**U.F.R SCIENCES ET TECHNIQUES**
*Département d'Informatique*
B.P. 1155
64013 PAU CEDEX
Téléphone secrétariat : 05.59.40.79.64
Télécopie : 05.59.40.76.54

# III- Problème de Flot Max /Cout Min

I- Modèle du problème de transport
II- Formulation du problème
III- Algorithme de Roy (Busacker et Gowen)

# I-Modèle du problème de transport

Soit **n** sites d'assemblage de smartphones :

$F_1$ , $F_2$ ,..., $F_n$

Chaque site $F_i$ fabrique **m** modèles $P_k$ de smartphones :

$P_1$ , $P_2$ ,..., $P_m$

Les produits doivent être acheminés vers **q** sites de consommation $C_j$ .

$C_1$ , $C_2$ ,..., $C_q$

Le problème:

On recherche le **réseau de flot** satisfaisant la **demande**.

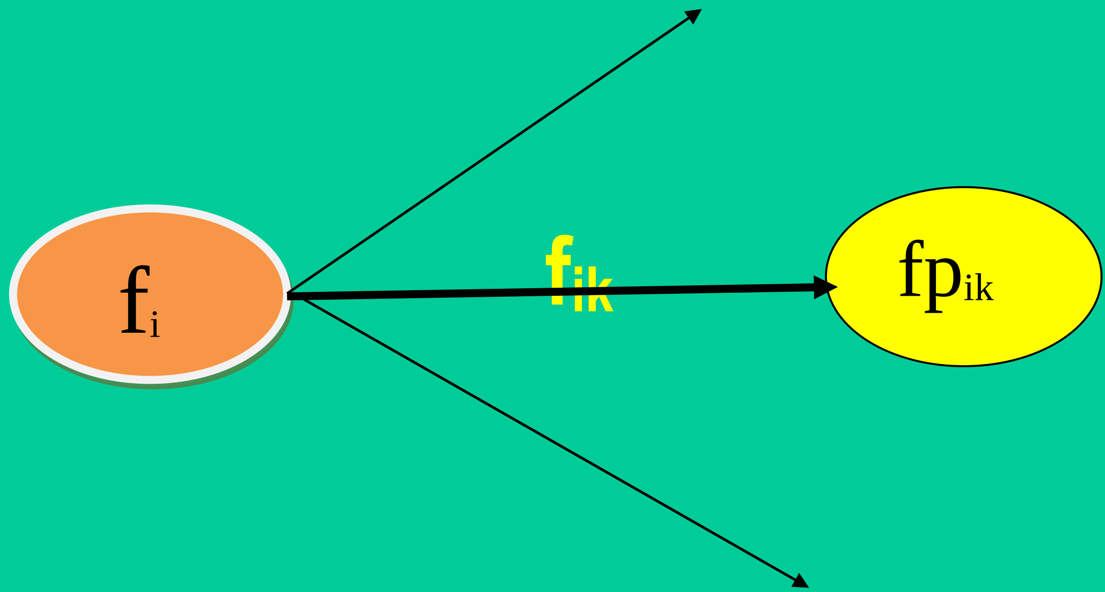
C'est le **réseau de transport**

## 1- <u>**Vue côté offre**</u>

Chaque couple

$$(\mathbf{F_i}\ ,\ \mathbf{P_k})$$

est caractérisé par la capacité de **production** ou **offre** notée $f_{ik}$ du site $\mathbf{F_i}$ en produits de modèle $\mathbf{P_k}$ .

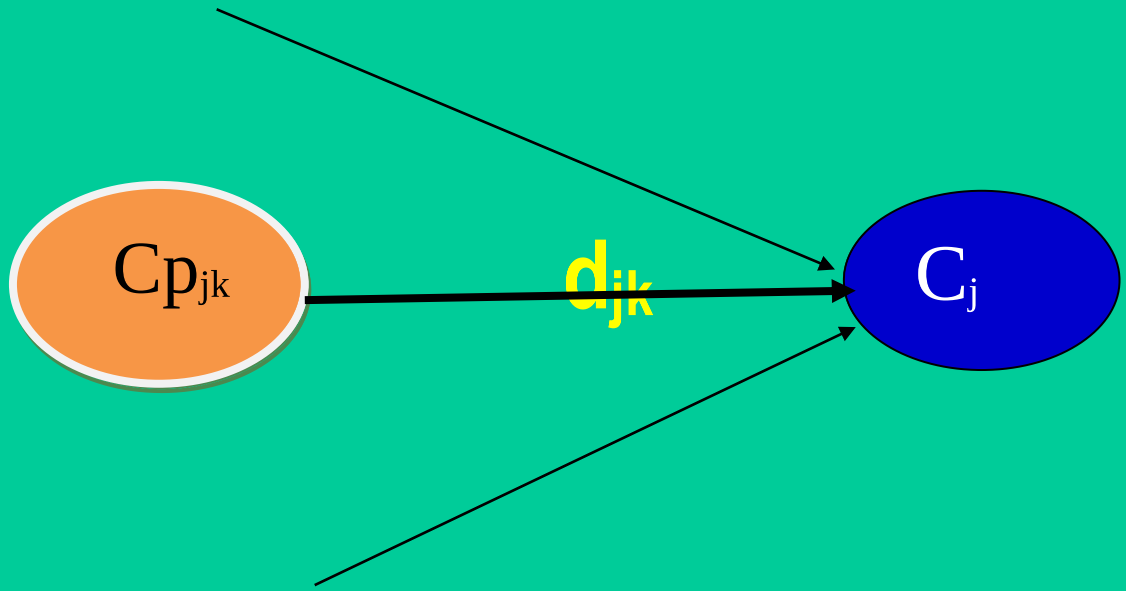

fi
fik
fpik

## 2- **Vue côté demande**

Réciproquement, chaque couple

$$(\mathbf{C_j} , \mathbf{P_k})$$

est caractérisé par la capacité de **consommation** ou **demande** notée $\mathbf{d_{jk}}$ en produits de modèle $\mathbf{P_k}$ sur le site $\mathbf{C_j}$.



Cpjk
djk
Cj

## 3- **Vue côté produit**

Un même modèle de produit $P_k$ peut être vu de deux façons:

-**côté offre** : $fp_{ik}$ pour représenter le produit $P_k$ offert par le site $F_i$

-**côté demande**: $Cp_{jk}$ pour représenter le produit $P_k$ demandé par le site de consommation $C_j$

## 4- **Vue côté «transport »**

Chaque couple

$(\mathbf{F_i}\ ,\ \mathbf{C_j})$

est défini par:

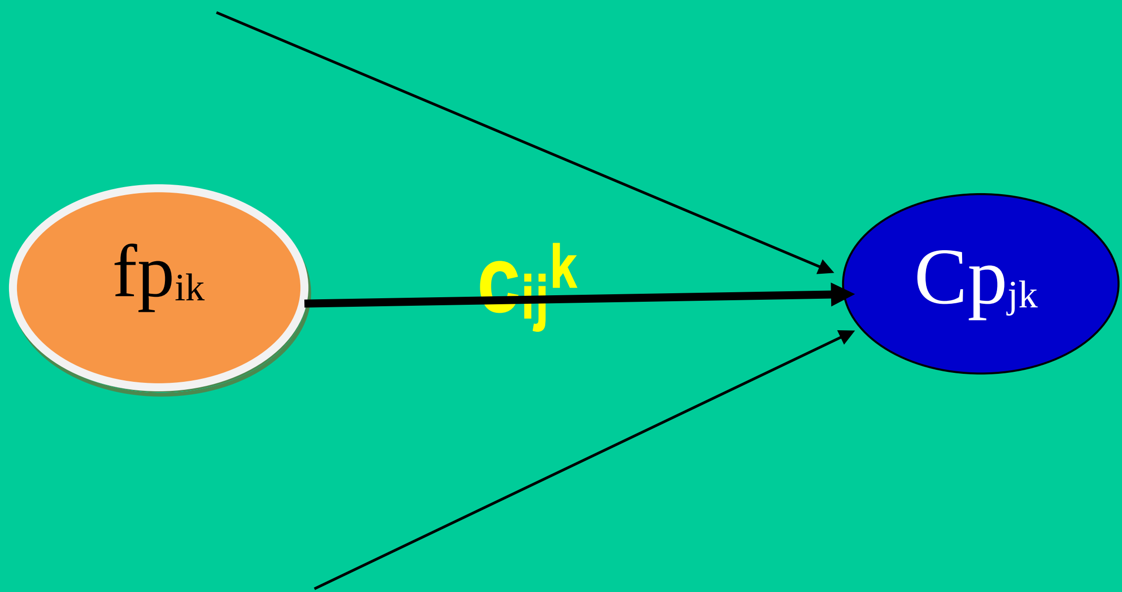
- le coût d'acheminement $c_{ij}^{k}$ d'un produit $\mathbf{P_k}$ depuis son site de fabrication $\mathbf{F_i}$ vers le lieu de consommation $\mathbf{C_j}$,

- et $l_{ij}^k$ et $u_{ij}^k$ , quantités respectives minimales et maximales et par produit $\mathbf{P_k}$ devant être livrées par le site $\mathbf{F_i}$ au site $\mathbf{C_j}$.

Ces quantités modélisent des contraintes qui peuvent être:

– du type **contractuel** (pour les $l_{ij}^k$ )

– ou bien refléter la **saturation** des lignes de transport (pour les $u_{ij}^k$ )


fpik
cijk
Cpjk

# Modèle de flot optimal/coût minimal

Ce problème ultra classique se modélise aisément comme un problème de **flot optimal de coût minimum**.

Le modèle de **réseau de transport** est défini selon les quatre étapes décrites ci-après.

**Etape 1**:

Tout d'abord, chaque site de fabrication **$F_i$** est représenté par une **source** de contribution:

$$\forall\, i \in 1..n \qquad b(F_i) = \sum_{k=1}^{m} f_{ik}$$

## <u>Etape 2</u>:

A l'autre bout du réseau, chaque site de consommation **$C_j$** est représenté par un **puits** de contribution:

$$\forall\, j \in 1..q \qquad b(C_j) = -\sum_{k=1}^{m} d_{jk}$$

## <u>Etape 3</u>:

Les produits sont, quant à eux, représentés par deux ensembles de sommets.

Le premier modélise les associations **fabricant/produit**.

Le second modélise les associations **consommateur / produit**.

Le premier est l'ensemble des sommets notés $\mathbf{fp_{ik}}$.

Chaque sommet $\mathbf{F_i}$ est relié au sommet $\mathbf{fp_{ik}}$ si et seulement si il fabrique le produit $\mathbf{P_k}$.

La capacité de ces arcs est fixée à $\mathbf{f_{ik}}$, capacité maximale de fabrication du produit $\mathbf{P_k}$ par le site $\mathbf{F_i}$ .

Le second est l’ensemble des sommets $\mathbf{cp_{jk}}$ .

Chaque sommet $\mathbf{C_j}$ est relié aux sommets $\mathbf{cp_{jk}}$ associés aux produits qu'il consomme par des arcs de capacité $\mathbf{d_{jk}}$.

