Rappels Complexité

La réduction

P_1 et P_2 deux problèmes.

On dira que le problème P_1 peut être *réduit* au problème P_2 s'il existe une transformation associant à chaque instance Π de P_1 une instance $f(\Pi)$ de P_2 ayant la même réponse que Π.

- **P est la classe des problèmes qu'on sait résoudre en temps polynomial sur une machine de Turing déterministe.**
- **NP est la classe des problèmes qu'on sait résoudre en temps polynomial sur une machine de Turing non-déterministe.**
- **On peut aussi définir NP comme la classe des problèmes admettant un témoin de solution de taille polynomiale, qu'on peut vérifier en temps polynomial sur une machine de Turing déterministe.**

Le GRAND problème

Problème : est-ce que l'inclusion de P dans NP est stricte ?

L'Institut Clay propose un prix de 1.000.000 $ pour une réponse à cette question (voir http://www.claymath.org/prizeproblems/pvsnp.htm)

La réduction polynomiale nous fournit un outil intéressant : en effet tous les problèmes de P sont polynomialement équivalents.

Et dans NP, il peut y avoir, éventuellement, d'autres classes d'équivalence.

Et parmis les différentes classes d'équivalence, il peut y avoir une classe des plus difficiles ! (un maximum, dans l'ordre quotient).

Ce maximum est dans NP mais pas dans P – sauf si P=NP !!!!!!!!!

Cette classe d'équivalence s'appelle la classe des problèmes NP-complètes.

Un problème Π est NP-complet,

si :

- **Π est dans NP**

et

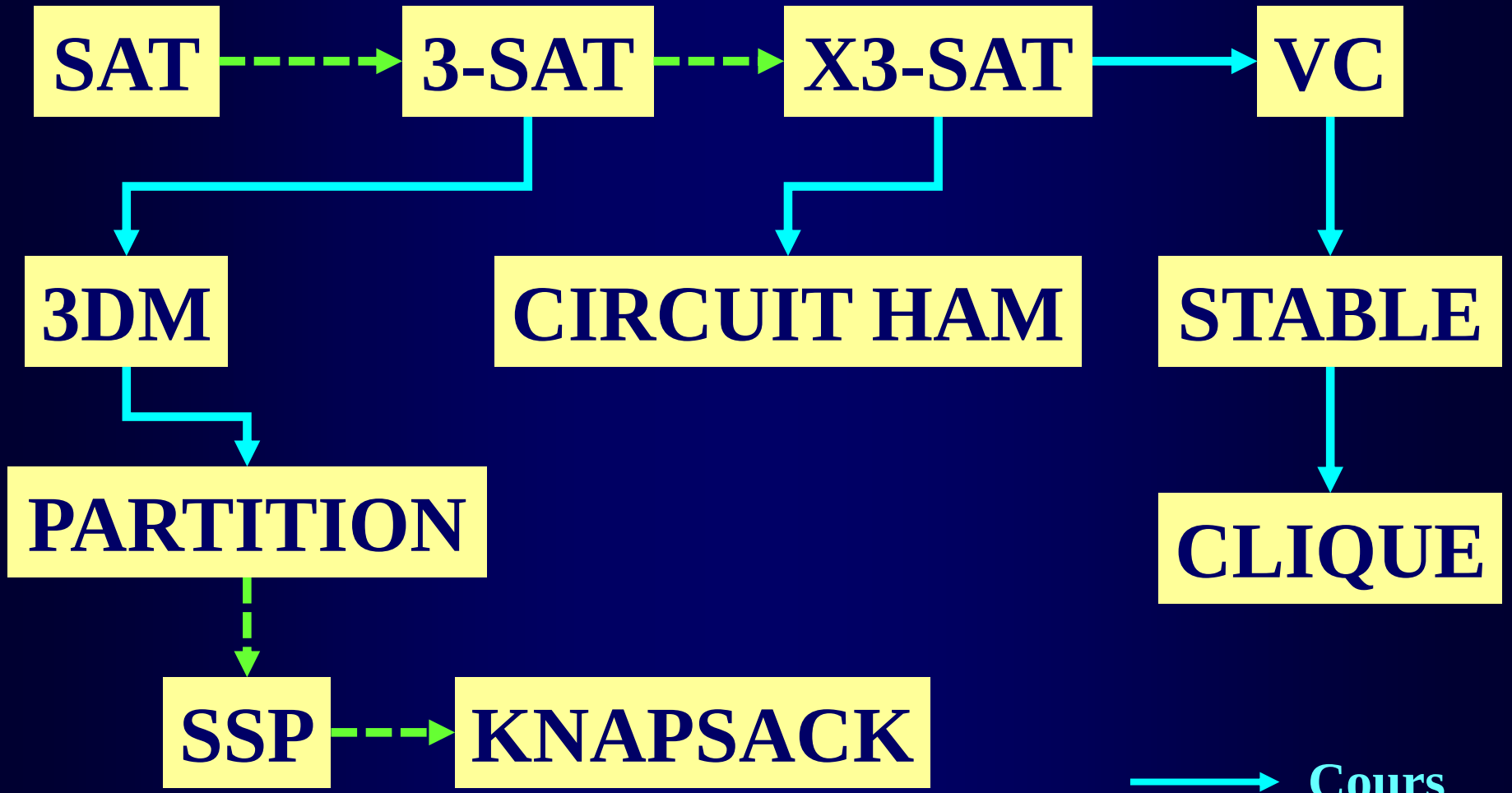
- **tout problème Π' de NP peut être réduit polynomialement à Π.**

