Introduction au TP2

D. Guastella, V. David

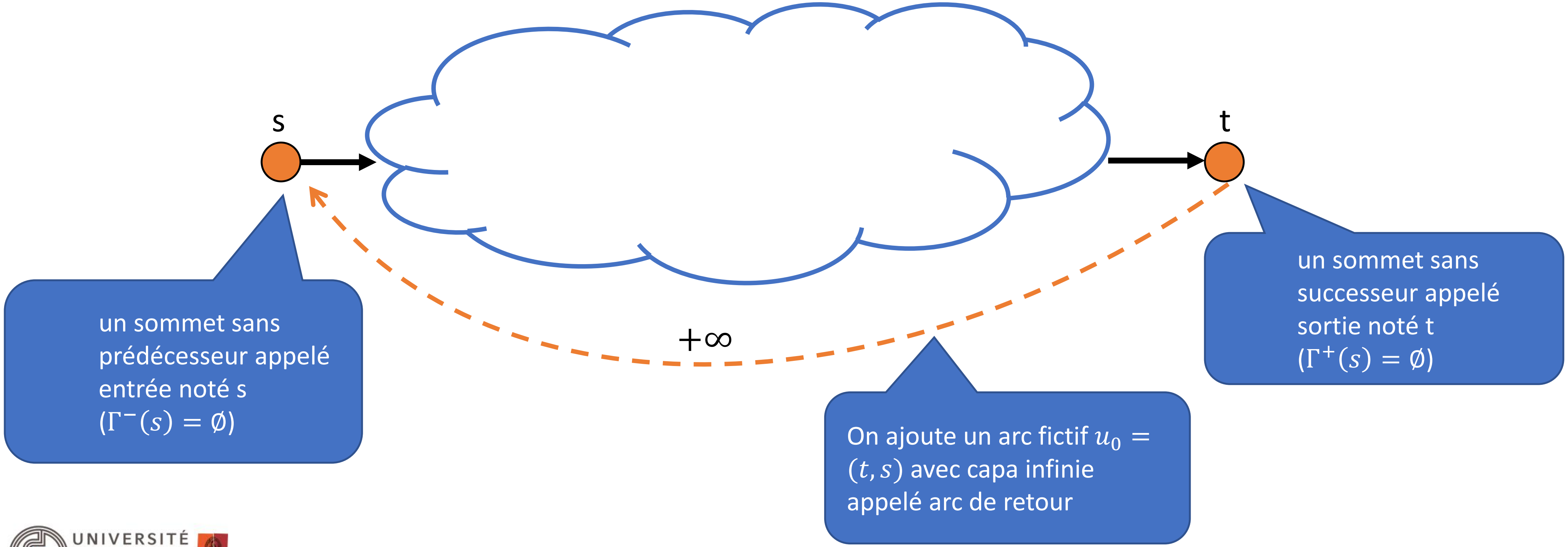
Flots

Flots : Définitions

- Un réseau de transport est un graphe orienté connexe
- $R = (X, U = \{u_1, \dots, u_m\})$ avec
  - un sommet sans prédécesseur appelé entrée noté s ($\Gamma^-(s) = \emptyset$)
  - un sommet sans successeur appelé sortie noté t ($\Gamma^+(s) = \emptyset$)
  - une application $capa = U \rightarrow \mathbb{N} \cup \{+\infty\}$ qui à chaque arc u associe sa capacité $capa(u) \geq 0$
- On ajoute un arc fictif $u_0 = (t, s)$ avec capa infinie appelé arc de retour

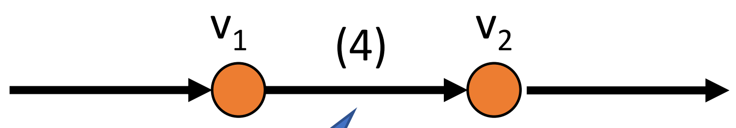
Exemple : Entrée - Sortie

- Modéliser un problème de flot maximal signifie dessiner un graphe de réseau de transport



Exemple : capacité d'un arc

- Modéliser un problème de flot maximal signifie dessiner un graphe de réseau de transport



une application $capa = U \to \mathbb{N} \cup \{+\infty\}$ qui à chaque arc u associe sa capacité $capa(u) \geq 0$

$$\boxed{capa(v_1, v_2) = 4}$$

Définition : valeur du flot

$$v(\varphi) = \quad v(u_0) = \quad \sum_{u \in \varpi^+(\{s\})} v(u) = \quad \sum_{u \in \varpi^-(\{t\})} v(u)$$