# Etape 4:

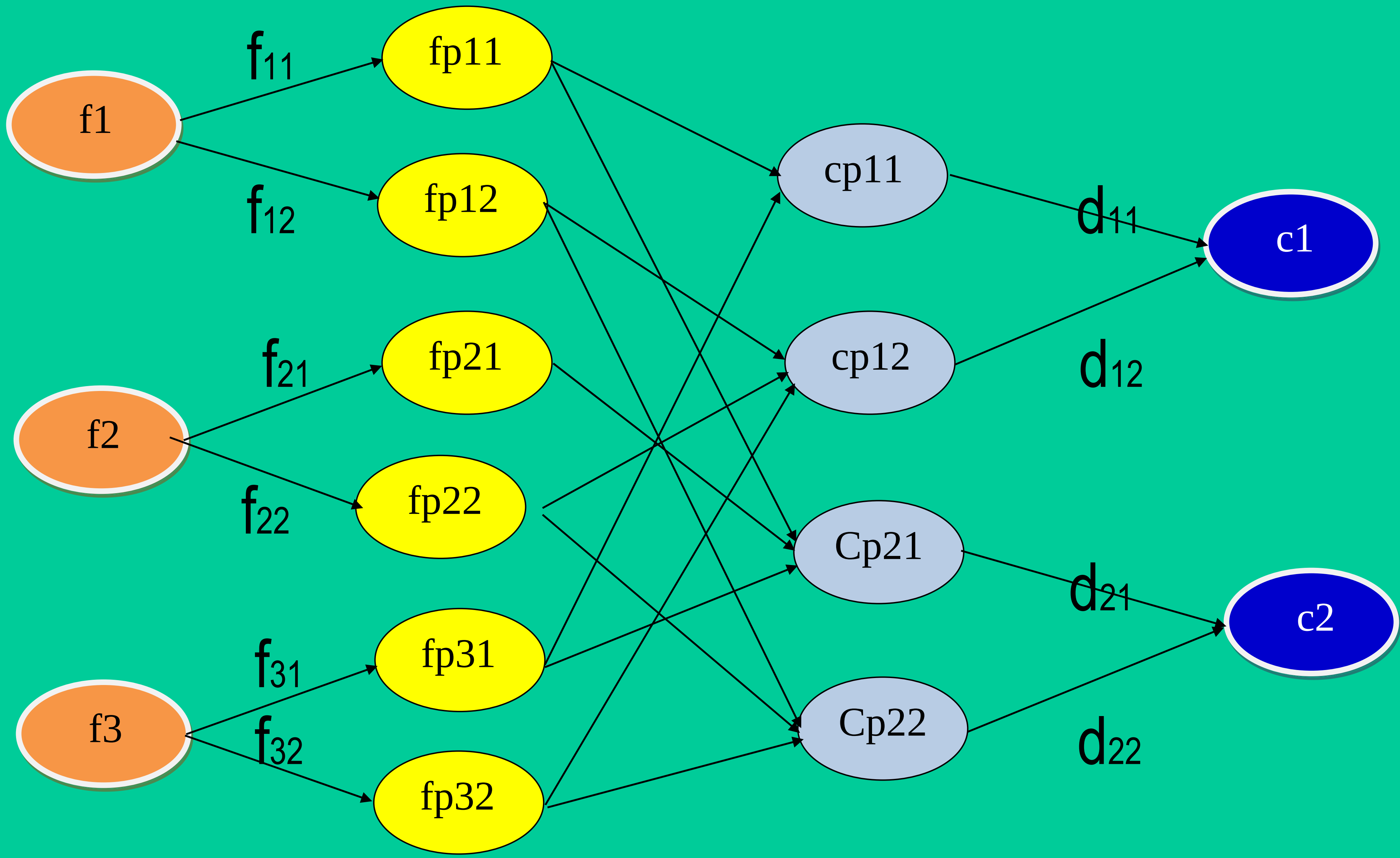
Finalement les **arcs de transport** relient les sommets **$fp_{ik}$** aux sommets **$cp_{jk}$** de même produit **$P_k$**.

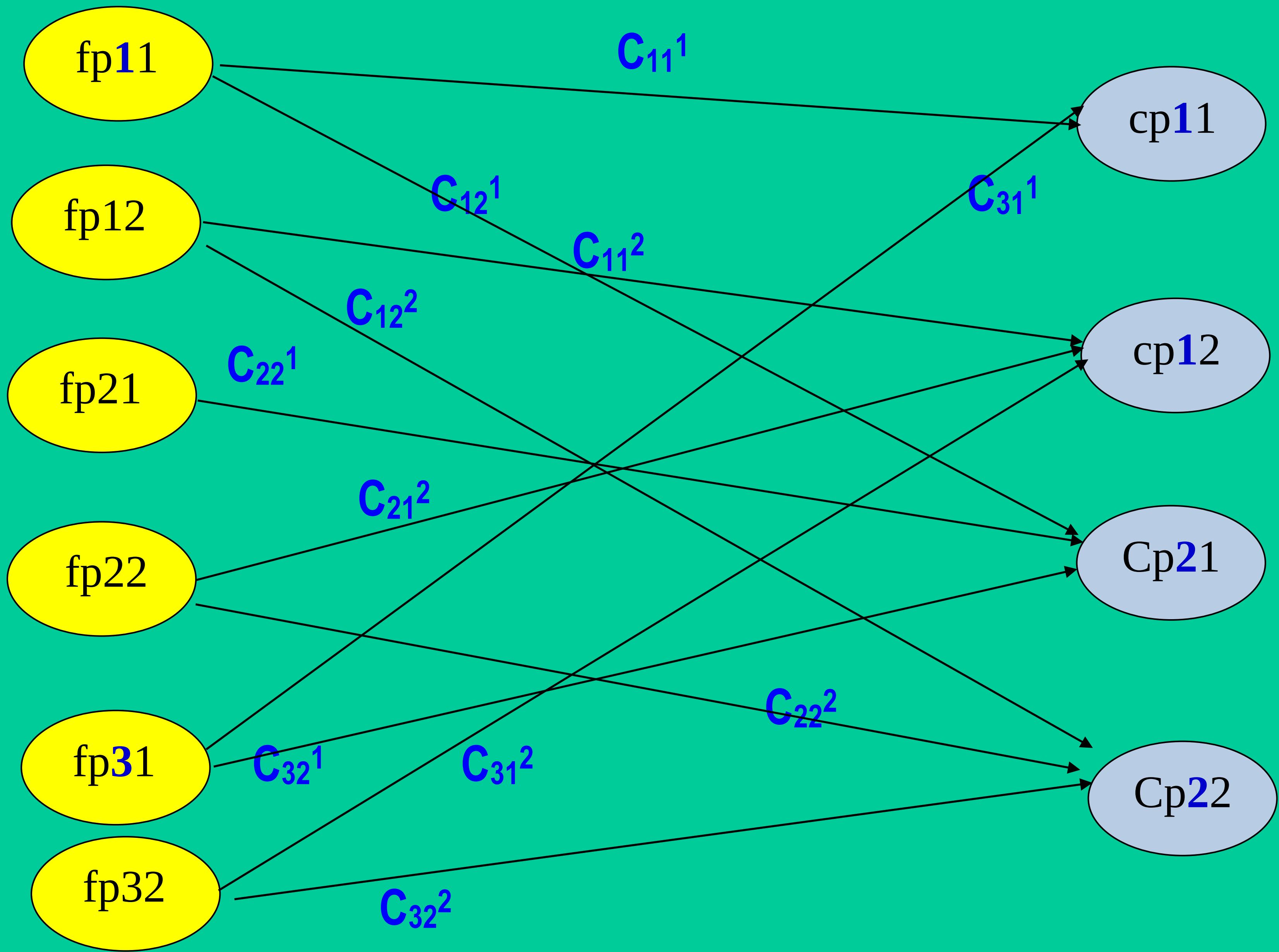
Leurs capacités sont **$l_{ij}^k$** (=0) et **$u_{ij}^k$** (>0) alors que leur coût est fixé à **$c_{ij}^k$**.

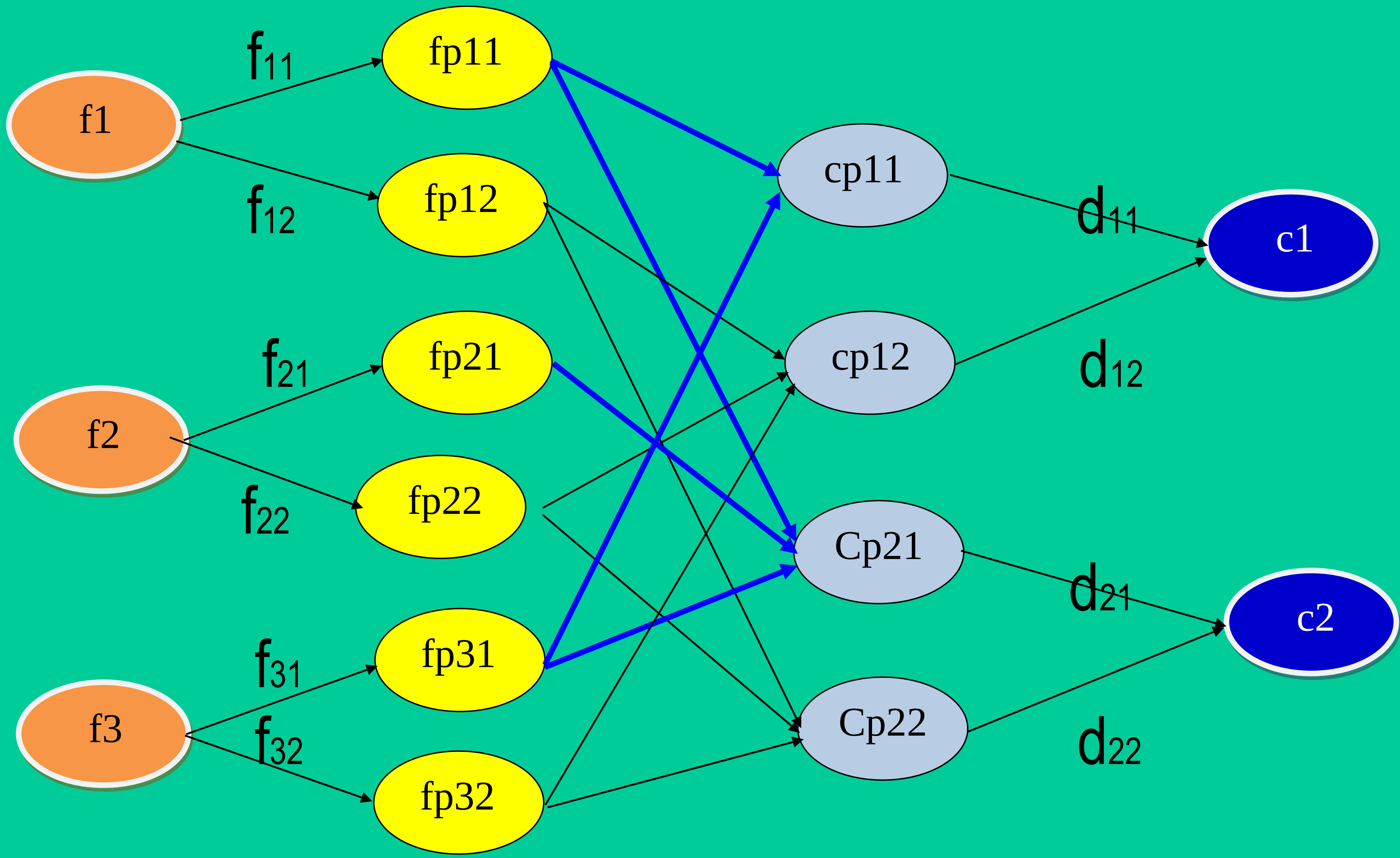
La recherche du **flot maximal à coût minimal** sur ce réseau garantit l'optimalité de la solution.