COOK, a prouvé que SAT est NP-complet.

Depuis d'autres problèmes sont prouvés NP-complets.

Nous pouvons utiliser tout cela, car il suffit de prouver qu'un problème déjà connu comme NP-complet se réduit à notre problème.

Preuves de NP-complétude (schéma)



Rappel

NOM : VC (transversal)

DONNEES : un graphe fini G(V,E), et un entier positif $K \leq |V|$

QUESTION : est-ce que le graphe admet un transversal (un ensemble de sommets contenant au moins une extrémité de toute arête) de cardinalité au plus K ?

Preuve de la NP-complétude

i) **VC $\in$ NP**

ii) **VC est NP-difficile**

X3-SAT $\propto$ VC

La transformation

Soit $F = C_1 \wedge C_2 \wedge ... \wedge C_q$ une instance de X3-SAT avec $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$.

On construit un graphe ayant $3q$ sommets qui correspondent aux $3q$ littéraux de la formule.

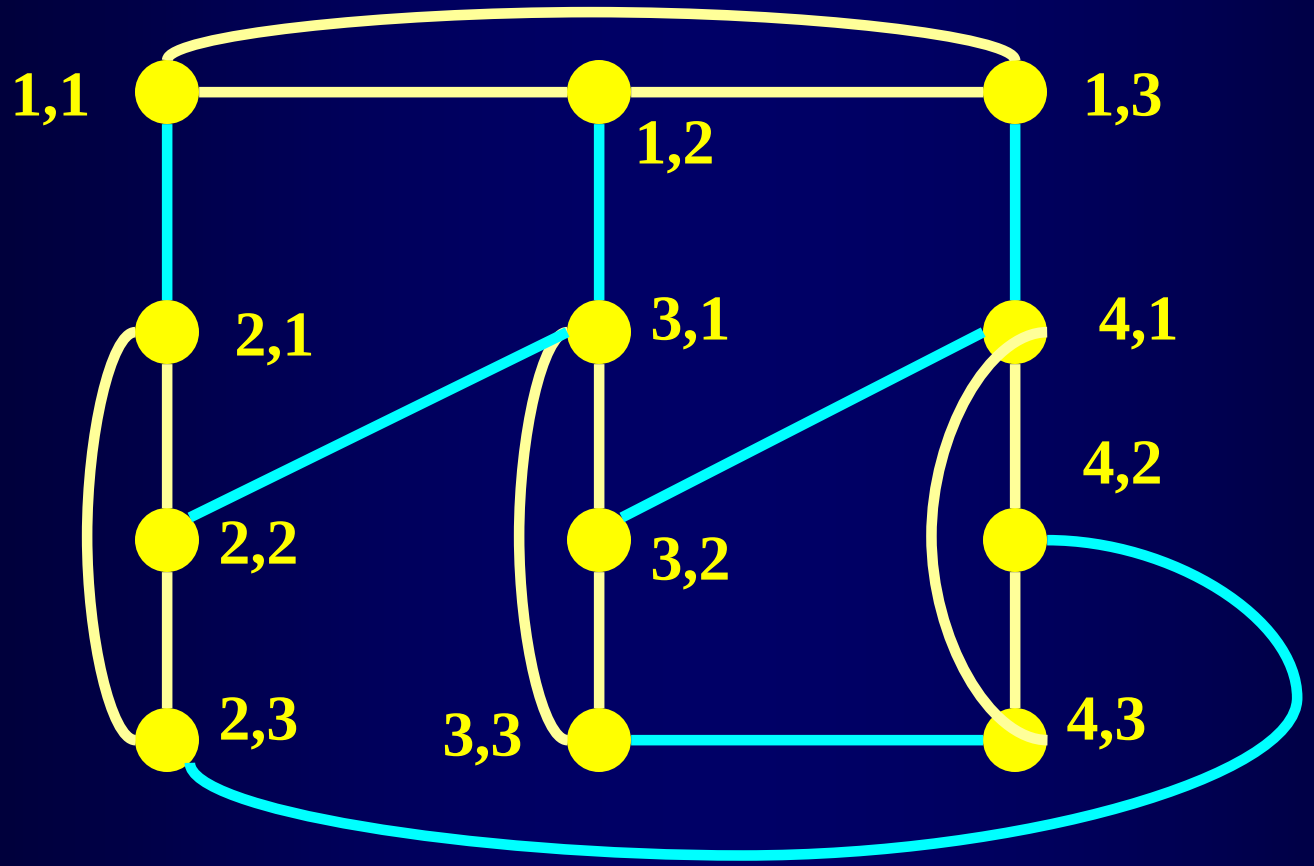
Le sommet (i,j), $1 \leq i \leq q$, $1 \leq j \leq 3$, qui correspond au littéral $l_{i,j}$ sera relié au sommet (m,n) si $i=m$ et $j \neq n$ ou si $i \neq m$ et $l_{i,j} = \neg l_{m,n}$.

La valeur de K sera $2q$.

La transformation se fait en temps polynomiale.