Flux sur arc de retour

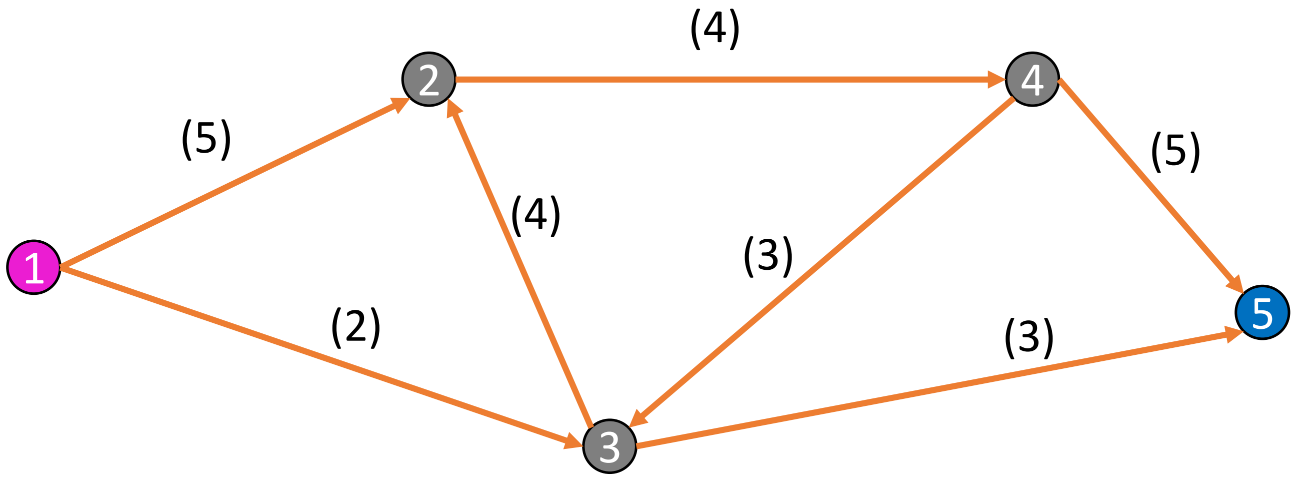
Somme des flux sortant de s

Somme des flux arrivant en t

- Flot de valeur max : maximise $v(\varphi)$ parmi tous les flots compatibles

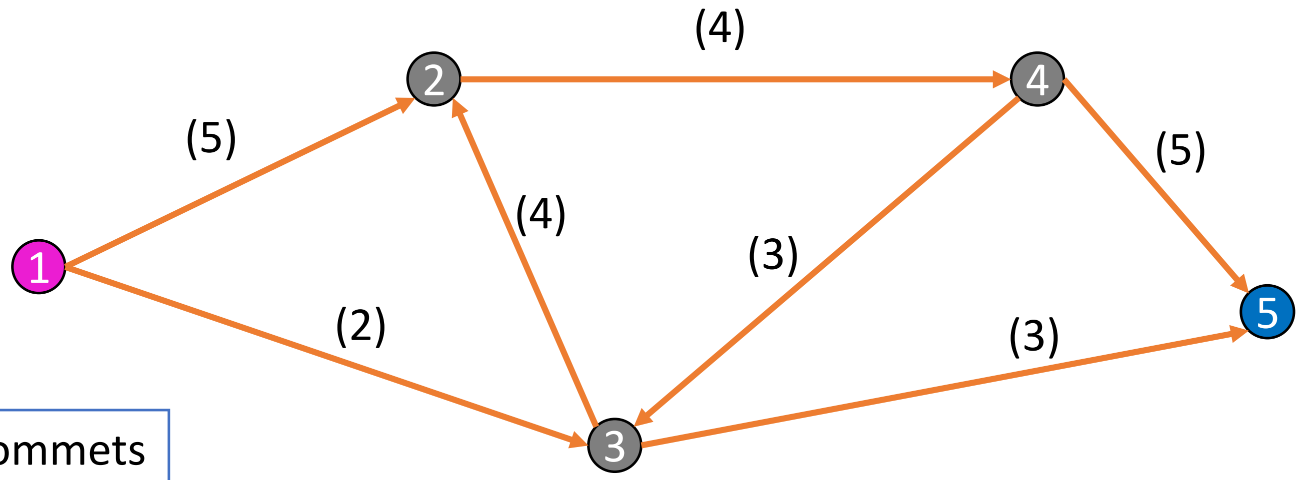
Dimacs

```
c hello world en dimacs!!
c voici un exemple d'encodage
p max 5 7
c sommet s
n 1 s
c sommet t
n 5 t
c maintenant la description des arcs
a 1 2 5
a 1 3 2
a 3 2 4
a 2 4 4
a 4 3 3
a 4 5 5
a 3 5 3
```



