commerce

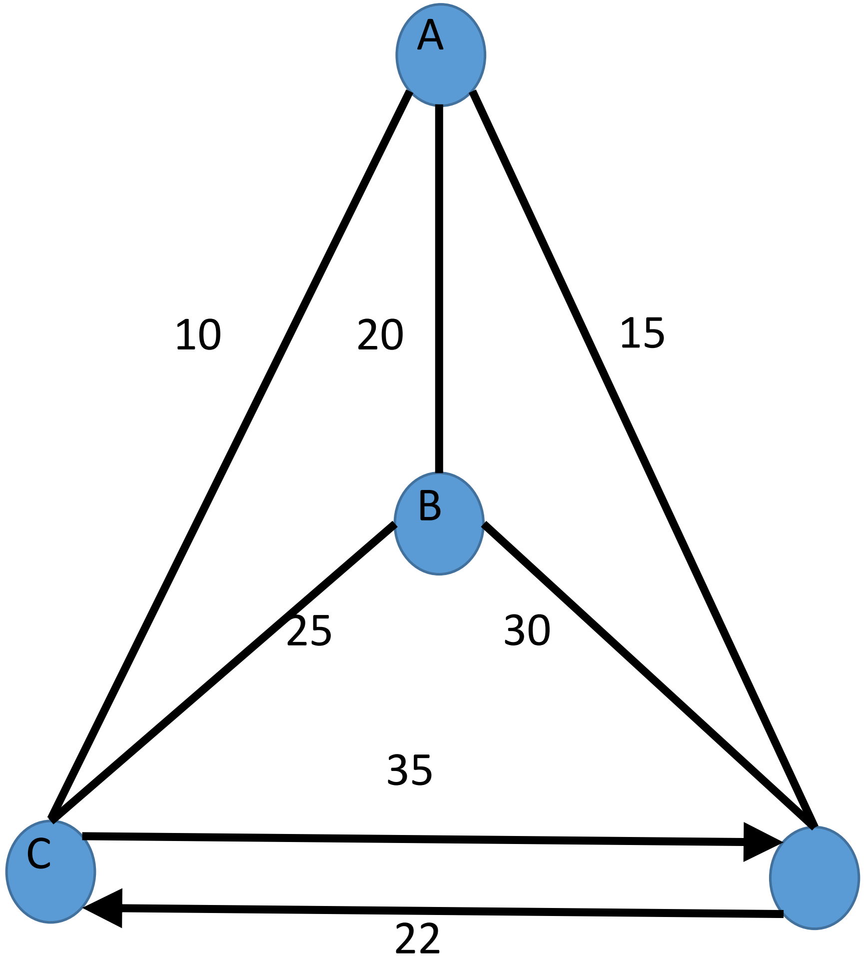
Instance: Un ensemble V de points/sommets et une distance d sur V.

Question: Trouver une *tournée de longueur minimum* passant par tous les points de V, c'est-à-dire un ordre $v_0,\ldots,v_{n-1}$ des points de V tel que $\sum d(v_i,v_{i+1})$ est minimum.

 **plusieurs variantes**:

- Asymétrique/symétrique: d(A,B) = d(B,A) ?
- Métrique: inégalité triangulaire vérifiée d(A,B)+d(B,C) <= d (A,C)

Exemple



Prenons comme source **s=A**

ABCDA a poids 20+25+35+15=95

Peut-on faire avec un poids plus faible ?

Exemple

Voir http://map.vroom-project.org/

Principaux algorithmes

- Exaustif:
  - $(n-1)!$ chemins
  - Meilleur algorithme exact: $O(n^2\,2^n)$ via *Programmation Dynamique*
- Approximation:
  - 2-approximation via *arbres de poids minimal ou 2-opt, …* $O(n^{O(1)})$
  - 3/2-approximation via *algorithme de Christofides* $O(n^3)$

Soit OPT la valeur optimale de la mesure coût(S) d'une solution S d'un algorithme.
Un algorithme d'approximation de facteur f s'il calcule une solution S telle que :

- OPT ≤ Coût(S) ≤f OPT pour les *problèmes de minimisation*
- OPT/f ≤ Coût(S) ≤ OPT pour les *problèmes de maximisation*