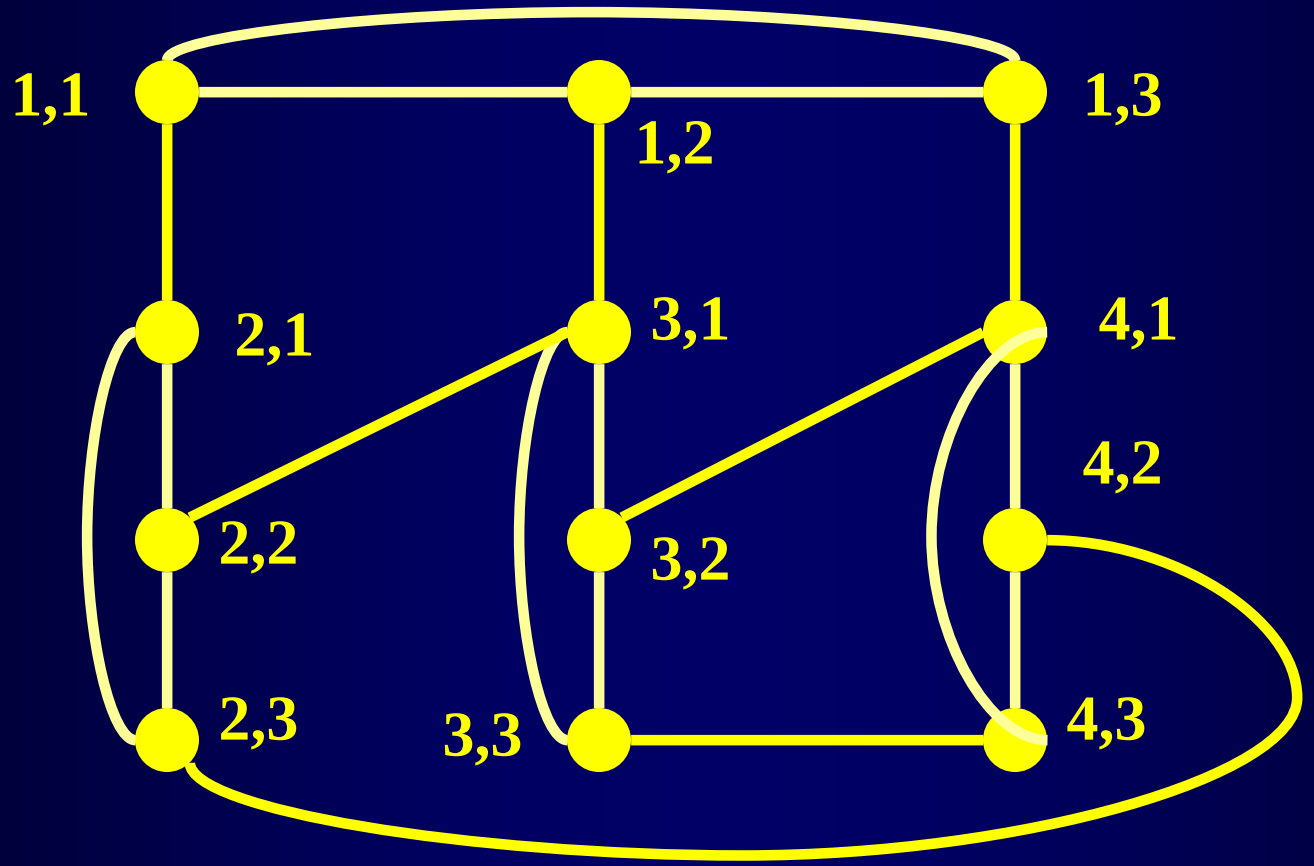
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