Dimacs

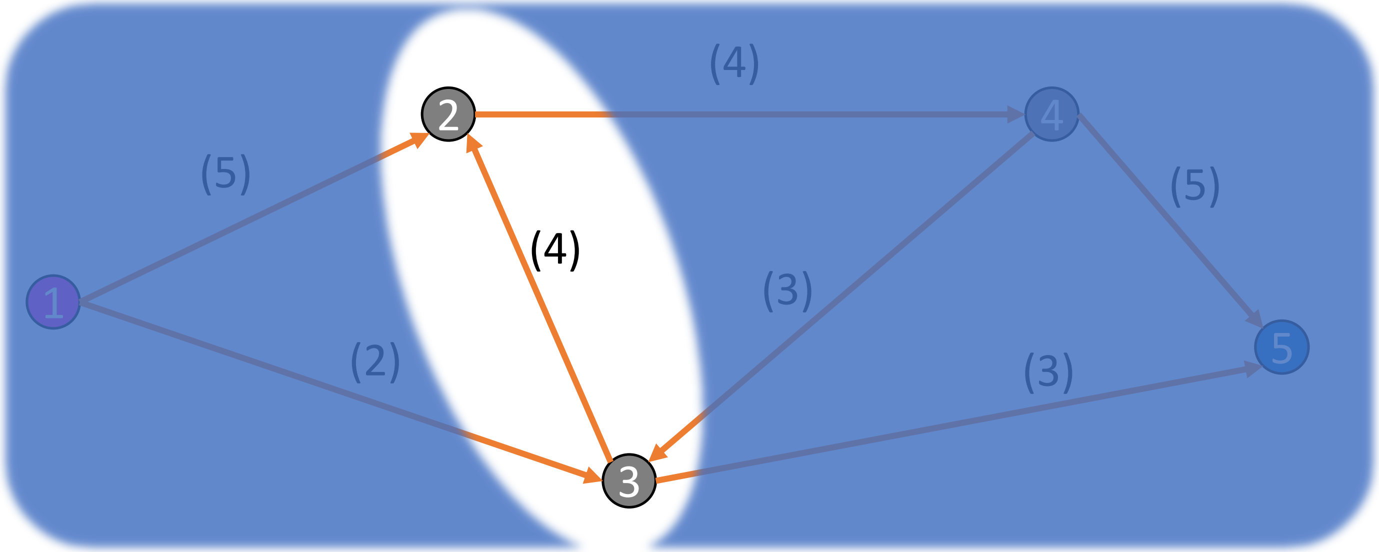
```
c hello world en dimacs!!
c voici un exemple d'encodage
p max 5 7
c sommet s
n 1 s
c sommet t
n 5 t
c maintenant la description des arcs
a 1 2 5
a 1 3 2
a 3 2 4
a 2 4 4
a 4 3 3
a 4 5 5
a 3 5 3
```