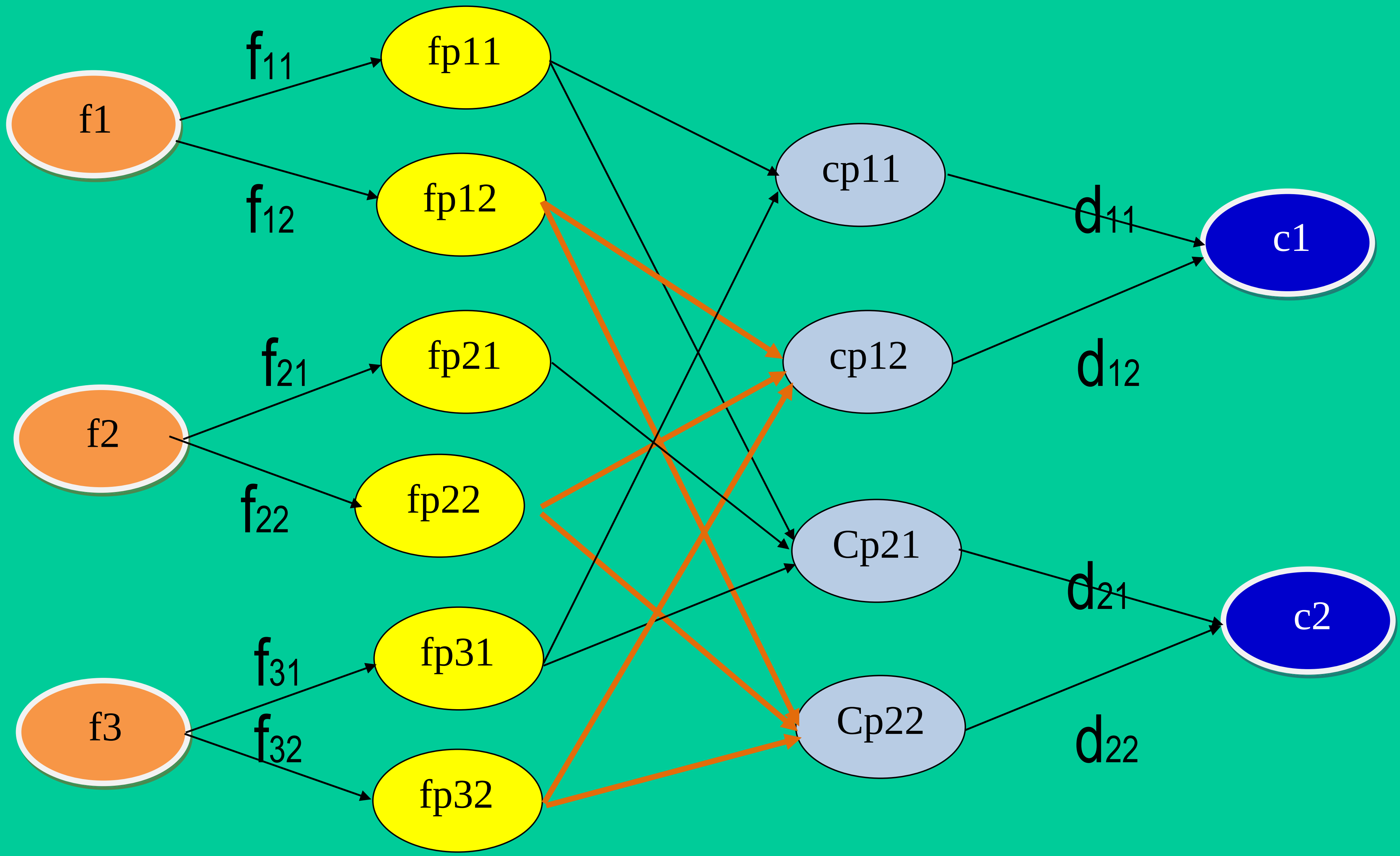
Il suffit alors de suivre les chaînes de flot pour retracer le trajet des marchandises.

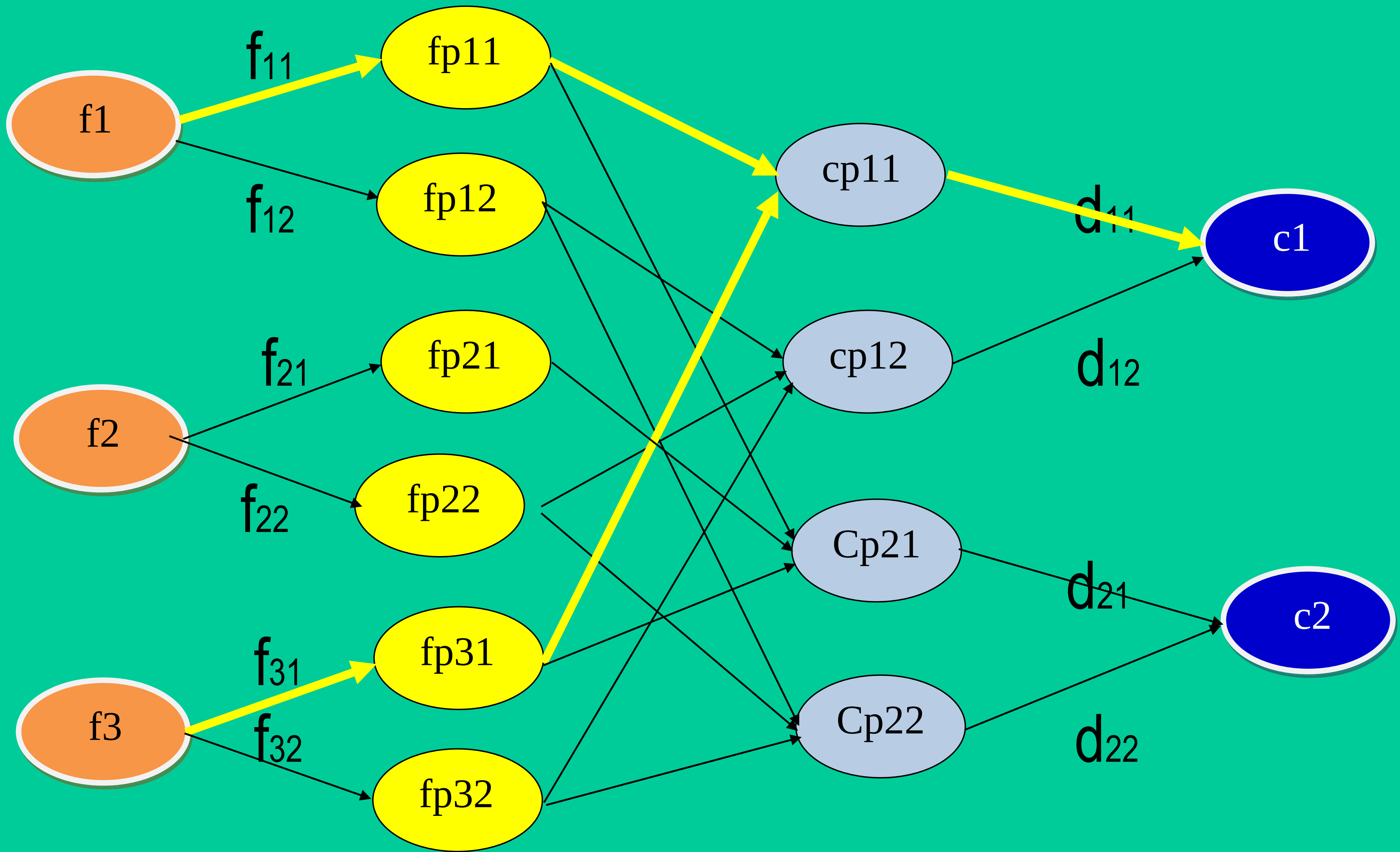
La figure suivante montre un cas à trois fabricants, deux produits, et deux centres de consommations.

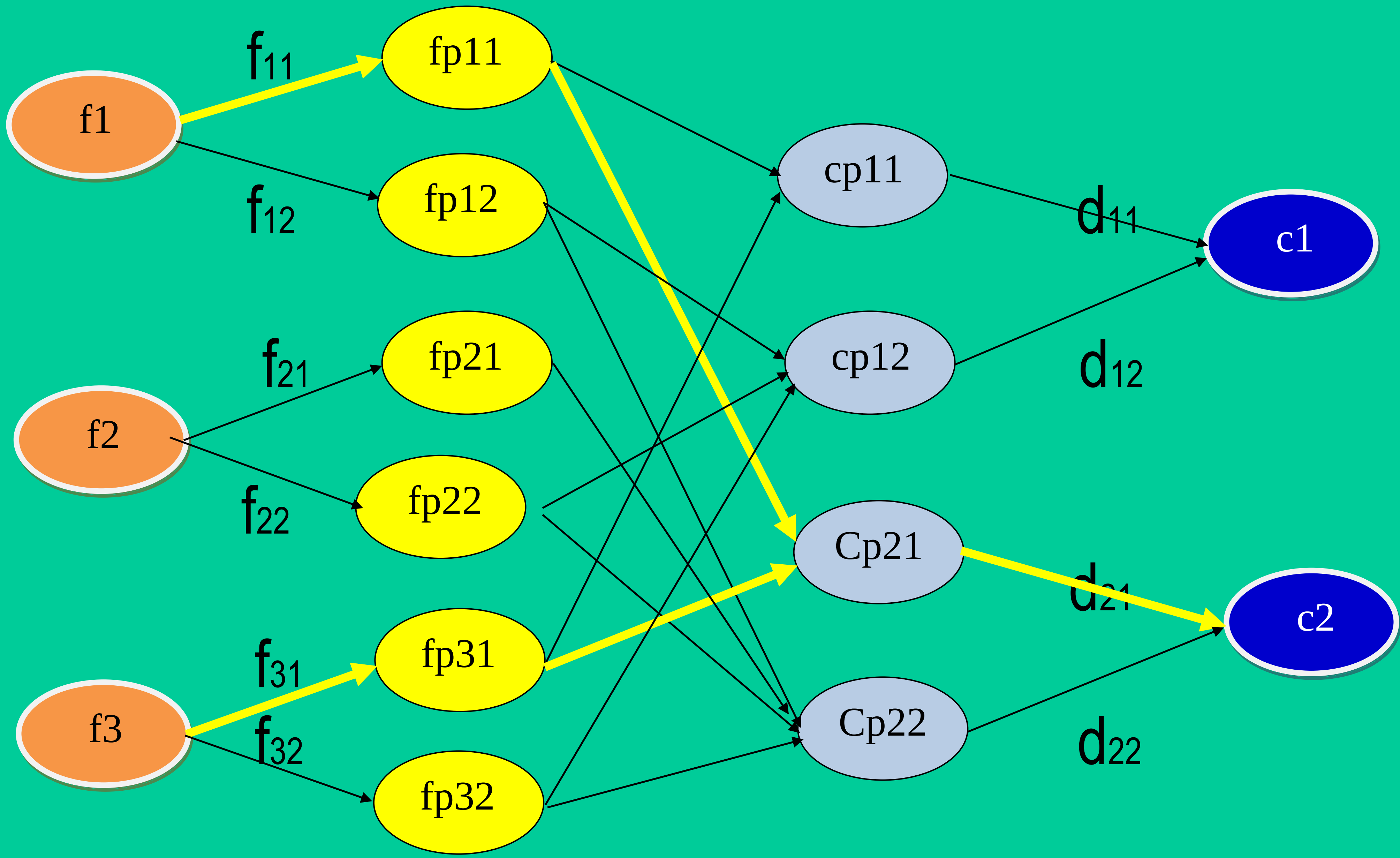

f1
f2
f3
f11
f12
f21
f22
f31
f32
fp11
fp12
fp21
fp22
fp31
fp32
cp11
cp12
Cp21
Cp22
d11
d12
d21
d22
c1
c2