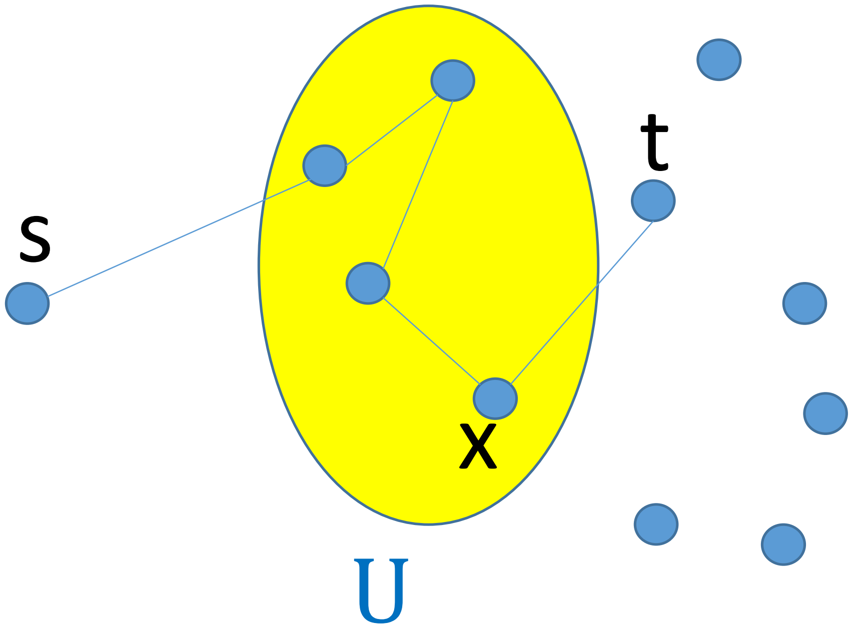
Programmation Dynamique

- Principe « Diviser pour régner »:
  - On suppose calculé les meilleurs résultats pour un problème de taille < k.
  - On mémorise ces résultats (souvent dans des tables)
  - Calculer le meilleur résultat pour la taille k en fonction des résultats <k en avec une opération de combinaison.
  - On itère jusqu'à obtenir la taille désirée n.
- Souvent:
  - Usage important de la mémoire
  - Calcul de combinaison « rapide »

Propriété pour le voyageur de commerce

- On suppose calculé les k-1 chemins minimaux **s** → …→ **x** passant par k-1 sommets fixés U
- Pour **t** donné, on mémorise *la meilleure combinaison*, le chemin de poids minimal de forme **s** → …→ **x** → **t**
- Combinaison = k-1 choix pour x

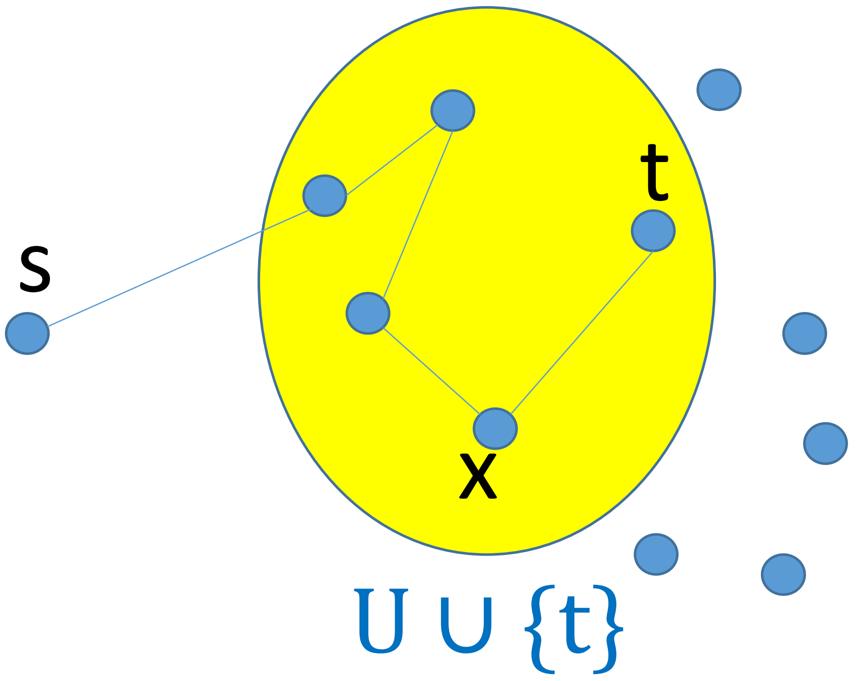
  Min (poids(**s** → …→ **x**) + d(**x**,**t**))



Propriété pour le voyageur de commerce

- On suppose calculé les k-1 chemins minimaux **s** → …→ **x** passant par k-1 sommets fixés U
- Pour **t** donné, on mémorise *la meilleure combinaison*, le chemin de poids minimal de forme **s** → …→ **x** → **t**
- Combinaison = k-1 choix pour x