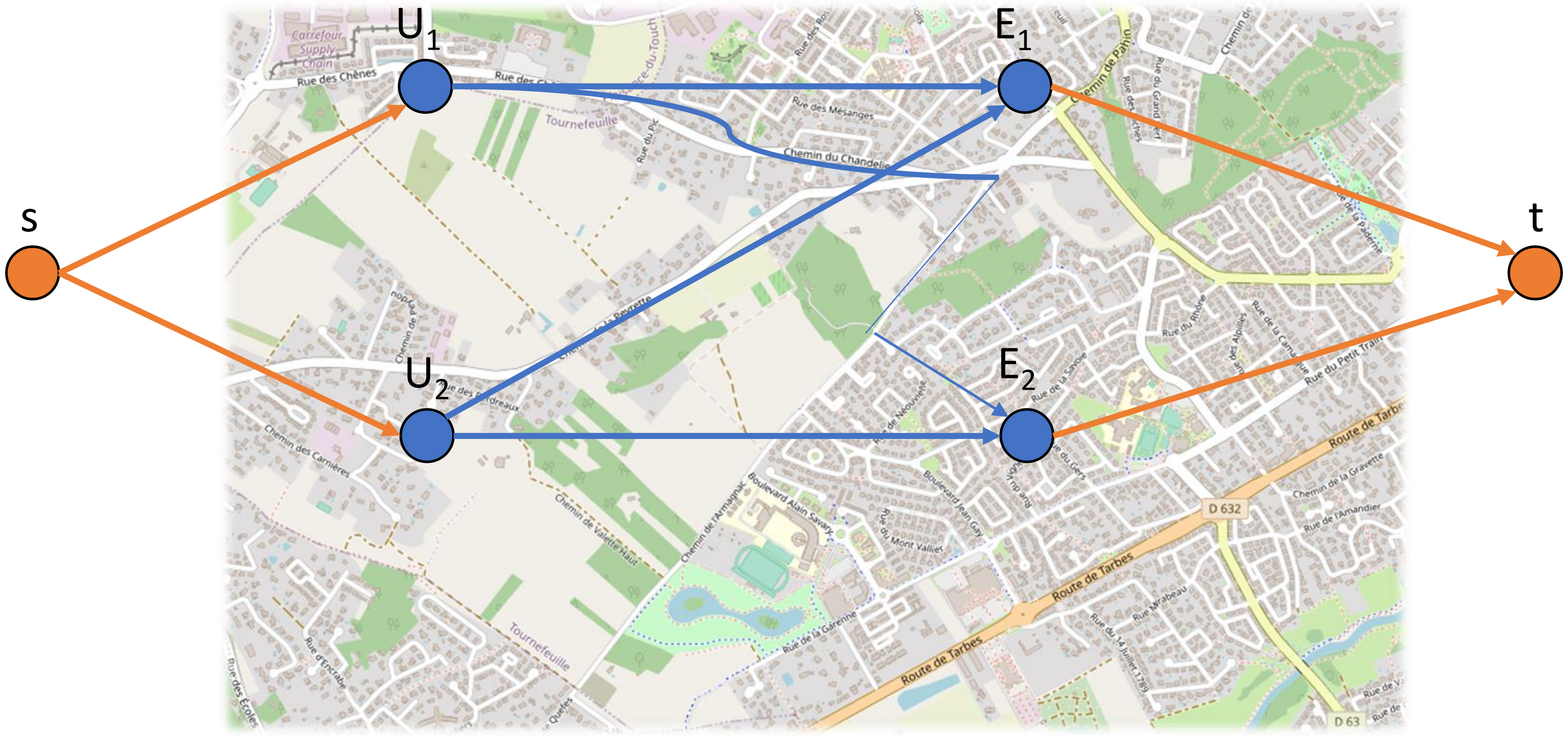
Attention: les indexes des sommets doivent commencer par 0