fp11
fp12
fp21
fp22
fp31
fp32
cp11
cp12
Cp21
Cp22
$C_{11}^1$
$C_{12}^1$
$C_{11}^2$
$C_{31}^1$
$C_{12}^2$
$C_{22}^1$
$C_{21}^2$
$C_{22}^2$
$C_{32}^1$
$C_{31}^2$
$C_{32}^2$


f1
f2
f3
$f_{11}$
$f_{12}$
$f_{21}$
$f_{22}$
$f_{31}$
$f_{32}$
fp11
fp12
fp21
fp22
fp31
fp32
cp11
cp12
Cp21
Cp22
$d_{11}$
$d_{12}$
$d_{21}$
$d_{22}$
c1
c2


f1
f2
f3
f11
f12
f21
f22
f31
f32
fp11
fp12
fp21
fp22
fp31
fp32
cp11
cp12
Cp21
Cp22
d11
d12
d21
d22
c1
c2


f1
f2
f3
f11
f12
f21
f22
f31
f32
fp11
fp12
fp21
fp22
fp31
fp32
cp11
cp12
Cp21
Cp22
d11
d12
d21
d22
c1
c2



f1
f2
f3
$f_{11}$
$f_{12}$
$f_{21}$
$f_{22}$
$f_{31}$
$f_{32}$
fp11
fp12
fp21
fp22
fp31
fp32
cp11
cp12
Cp21
Cp22
$d_{11}$
$d_{12}$
$d_{21}$
$d_{22}$
c1
c2

# II- Formulation du problème

On considère que dans un réseau de flot (G,s,t) chaque arc (i,j) est muni d'un coût $c_{ij}$ dit **coût unitaire de flux**.

Le **coût du flot** porté par ce réseau est:

$$\sum_{(i,j) \in G} x_{ij} \times c_{ij}$$

Le problème de «**flot Max/cout Min**» consiste à :

- trouver un flot **V** **maximum** sur le réseau
- tel que le coût de ce flot soit **minimum**.

Beaucoup de problème dans la vie courante se ramènent à la recherche d'un flot maximal à coût minimal.

Les algorithmes les plus performants travaillent alors en deux temps.

**Dans un premier temps**:

On cherche le **flot maximal**: ce qui permet de **fixe**r sa **valeur V**.

**Dans un deuxième temps:**

-soit on recherche *ex nihilo* un **nouveau flot** de **valeur V** de coût minimal,

- soit on **modifie le flot** obtenu précédemment afin de rendre son coût minimal.

Ce problème est d'une **importance cruciale** en recherche opérationnelle.

En effet, comme nous le verrons plus tard, il sert à modéliser de nombreux cas concrets.

# Techniques de résolution

Il existe deux techniques de résolution du problème:

1- formulation comme un **programme linéaire** et application particulière du **simplexe** (simplexe réseau),

2- application d'algorithmes de **chemins augmentants**.

# Formulation par un Programme Linéaire(PL)

## 1-Rappel des **lois sur le flot**

$$0 \leq x_{ij} \leq u_{ij} \qquad \forall (i,j) \in A \qquad \text{(flux borné)}$$

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = 0 \qquad \forall i \in X-\{s,t\}$$

(conservation du flot)

$$V = \sum_{j:(s,j) \in A} x_{sj} = - \sum_{j:(j,t) \in A} x_{jt}$$

(flot sortant de s) (flot entrant en t)

## 2-Problème de **flot max** sous la forme d'un **PL**

$$\text{Max} \sum_{j:(s,j) \in A} x_{sj}$$

**Sc**

$$\sum_{j:(i,j) \in A} x_{ij} \; - \; \sum_{j:(j,i) \in A} x_{ji} = 0 \quad \forall i \in X-\{s,t\}$$

$$\sum_{j:(s,j) \in A} x_{sj} = - \sum_{j:(j,t) \in A} x_{jt}$$

$$\mathbf{0} \leq x_{ij} \leq \mathbf{u}_{\mathbf{ij}} \qquad \forall (i,j) \in A$$

3-Problème de **flot max/coût min** sous la forme d'un PL

$$\mathbf{Min} \sum_{(i,j) \in X} C_{ij}\, X_{ij}$$

Sc

$$\sum_{j:(i,j) \in A} X_{ij} \;-\; \sum_{j:(j,i) \in A} X_{ji} = 0 \quad \forall i \in X-\{s,t\}$$

$$\sum_{j:(s,j) \in A} X_{sj} = - \sum_{j:(j,t) \in A} X_{jt}$$

$$0 \leq X_{ij} \leq u_{ij} \qquad \forall (i,j) \in A$$

# 3-Problème de flot max/cout min: cas général

$$\textbf{Min} \sum_{(i,j) \in X} C_{ij} \, x_{ij}$$

Sc

$$\sum_{j:(i,j) \in A} X_{ij} - \sum_{j:(j,i) \in A} X_{ji} = \mathbf{b_i} \quad \forall i \in X$$

$$0 \leq x_{ij} \leq u_{ij} \qquad \forall (i,j) \in A$$

# III- Algorithme de Roy (Busacker et Gowen)

**Principe** :

Il consiste, **itérativement**, à améliorer un flot qui doit être initialisé à **0** .

Cette hypothèse garantit qu'on arrive **toujours** à un flot optimal.

**Remarque**:

Si, on commençait par un flot **aléatoire**, il ne serait pas garanti qu'on arrive à un flot optimal.

# <u>A chaque itération</u>:

- construire un graphe d'écart $G(x)$ qui dépend du flot actuel $X$ et du graphe du réseau $G$.

-Si $G$ a un arc $(i,j)$ avec coût $c_{ij}$,

-si $x_{ij} < u_{ij}$ alors le coût de l'arc $(i,j) \in G(x)$ est $c_{ij}$

-si $x_{ij} > 0$ alors le coût de l'arc $(j,i) \in G(x)$ est $-c_{ij}$

On cherche, dans le graphe G(x) le plus court chemin $\pi$ de la source **s** au puits **t**

**Si** le chemin $\pi$ existe :

**Alors**

-calculer la **capacité résiduelle minimum** du chemin $\pi$ notée $r_{ij}{}^{*}$

- **mettre à jour** le graphe G

**<u>Sinon</u>**:

--on ne trouve pas de chemin **$\pi$**

L'algorithme se **termine** : alors « le flot est **maximum** et son coût est **minimum** »

**Remarque**:

La recherche du **chemin de coût minimum** $\pi$ applique un algorithme bien connu de recherche du **plus court chemin** :

- entre deux sommets
- avec des arcs pouvant avoir un **coût négatif**.

# Procedure BusackerGowen

**Procedure** BusackerGowen( G; c;w)

$V_{max}= 0;$ $\forall (i,j) \in G \bullet X_{ij} = 0$

$C_{min}= 0;$

Construire le graphe d’écart G(x),

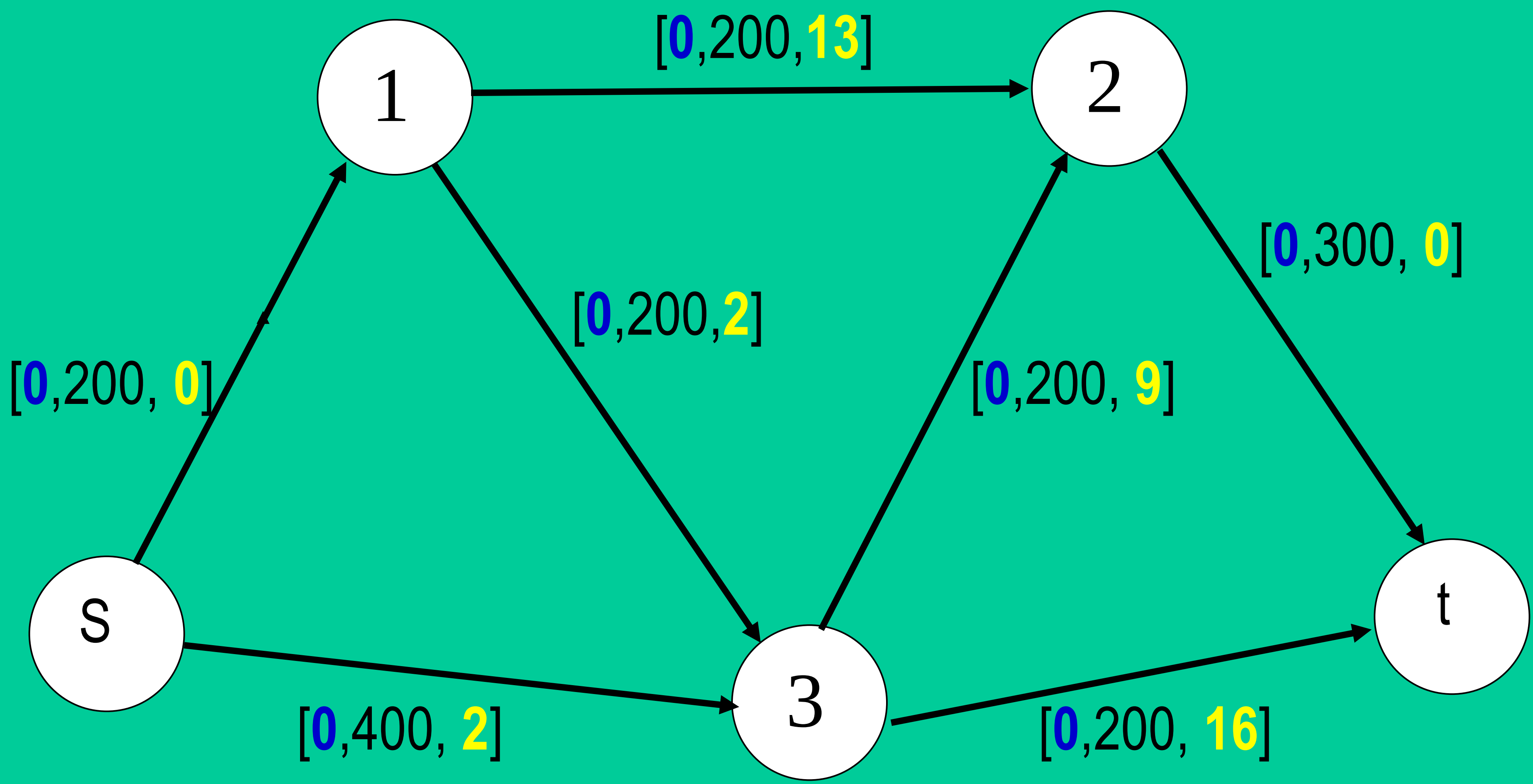
**TantQue**

« dans **G(x)**, on trouve un chemin $\pi$ de **s** à **t** de coût minimum $\varphi$»