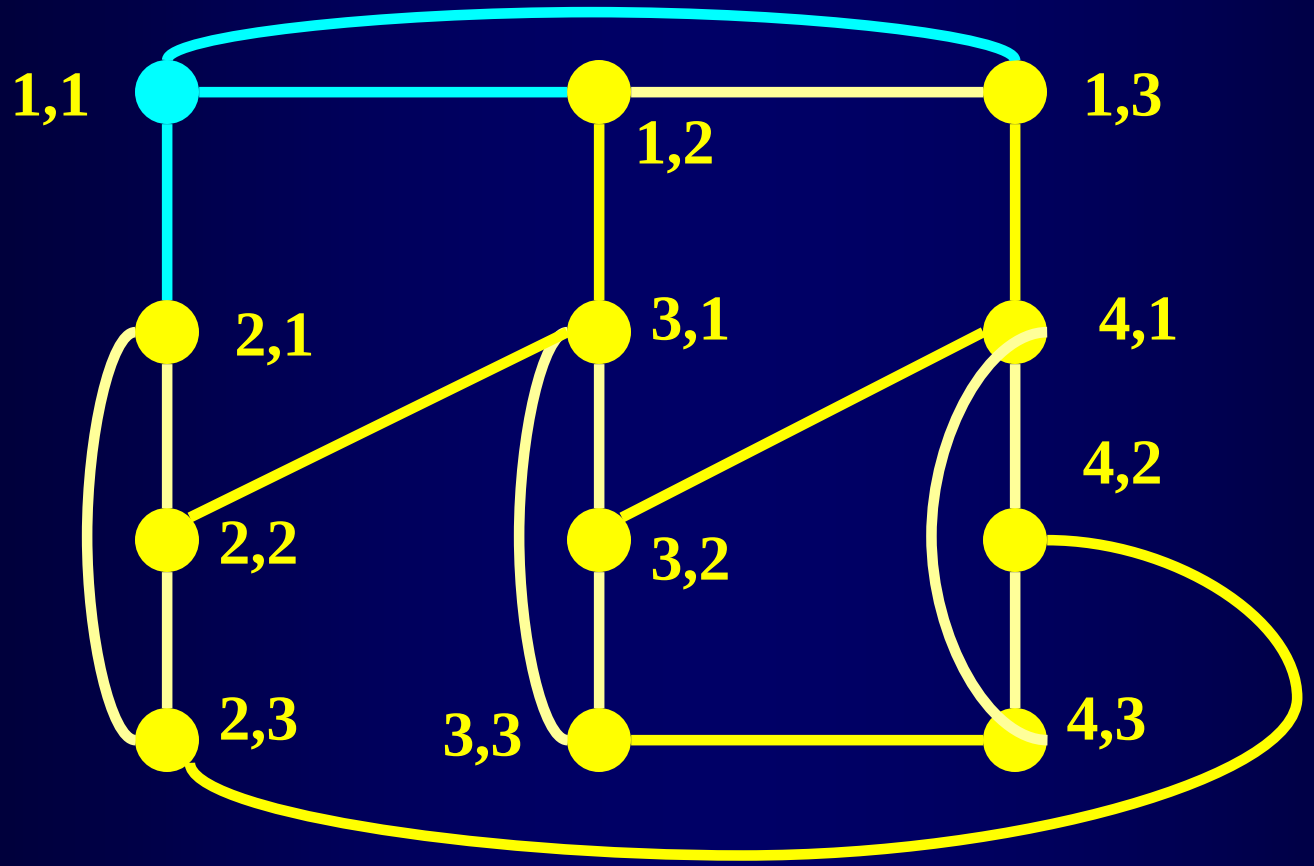
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