Dimacs

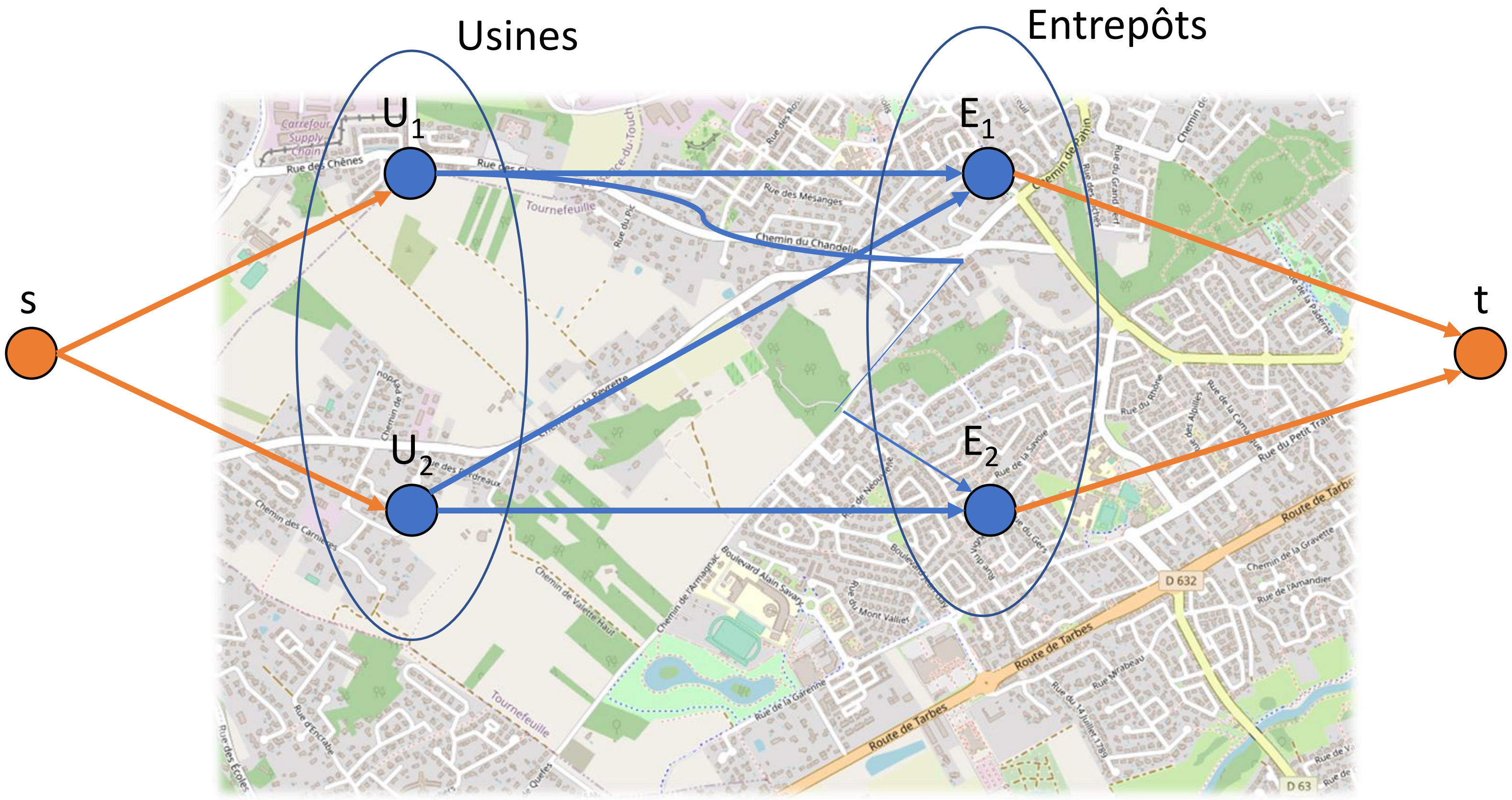
```
c hello world en dimacs!!
c voici un exemple d'encodage
p max 5 7
c sommet s
n 1 s
c sommet t
n 5 t
c maintenant la description des arcs
a 1 2 5
a 1 3 2
a 3 2 4
a 2 4 4
a 4 3 3
a 4 5 5
a 3 5 3
```