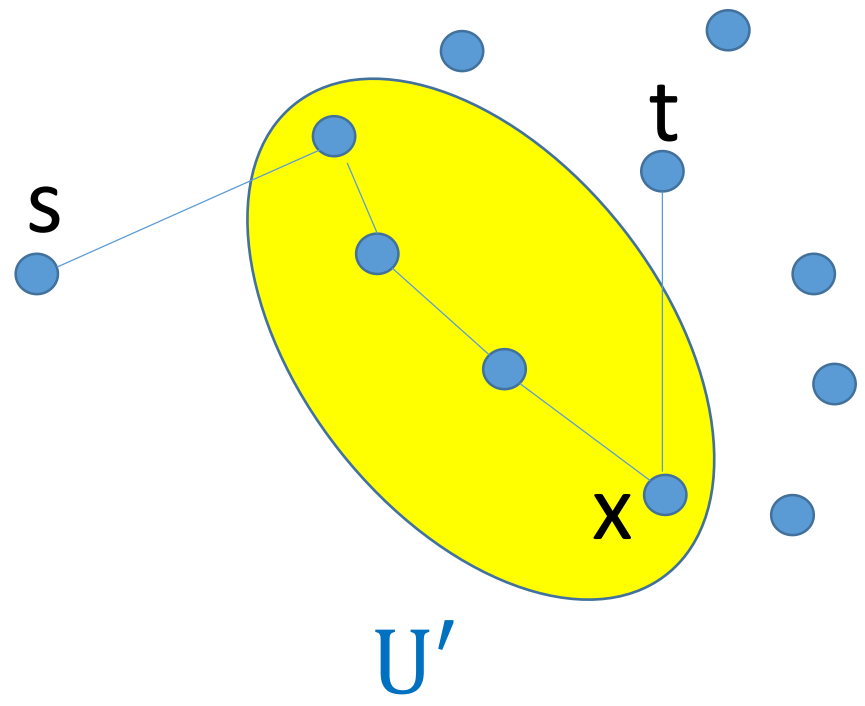
  Min (poids(**s** → …→ **x** → **t**) + d(**x**,**t**))



Propriété pour le voyageur de commerce

- On suppose calculé les k-1 chemins minimaux **s** → …→ **x** passant par k-1 sommets fixés U
- Pour **t** donné, on mémorise *la meilleure combinaison*, le chemin de poids minimal de forme **s** → …→ **x** → **t**
- Combinaison = k-1 choix pour x

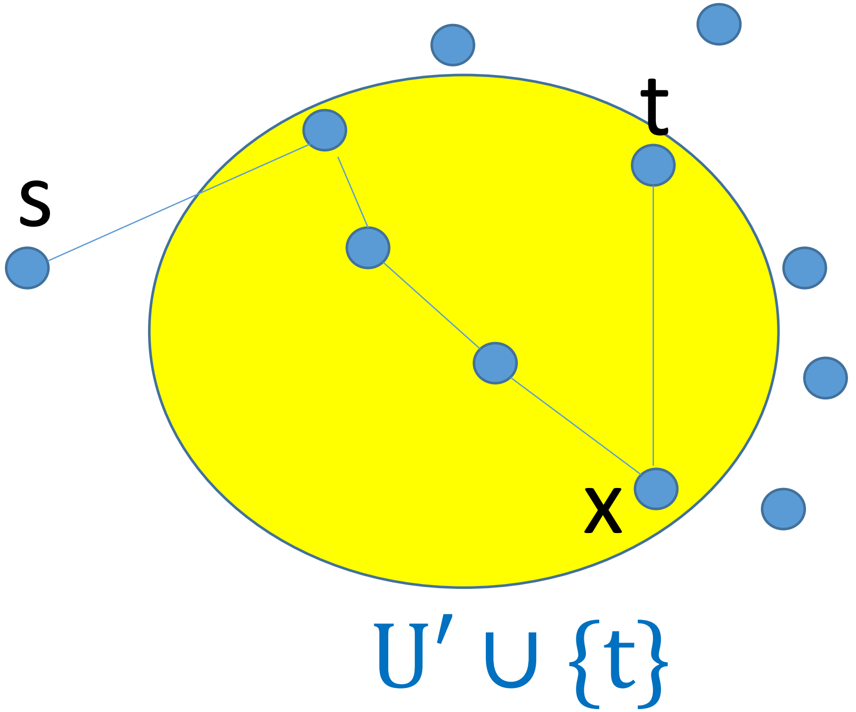
  Min (poids(**s** → …→ **x** → **t**) + d(**x**,**t**))
- On itère sur tous les ensembles U de taille k-1



Propriété pour le voyageur de commerce

- On suppose calculé les k-1 chemins minimaux **s** → …→ **x** passant par k-1 sommets fixés U
- Pour **t** donné, on mémorise *la meilleure combinaison*, le chemin de poids minimal de forme **s** → …→ **x** → **t**
- Combinaison = k-1 choix pour x