- calculer la capacité résiduelle minimum $r_u^*$ sur $\pi$
- dans G, **augmenter** de $r_u^*$ le flux sur les arcs directs($\pi$+)
- dans G, **diminuer** de $r_u^*$ le flux sur les arcs indirects($\pi$-)
- $C_{min} = C_{min} + r_u^* . \varphi$ ; $V_{max} = V_{max} + r_u^*$.
- **mettre à jour** le graphe d'écart **G(x)**

**FinTant**

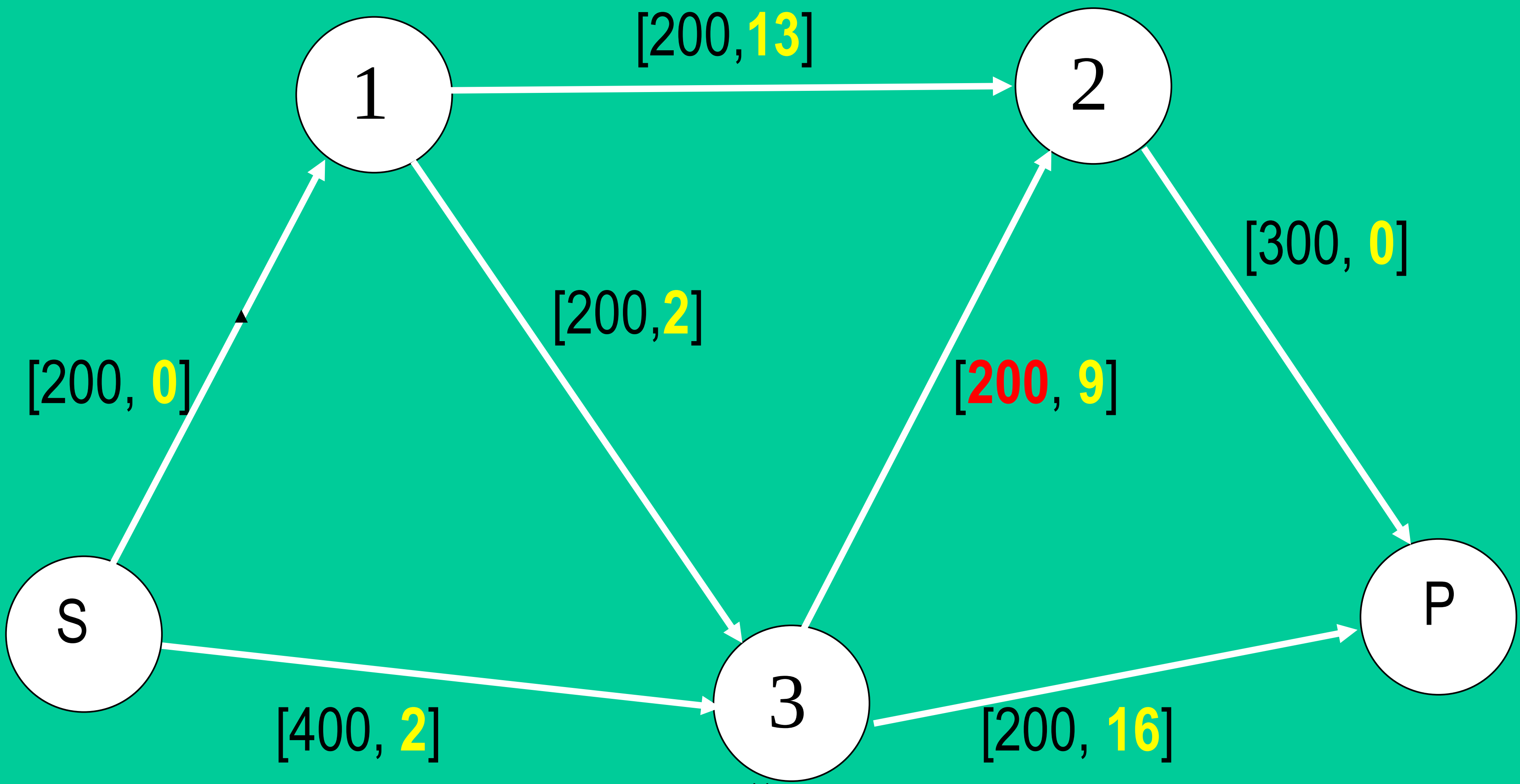
# Exemple d'application



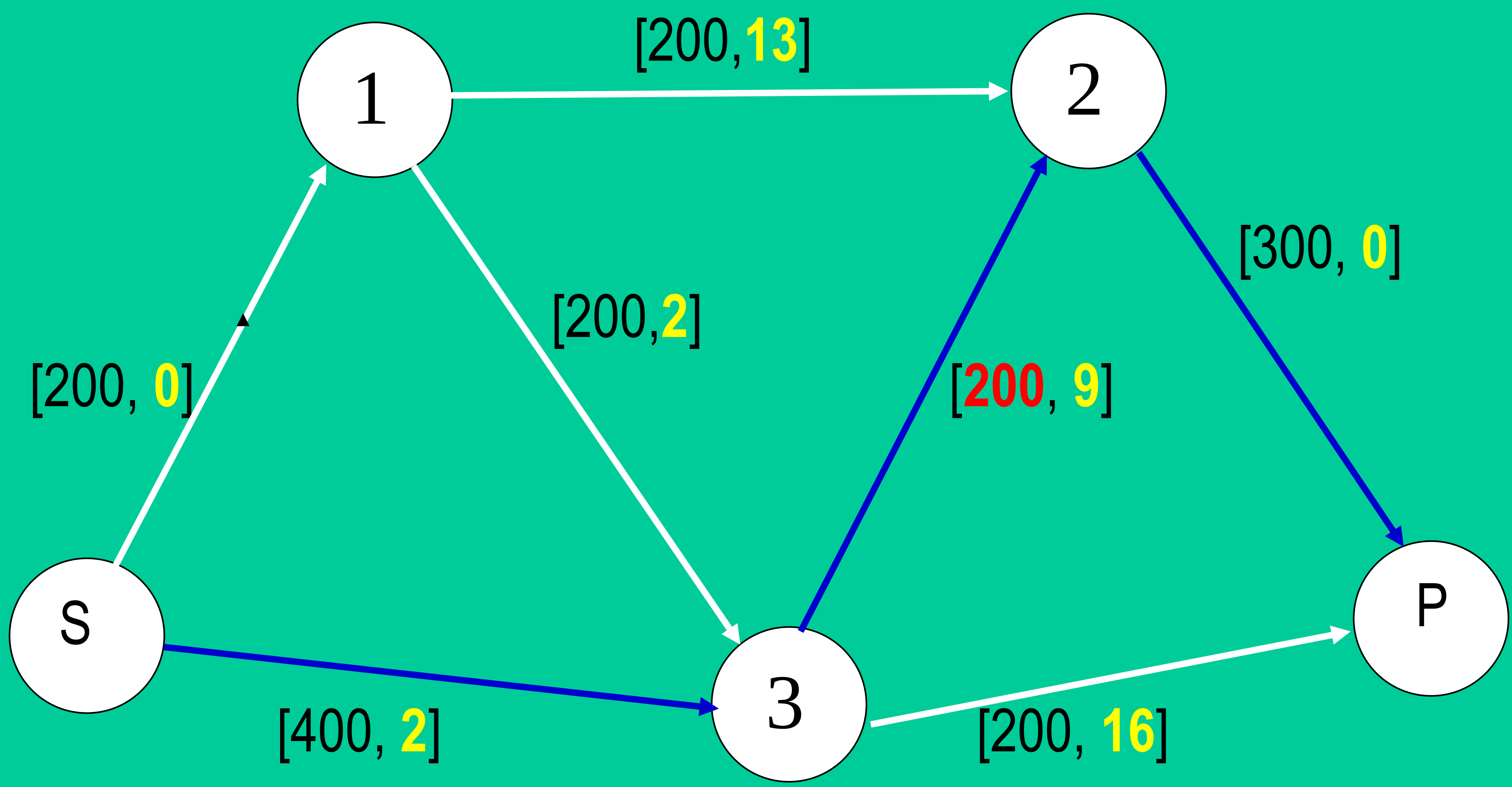

$V_{max} = 0$ $C_{min} = 0$

# Itération 1:

# Graphe d'écart



# Recherche du chemin $\pi$



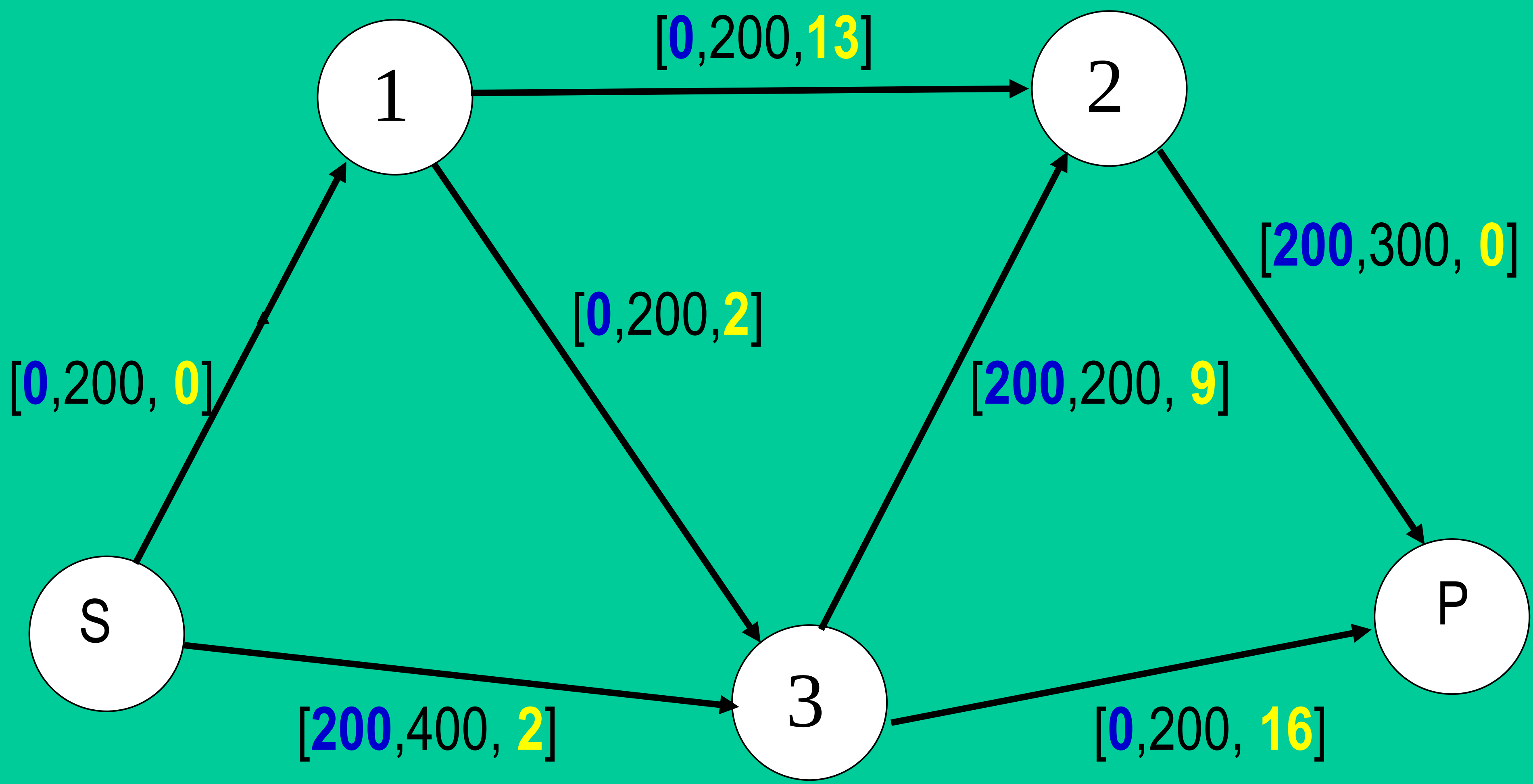
## Mises à jour de G :