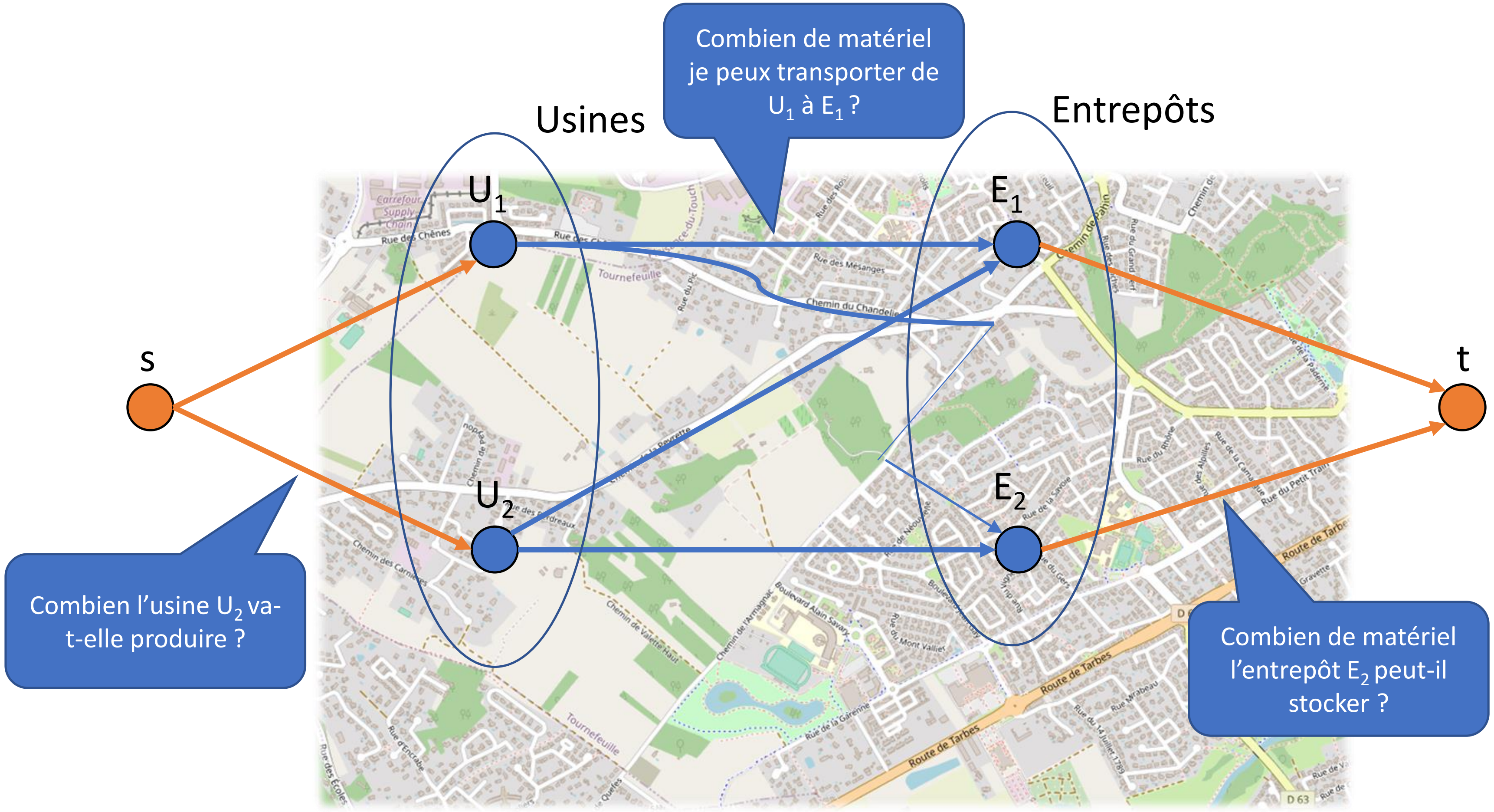
Flots



Flots



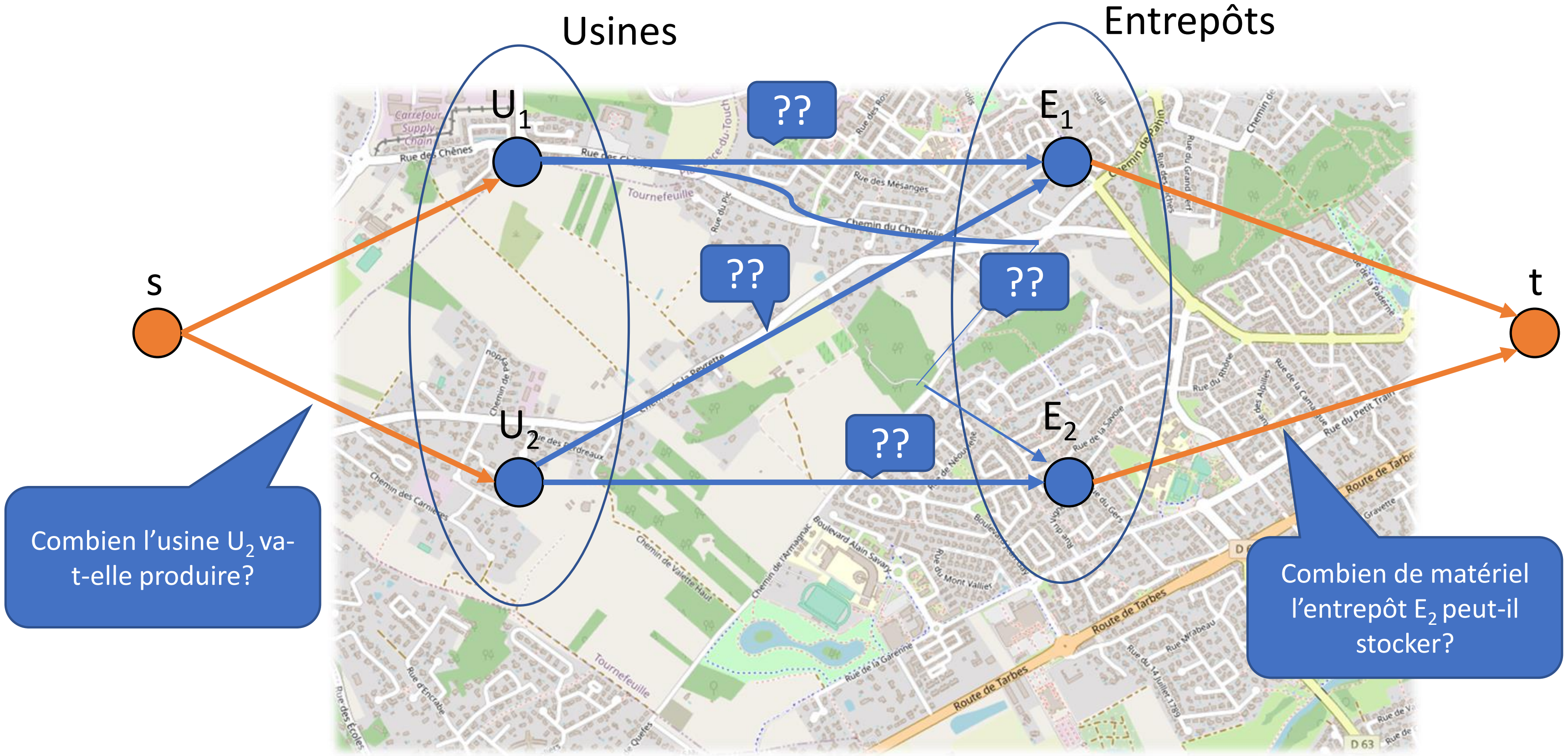
Flots


Usines
Entrepôts
Combien de matériel je peux transporter de U_1 à E_1 ?
U_1
U_2
E_1
E_2
s
t
Combien l'usine U_2 va-t-elle produire ?
Combien de matériel l'entrepôt E_2 peut-il stocker ?