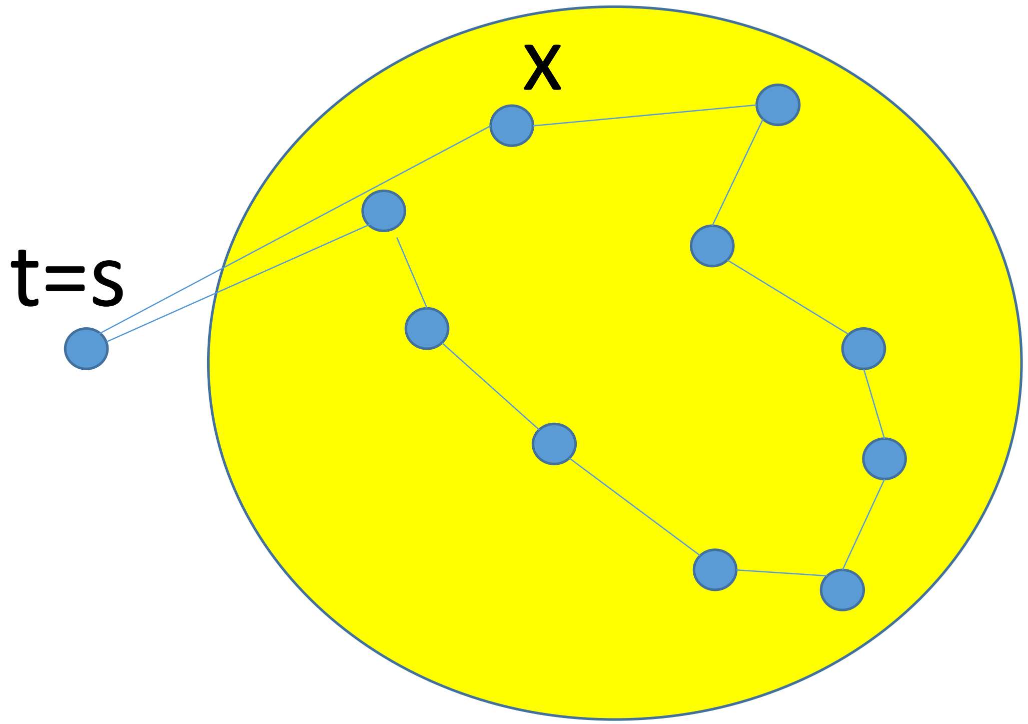
  Min (poids(**s** → …→ **x** → **t**) + d(**x**,**t**))
- On itère sur tous les ensembles U de taille k-1



Propriété pour le voyageur de commerce

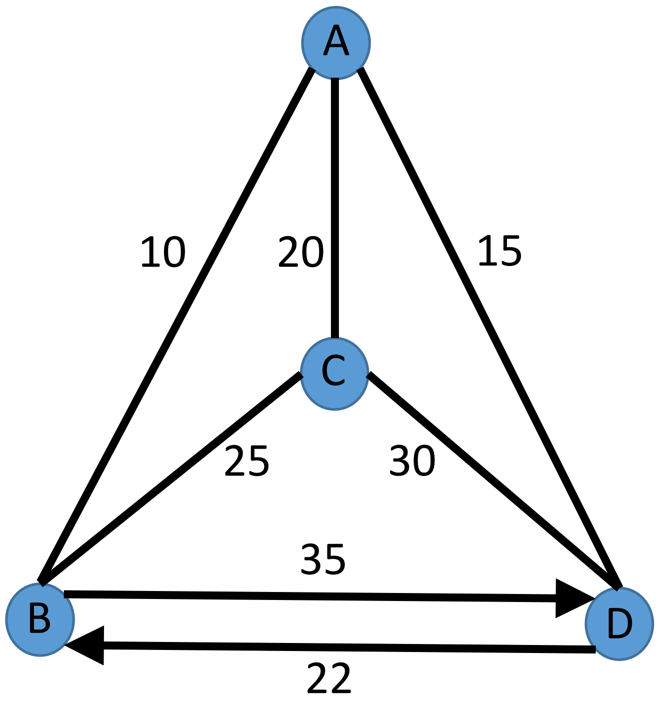
- On suppose calculé les k-1 chemins minimaux **s** → …→ **x** passant par k-1 sommets fixés U
- Pour **t** donné, on mémorise *la meilleure combinaison*, le chemin de poids minimal de forme **s** → …→ **x** → **t**
- Combinaison = k-1 choix pour x

  Min (poids(**s** → …→ **x** → **t**) + d(**x**,**t**))
- On itère sur tous les ensembles U de taille k-1 puis sur toutes les tailles.



$|U| = n - 1$

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U - \{t\}} L(x, U - \{t\}) + d(x,t)$$

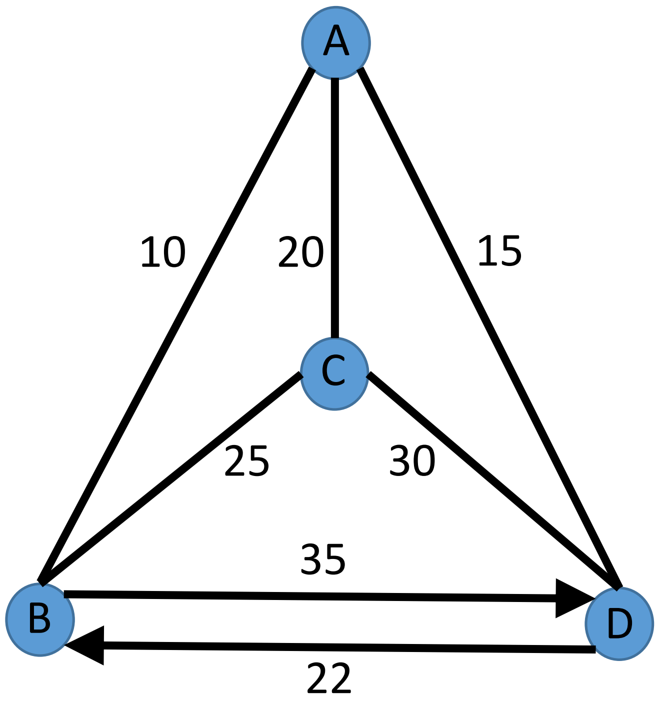


$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$

$$OPT = \mathrm{L}(s, V)$$

t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.				
C	indef	20 AC	Ind.				
D	indef	ind	15 AD				