$\varphi = 2+9+0 = 11$

$r_u^* = \min(400, 200, 300) = 200$

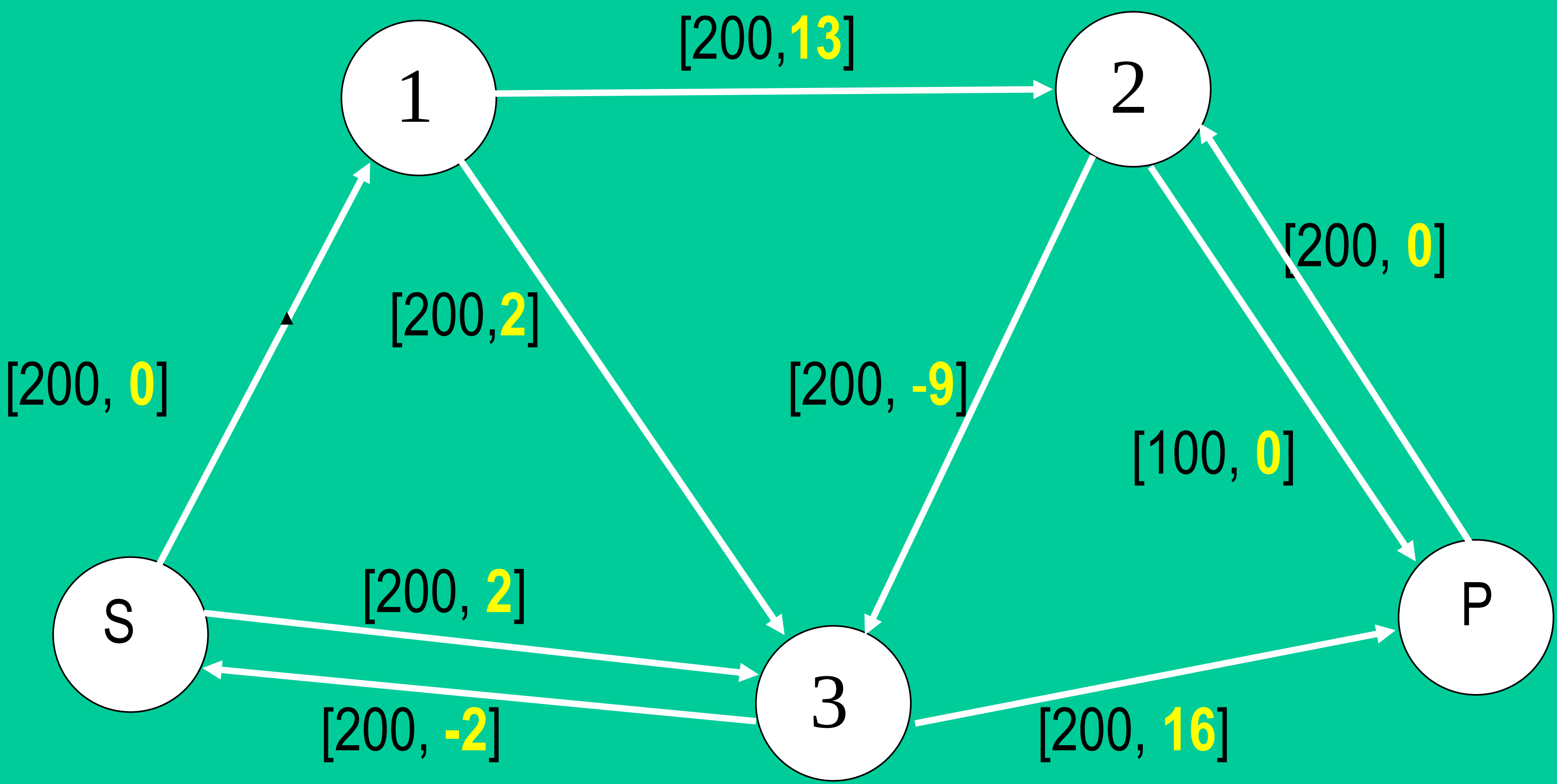
$C_{min} = 0 + 11 \times 200 = 2200$

$V_{max} = 0 + 200 = 200$

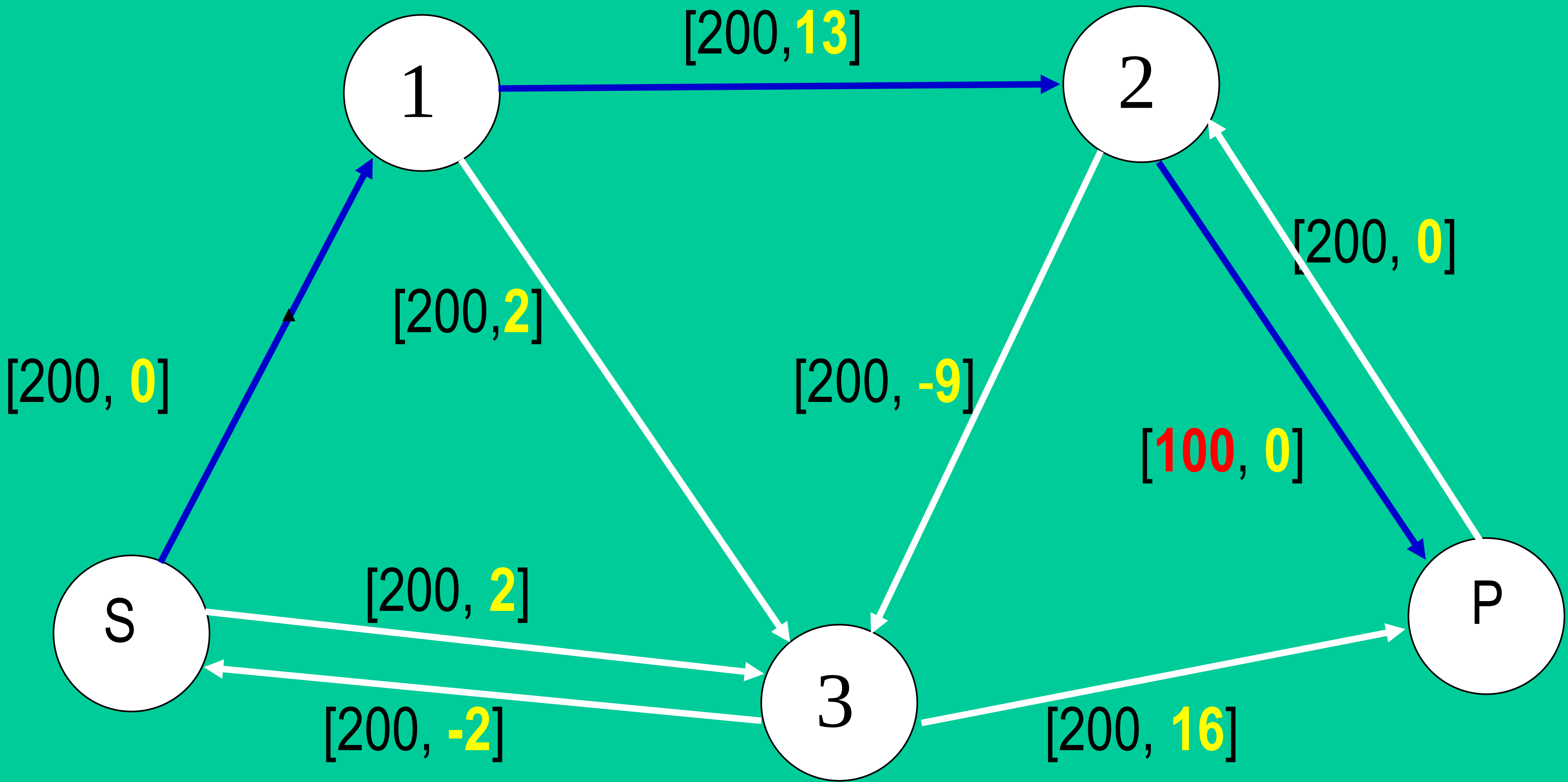

1
2
S
3
P
[0,200,13]
[0,200, 0]
[0,200,2]
[200,200, 9]
[200,300, 0]
[200,400, 2]
[0,200, 16]