Flots

Et si on imposait des contraintes sur un trajet entre les usines et les entrepôts ?

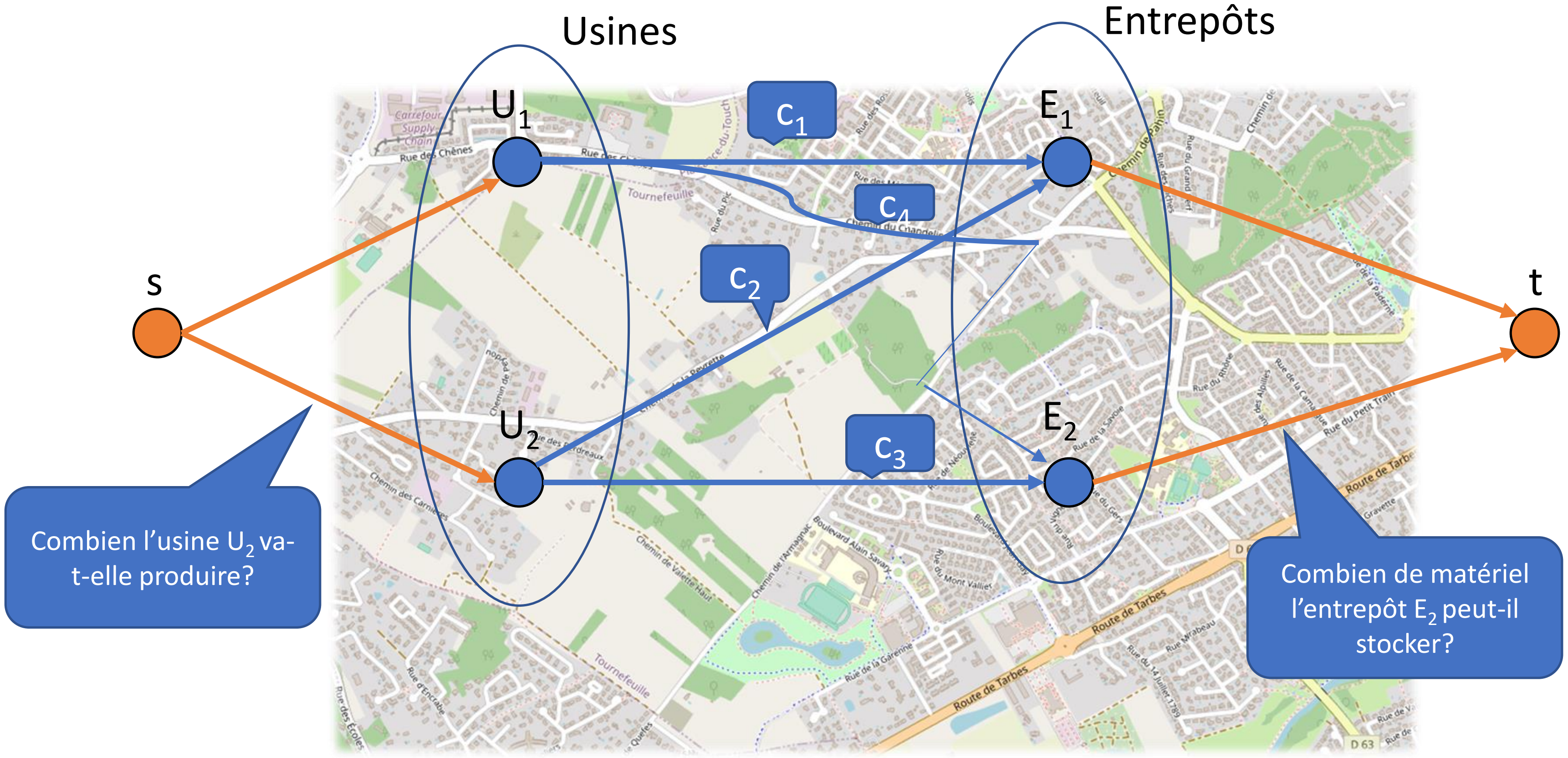
- Contraintes : certains camions ne peuvent pas prendre de chemins étroit, limitant la capacité de transport d'un trajet