$$L(t, U) = d(s, t) \quad \text{si } |U| = 1$$
$$L(t, U) = \min_{x \in U - \{t\}} L(x, U - \{t\}) + d(x, t)$$



$L(t, U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$

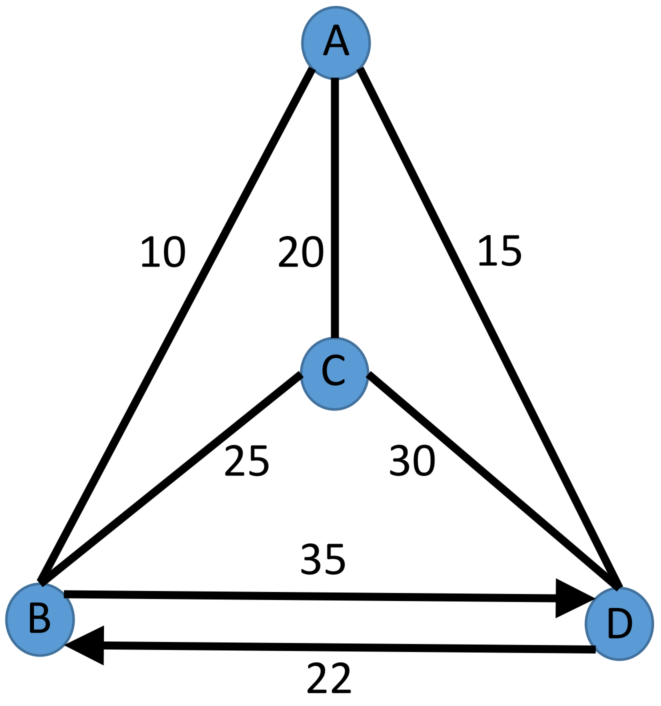
$L(B, \{B, C\})$ = poids(A→C) + d(C,B) = 20+25 =45

t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB			
C	indef	20 AC	Ind.				
D	indef	ind	15 AD				

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U-\{t\}} L(x, U-\{t\}) + d(x,t)$$

$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$

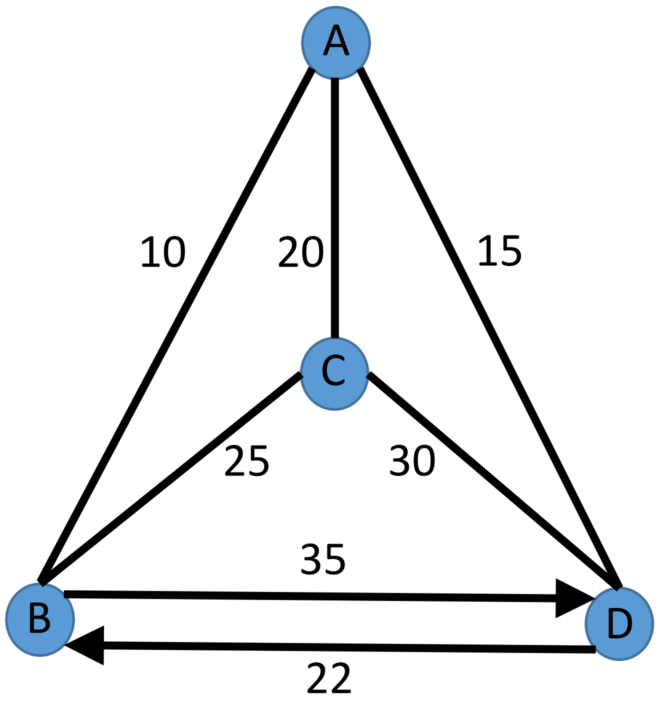
$L(D,\{C,D\})$ = poids(A→C) + d(C,D) = 20+30 =50



t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	
D	indef	ind	15 AD	Ind.	45 ABD	$\boxed{\text{50 ACD}}$	

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U-\{t\}} L(x, U-\{t\}) + d(x,t)$$