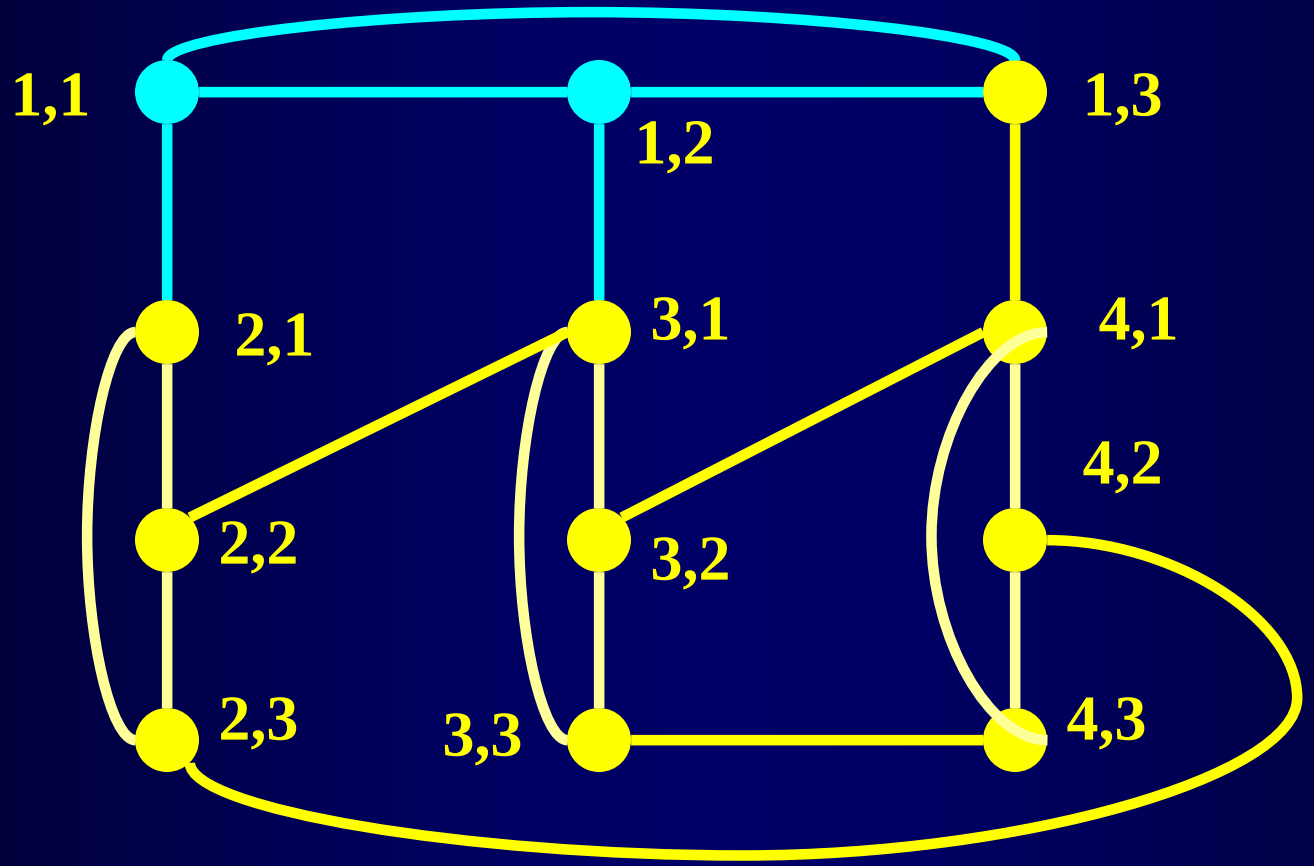
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