# Itération 2:

# Graphe d'écart



# Recherche du chemin $\pi$



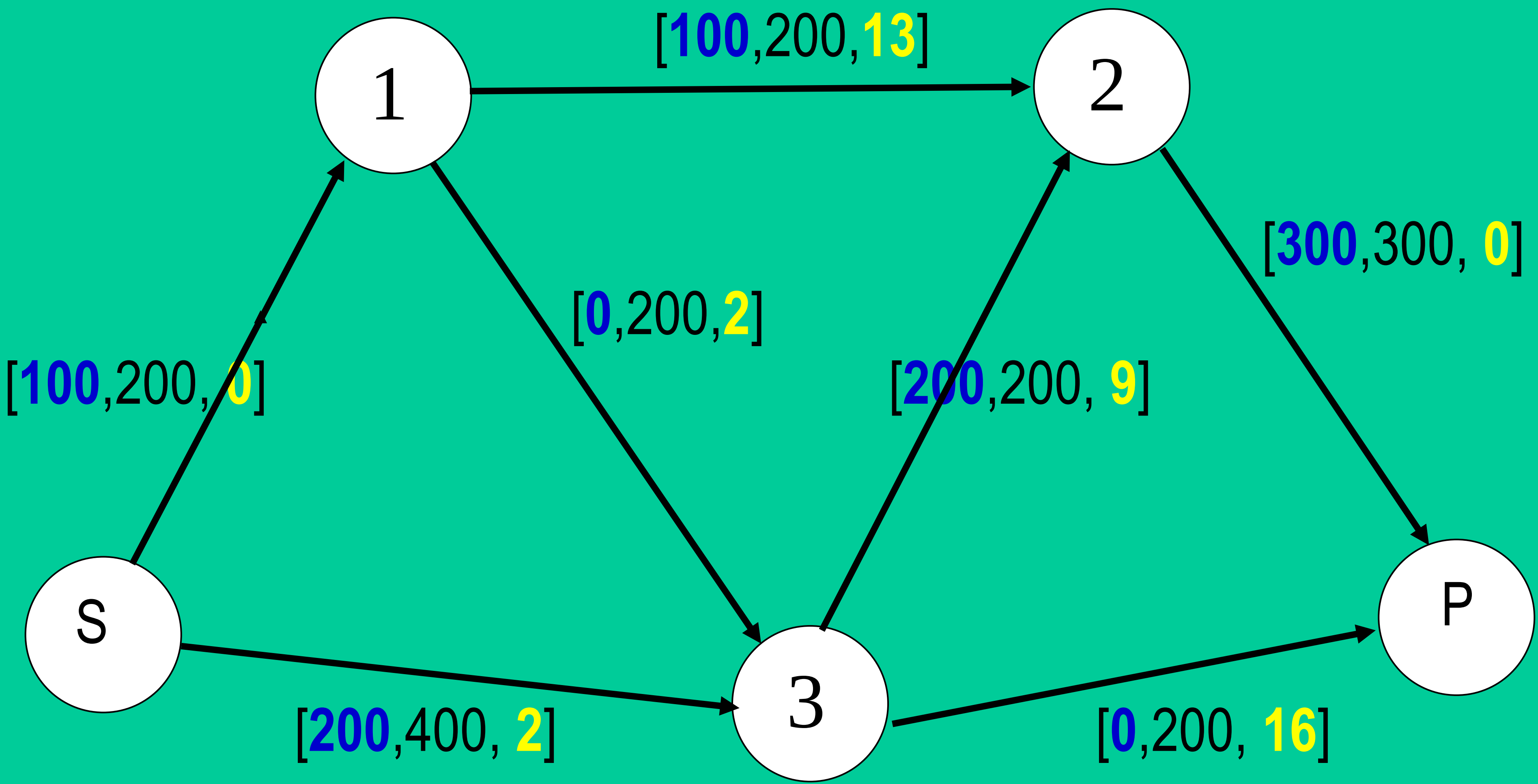
# Mises à jour de G

$\varphi = 0+13+0 = 13$

$r_{ij}{}^* = \min(200, 200, 100) = \mathbf{100}$

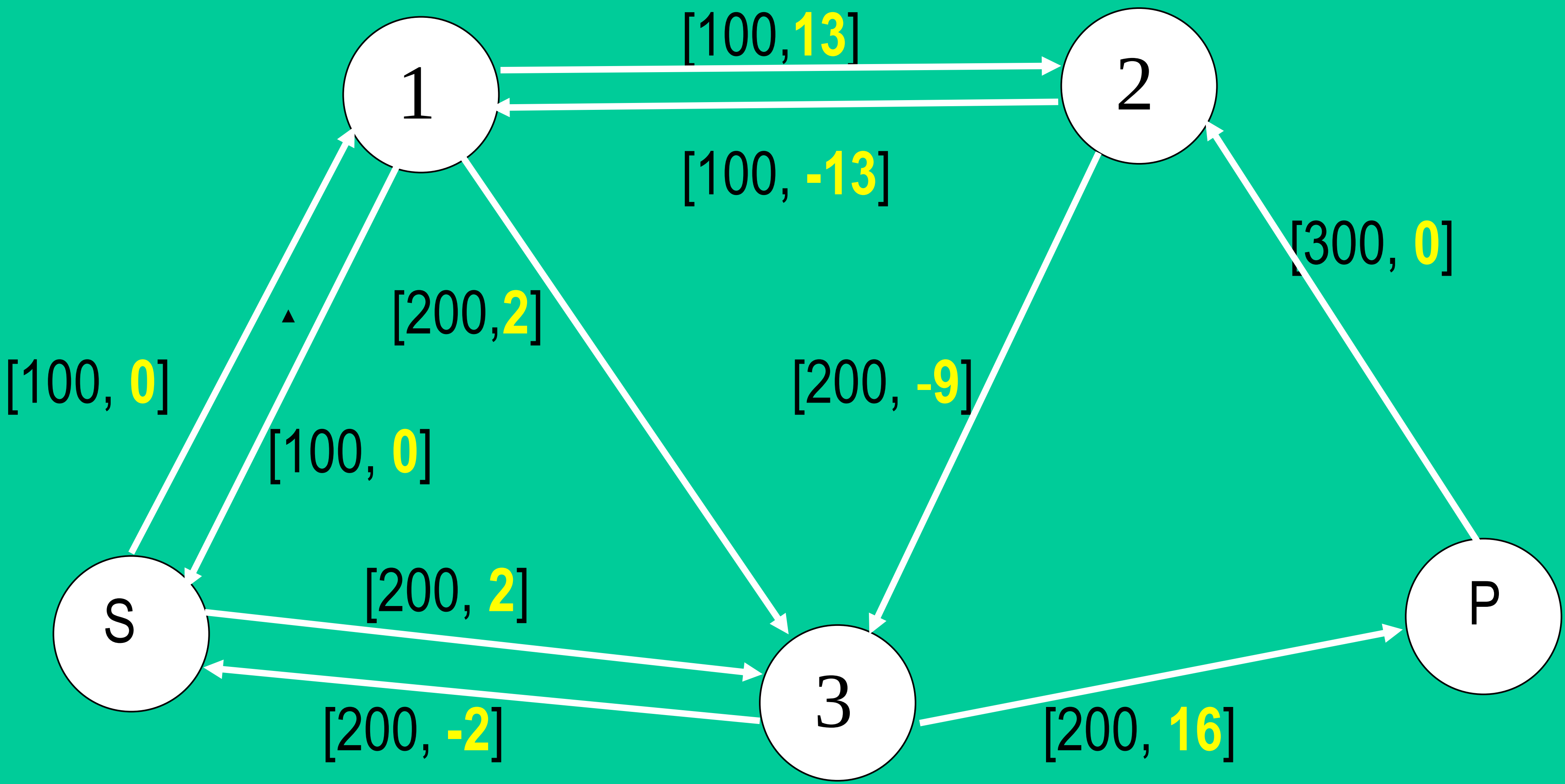
$C_{min} = 2200 + \mathbf{13} \times \mathbf{100} = 3500$

$V_{max} = 200 + \mathbf{100} = 300$

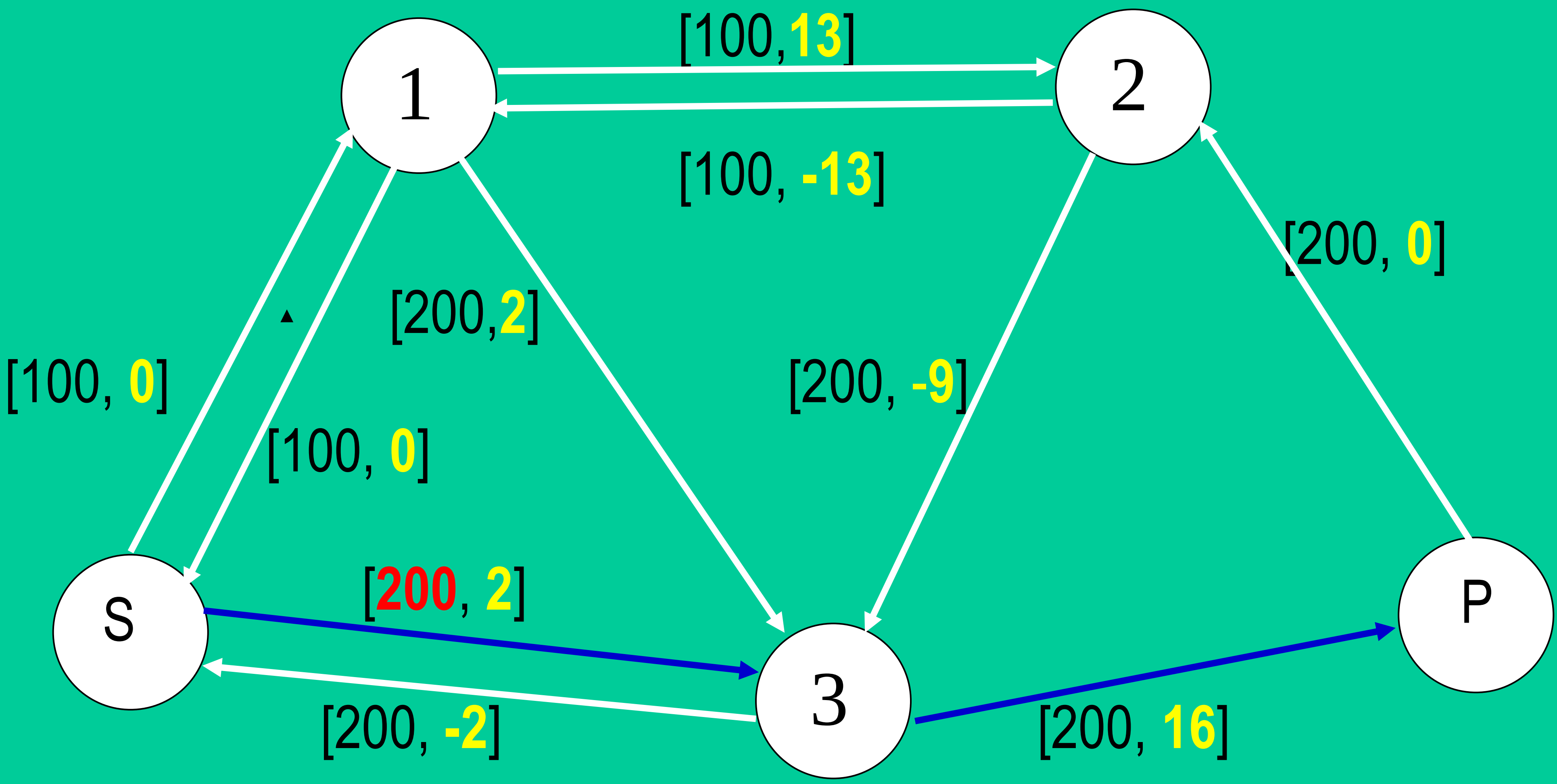

1
2
S
3
P
[100,200,13]
[0,200,2]
[300,300, 0]
[100,200, 0]
[200,200, 9]
[200,400, 2]
[0,200, 16]