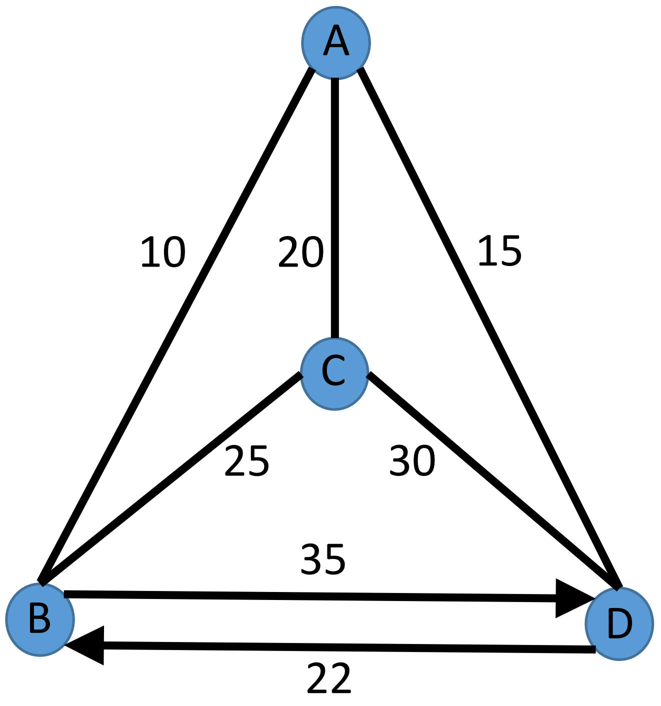
$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$



t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	Min(L(C,{CD})+d(C,B), L(D,{CD}+d(D,B))=70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U-\{t\}} L(x, U-\{t\}) + d(x,t)$$

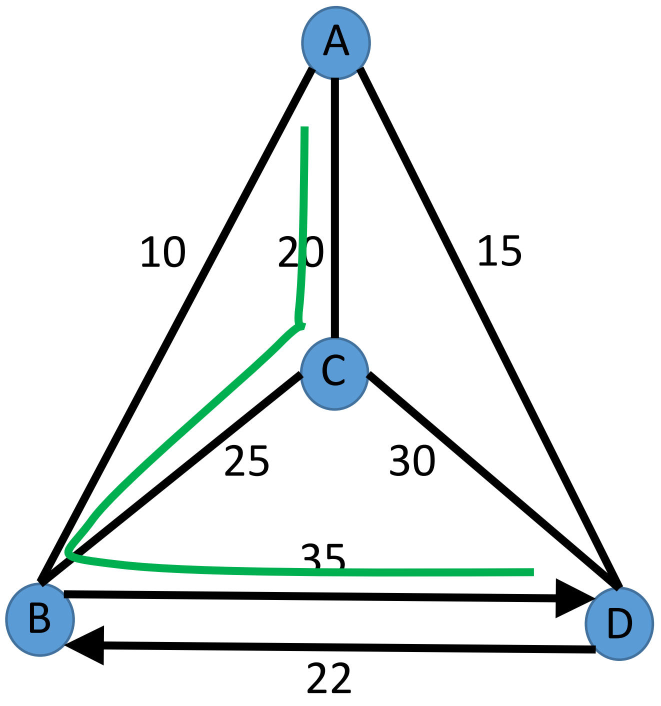
$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$



t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	Min(L(C,{CD})+d(C,B), L(D,{CD}+d(D,B))=70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	Min(37+25,45+22)=62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U-\{t\}} L(x, U - \{t\}) + d(x,t)$$

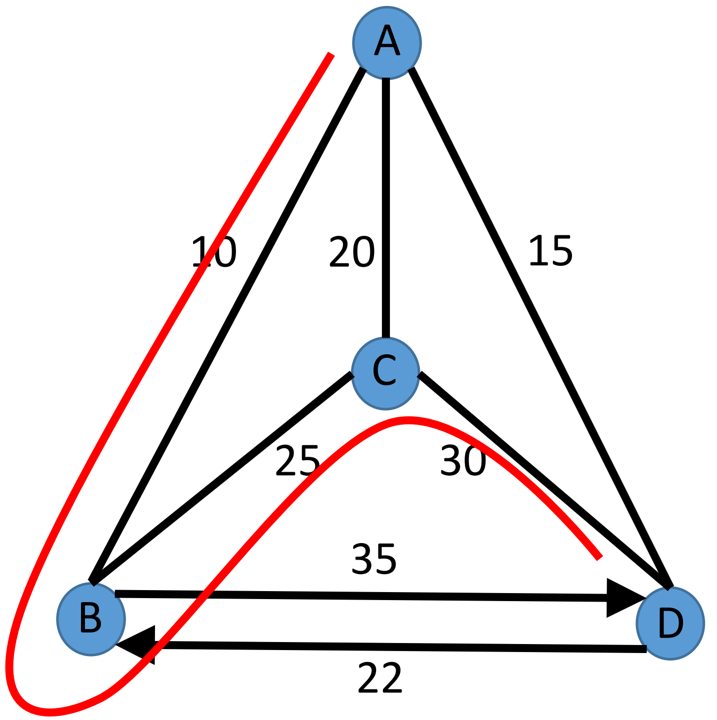
$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$



t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	Min(L(C,{CD})+d(C,B), L(D,{CD}+d(D,B))=70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	Min(37+25,45+22)=62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	Min(45+35,