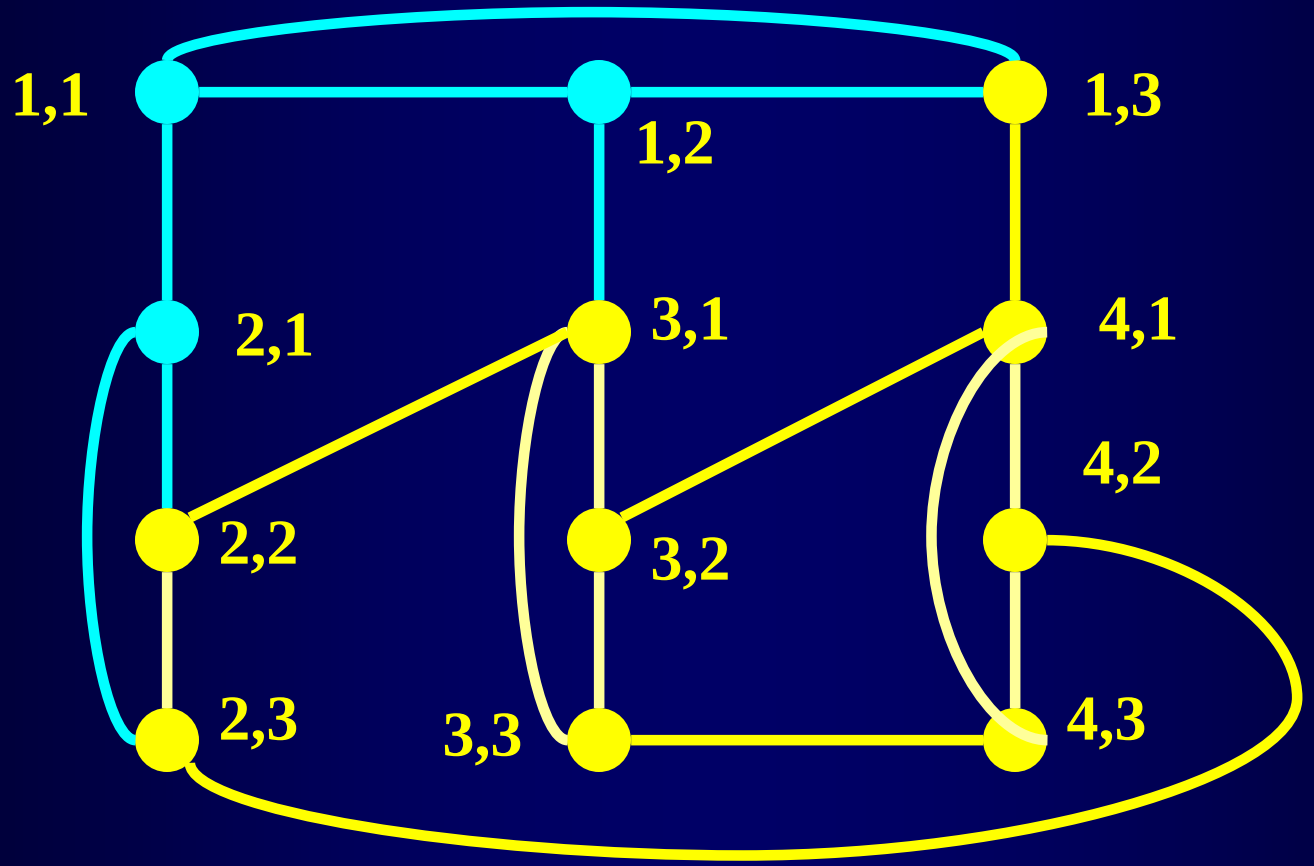
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