# Itération 3:

## Graphe d'écart



# Recherche du chemin $\pi$



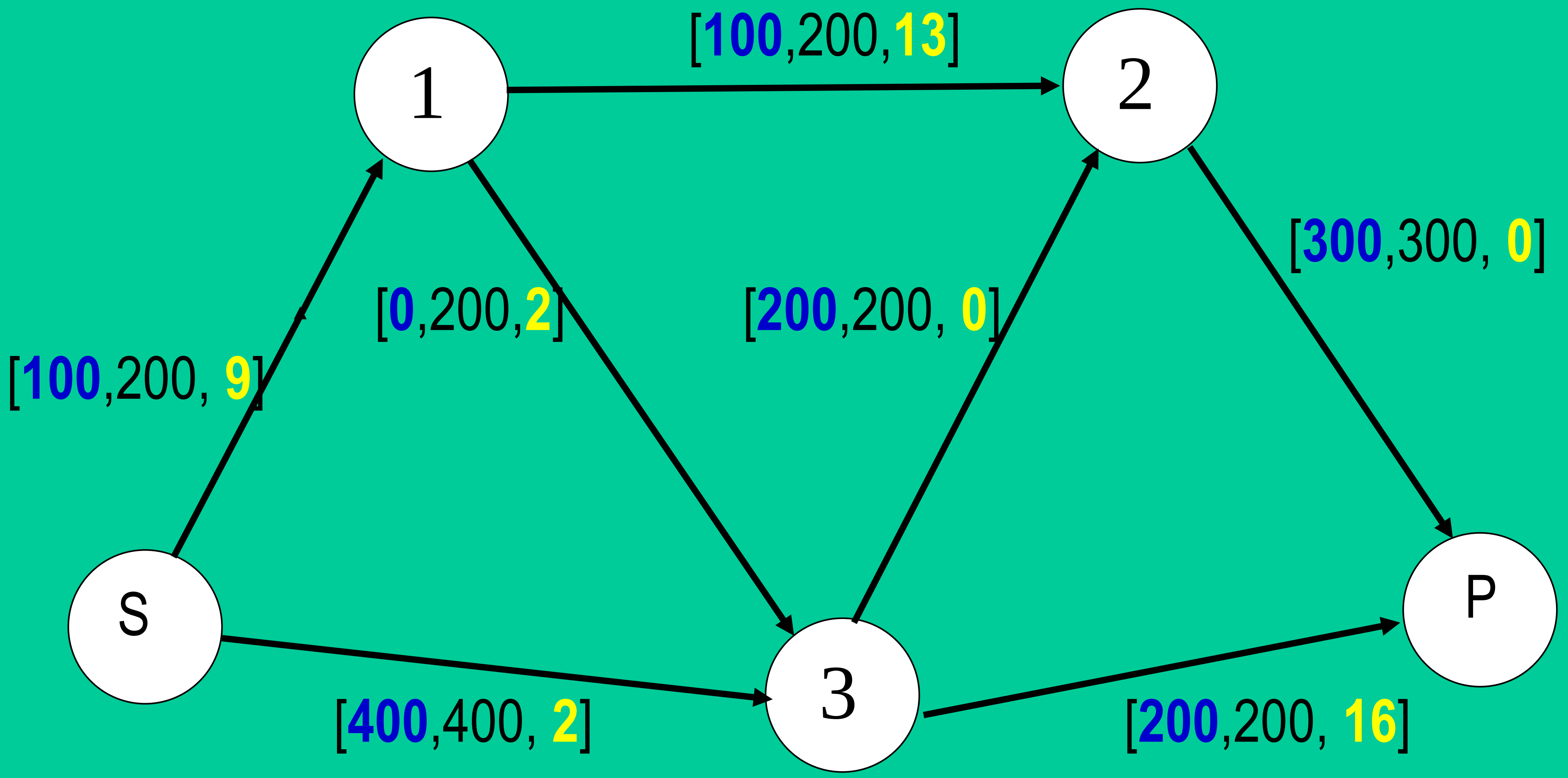
# Mises à jour de G

$\varphi = 2+16 = 18$

$r_{ij}^* = \min(200, 200) = 200$

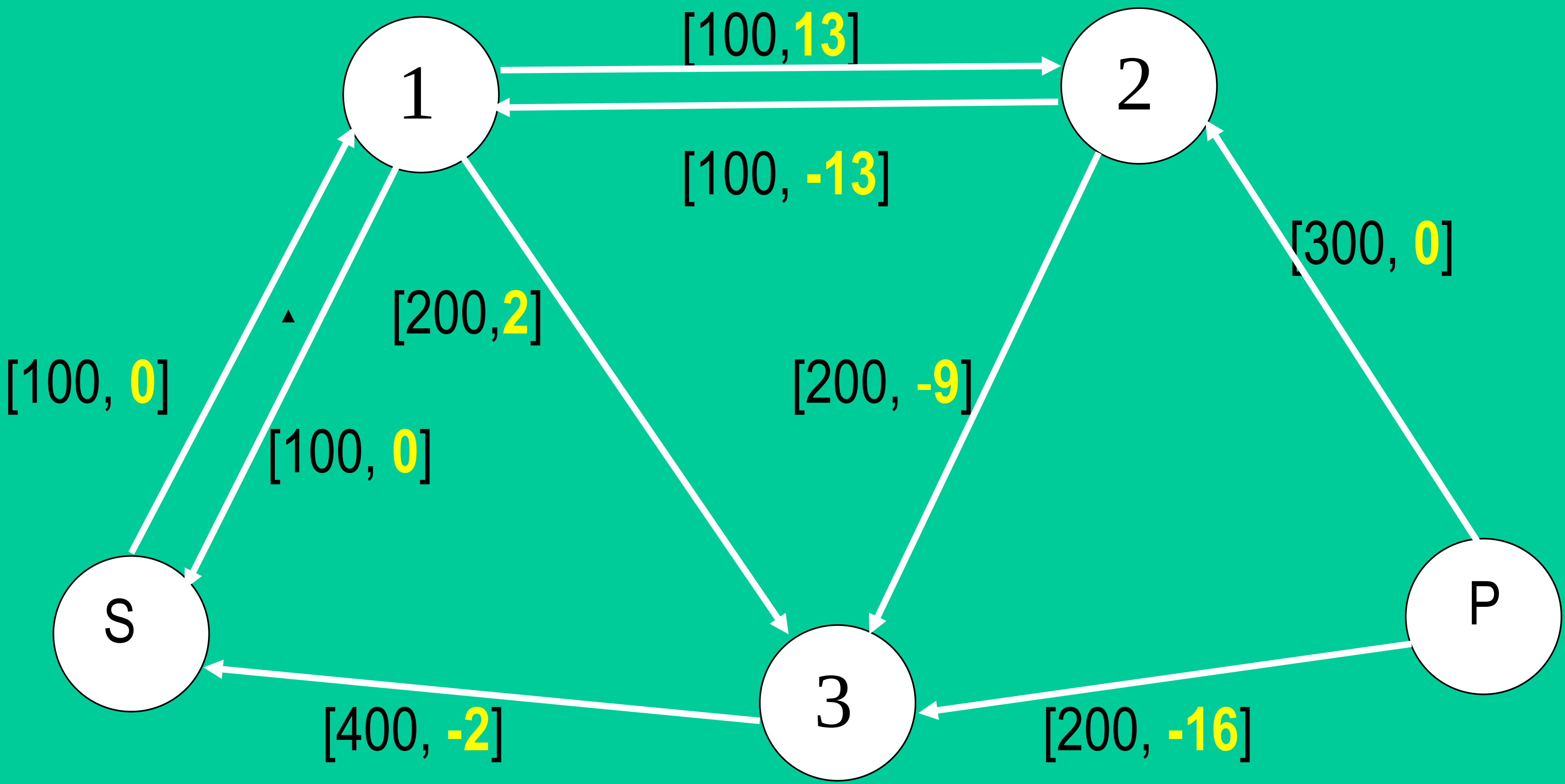
$C_{min} = 3500 + 18 \times 200 = 7100$

$V_{max} = 300 + 200 = 500$

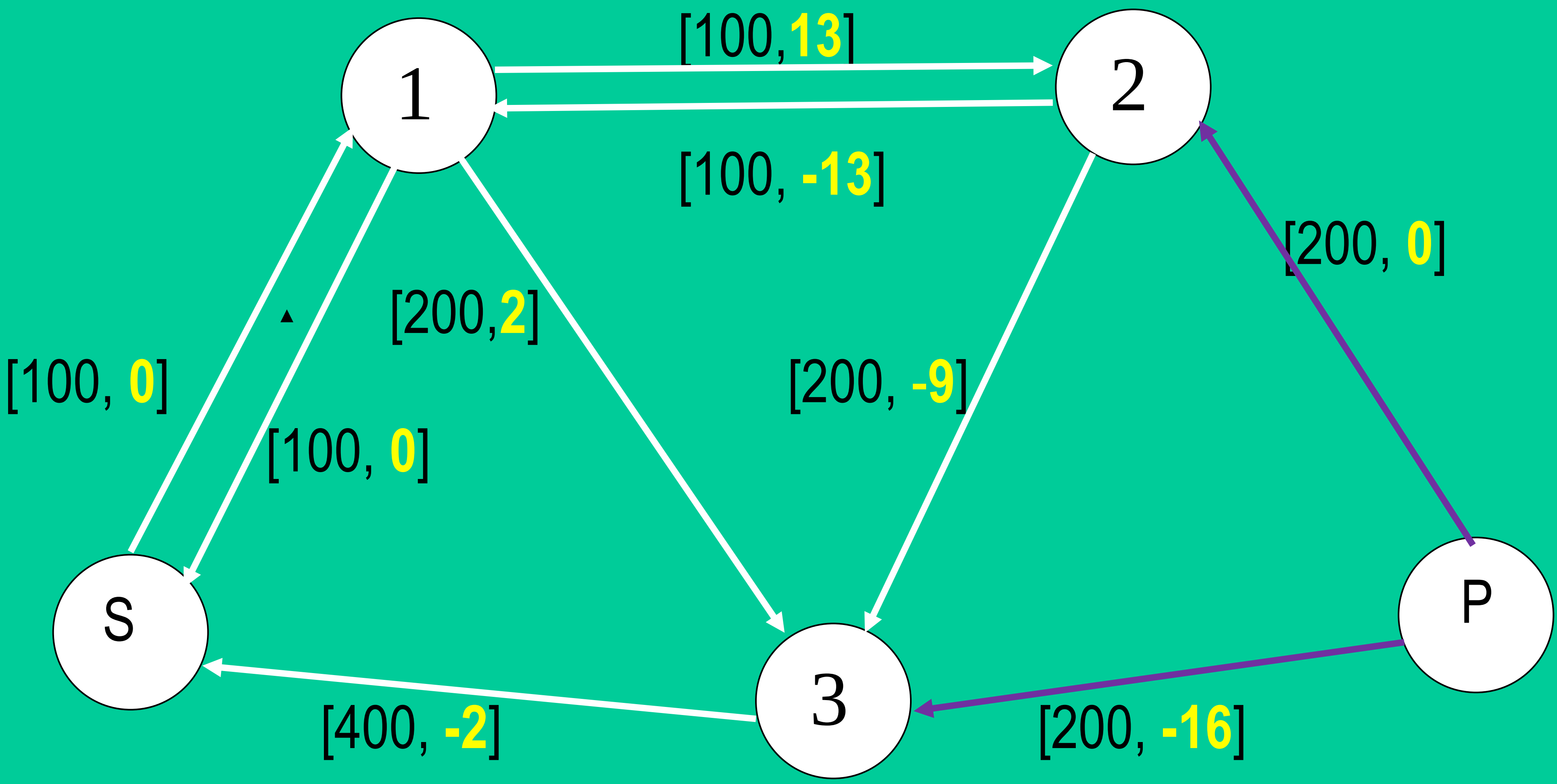

1
2
S
3
P
[100,200,13]
[100,200, 9]
[0,200,2]
[200,200, 0]
[300,300, 0]
[400,400, 2]
[200,200, 16]

# Itération 4:

# Graphe d'écart



# Recherche du chemin $\pi$ : échec




$V_{max}$= 500 $C_{min}$ = 7100