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U-\{t\}} L(x, U-\{t\}) + d(x,t)$$

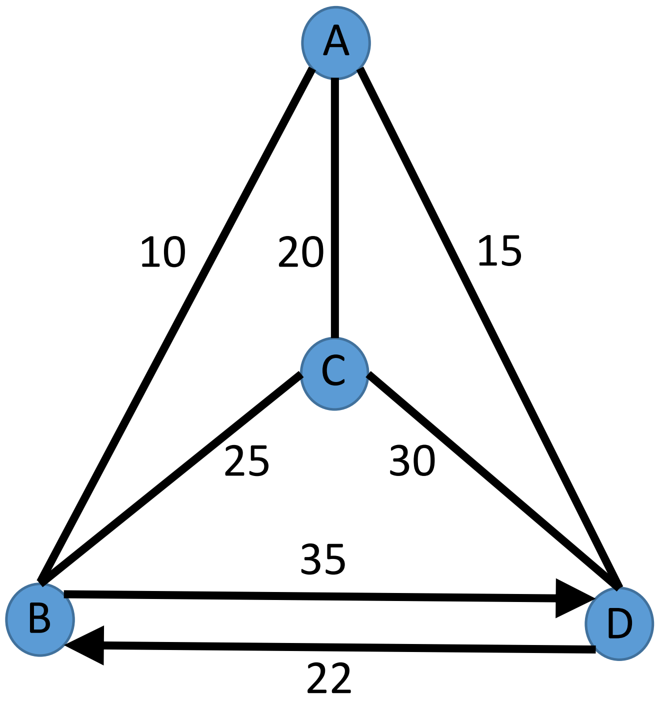
$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$



t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	Min(L(C,{CD})+d(C,B), L(D,{CD}+d(D,B))=70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	Min(37+25,45+22)=62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	Min(45+35,35+30)=65 ABCD

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U-\{t\}} L(x, U-\{t\}) + d(x,t)$$

$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$

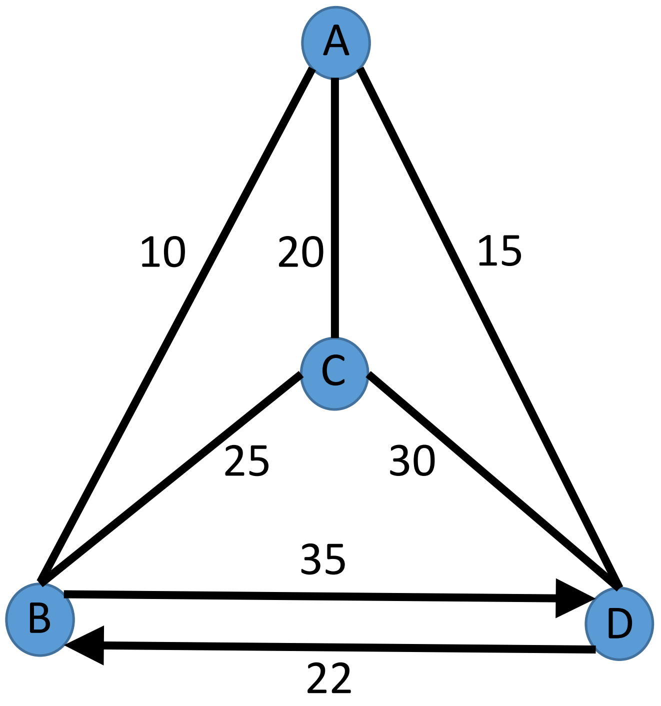


t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	Min(L(C,{CD})+d(C,B), L(D,{CD}+d(D,B))=70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	Min(37+25,45+22)=62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	Min(45+35,35+30)=65 ABCD

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U - \{t\}} L(x, U - \{t\}) + d(x,t)$$

$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$

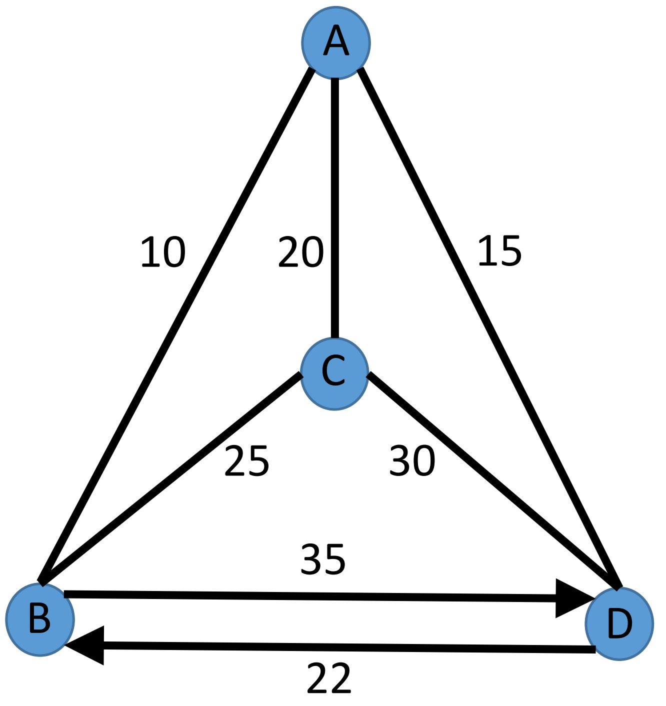
$OPT = \mathrm{L}(\mathrm{s}, V)$



t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	65 ABCD

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U - \{t\}} L(x, U - \{t\}) + d(x,t)$$