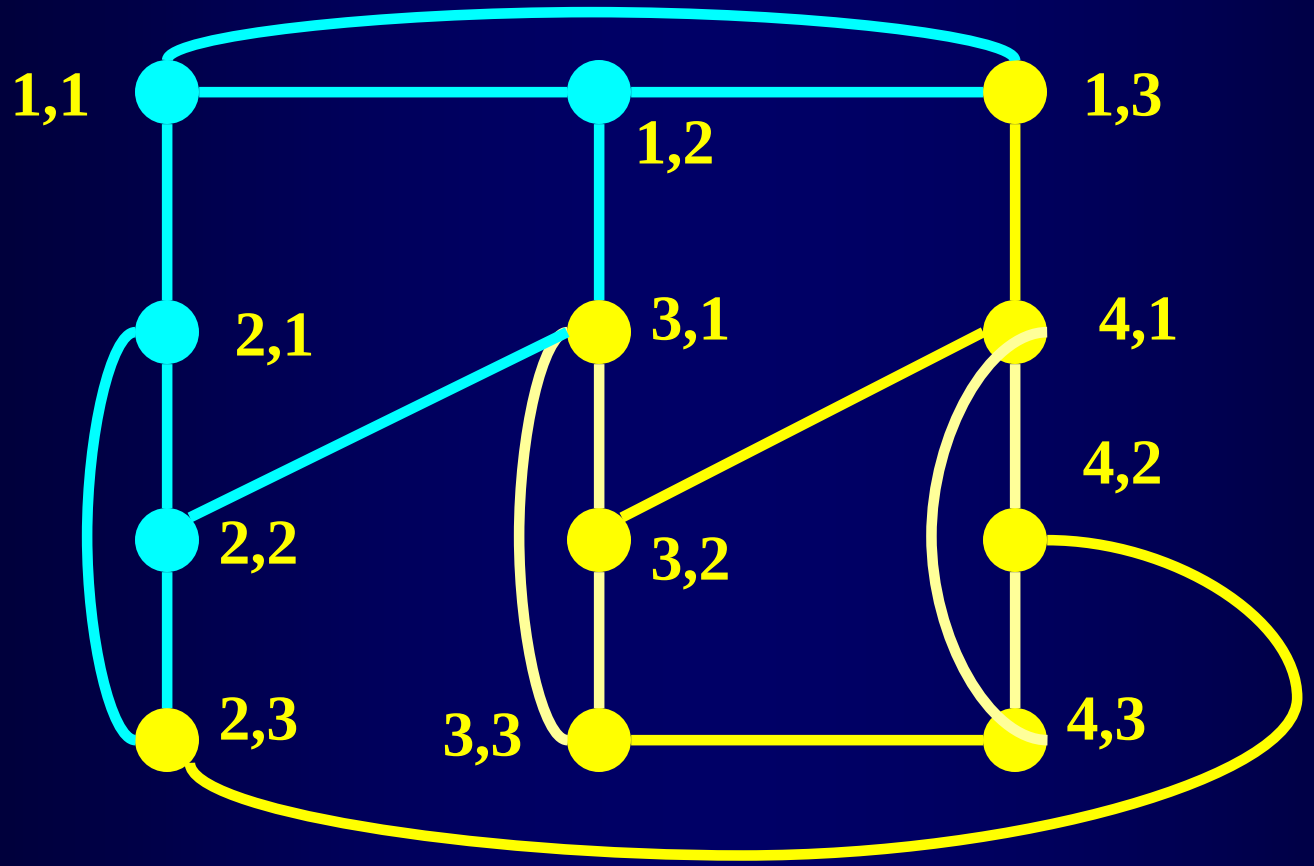
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