Flots

Et si on imposait des contraintes sur un trajet entre les usines et les entrepôts ?

- Contraintes :
  certains camions ne peuvent pas prendre de chemins étroit, limitant la capacité de transport d'un trajet



Théorème : min-cut max-flow

Soit F l'ensemble des flots compatibles et K l'ensemble des coupes dans un réseau de transport :

$$\max_{\phi \in F} v(\phi) = \min_{cp \in K} capa(cp)$$

-> La valeur maximum du flot est égale à la capacité minimum d'une coupe

Programmation Linéaire

Programmation linéaire : Définitions

Un programme linéaire (en forme générale) est défini par:

- Des variables de décision : x_1, x_2, … x_n
- Un objectif z à optimiser: min/max $\quad z = \sum_{j=1}^{n} c_j x_j$
- *m* contraintes linéaires : $\quad \sum_{j=1}^{n} a_{ij} x_j \preccurlyeq b_i \qquad \forall i = 1,2,\ldots,m$
- *n* contraintes de positivité: $\quad x_j \preccurlyeq 0 \qquad \forall j \subseteq \{1,2,\ldots,n\}$

avec $\preccurlyeq \in \{\leq, \geq, =\}$

Exemple du barman:

Variables de décision :	x : nombre de bidons de "soleil couchant" (à 80€) y : nombre de bidons de "balancé" (à 60€)
Fonction Objectif :	$\max\ (z = 80x + 60y)$
CONTRAINTES	$5x + 3y \leq 30$ (stock jus d'orange) $2x + 3y \leq 24$ (stock jus pample.) $1x + 3y \leq 18$ (stock jus de framboise) $x, y \geq 0$ (contraintes de positivité)

Programmation linéaire

- Problème de transport (suite): maintenant nous considérons les coûts de transports entre les usines et les entrepôts

	E1	E2	E3
U1	4	3	6
U2	3	5	3

- Minimiser les coûts de transports signifie

$$\min \sum_{ij} C_{ij} t_{ij}$$

Variable identifiante le coût de transport entre U_i et E_j

Coût de transport entre U_i et E_j (ex. C_{13}=6)

Programmation linéaire

- Problème de transport (suite): maintenant nous considérons les coûts de transports entre les usines et les entrepôts

	E1	E2	E3
U1	4	3	6
U2	3	5	3

- Attention: on a des contraintes! Par exemple:

« l'entrepôt 1, noté E1, a une demande de 50 »

- *Ça peut se traduire comme « E1 a une demande d'au moins 50 »*
- On peut la modéliser comme $t_{11} + t_{21} \geq 50$

Programmation linéaire (liens sur le sujet)

- https://www.zweigmedia.com/simplex/simplex.php?lang=en

- Pour la visualisation:
  - https://www.zweigmedia.com/utilities/lpg/index.html?lang=en