$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \dots, vk, t\}$

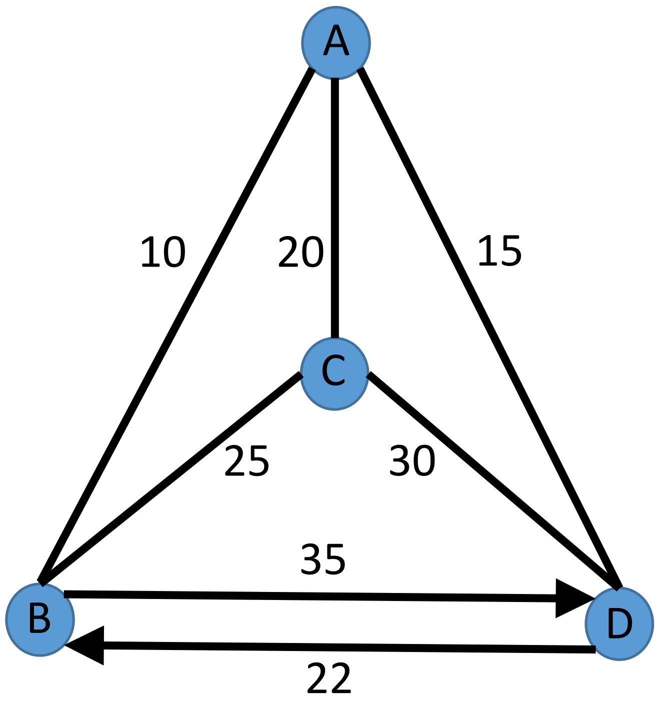


$OPT = \mathrm{L}(s, V)$ = L(A, {A, B, C, D}) = Min(**70+10**, 62+20, **65+15**) = 80
ADCBA ou ABCDA

t \ U	{B}	{C}	{D}	{B,C}	{B,D}	{C,D}	{B,C,D}
B	L(B,{B}) = d(A,B)=10	ind	Ind.	45 ACB	37 ADB	Ind.	70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	65 ABCD

$$L(t,U) = d(s,t) \quad \text{si } |U| = 1$$
$$L(t,U) = \min_{x \in U-\{t\}} L(x, U-\{t\}) + d(x,t)$$

$L(t,U)$ = longueur du plus court chemin s → …→ x → t passant par tous les sommets de $U = \{v1, v2, \ldots, vk, t\}$



$OPT = L(s,V)$ = $L(A, \{A, B, C, D\})$ = Min(**70+10**, 62+20, **65+15**) = 80
ADCBA ou ABCDA

t \ U	{B} 0100	{C} 0010	{D} 0001	{B,C} 0110	{B,D} 0101	{C,D} 0011	{B,C,D} 0111
B	L(B,{B}) = d(A,B)=10	Ind	Ind.	45 ACB	37 ADB	Ind.	70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	65 ABCD

Détail d'implémentation

- Bien coder les sous-ensembles: **bitmap**
- **Facilement générer le prochain sous-ensemble**: pour traiter un ensemble U, il faut avoir déjà traité tous ses sous-ensembles.

t \ U	{B} 0100	{C} 0010	{D} 0001	{B,C} 0110	{B,D} 0101	{C,D} 0011	{B,C,D} 0111
B	L(B,{B}) = d(A,B)=10	Ind	Ind.	45 ACB	37 ADB	Ind.	70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	65 ABCD

Analyse de complexité

- Il y a 2^n sous-ensembles de cardinalité ≤ n mais il faut exclure {} et V → $2^n - 2$ **colonnes**;
- On a n-1 **lignes** donc (n-1)*$(2^n - 2)$ **entrées**
- Une combinaison est un minimum sur au plus n-1 valeurs → $O(n^2\, 2^n)$ calculs

t \ U	{B} 0100	{C} 0010	{D} 0001	{B,C} 0110	{B,D} 0101	{C,D} 0011	{B,C,D} 0111
B	L(B,{B}) = d(A,B)=10	Ind	Ind.	45 ACB	37 ADB	Ind.	70 ADCB
C	indef	20 AC	Ind.	35 ABC	Ind.	45 ADC	62 ADBC
D	indef	ind	15 AD	Ind.	45 ABD	50 ACD	65 ABCD