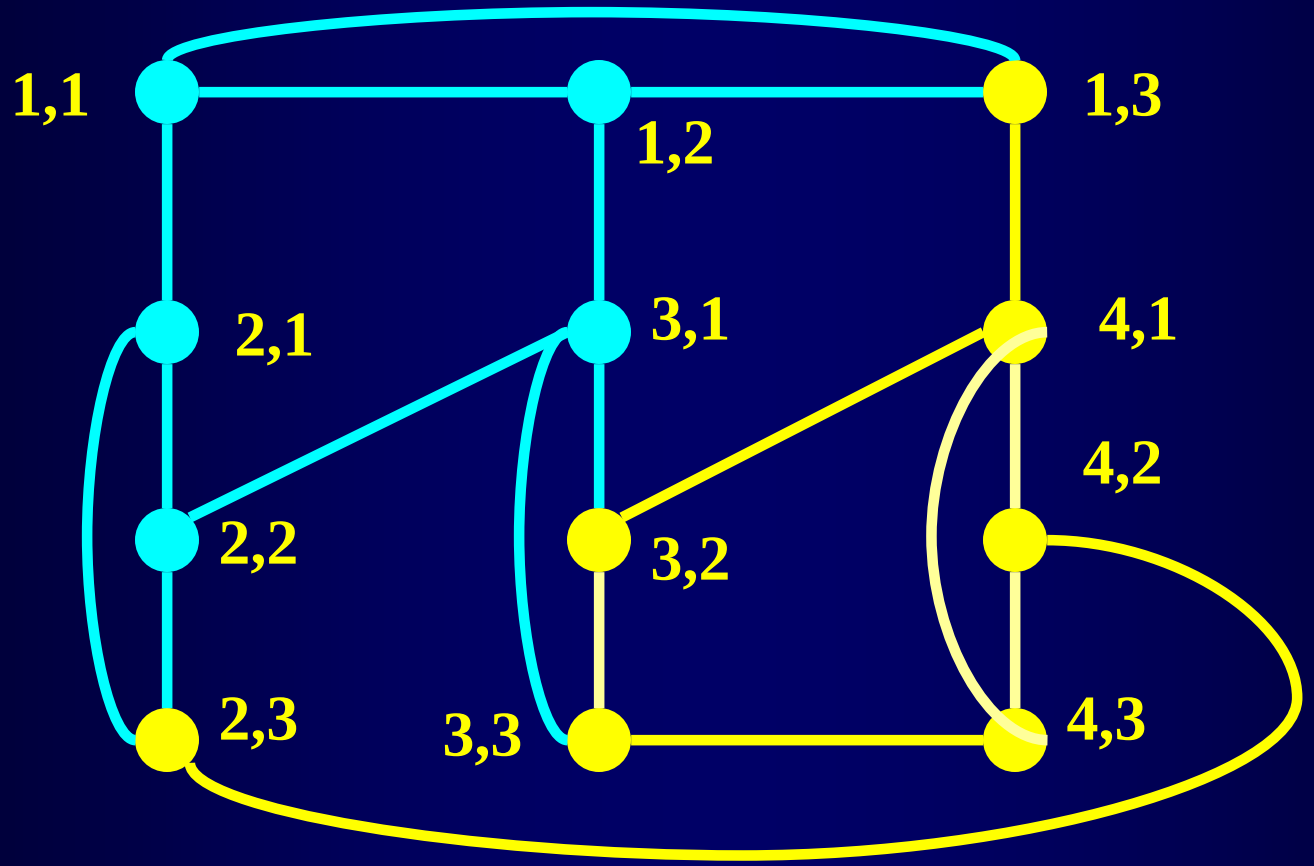
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



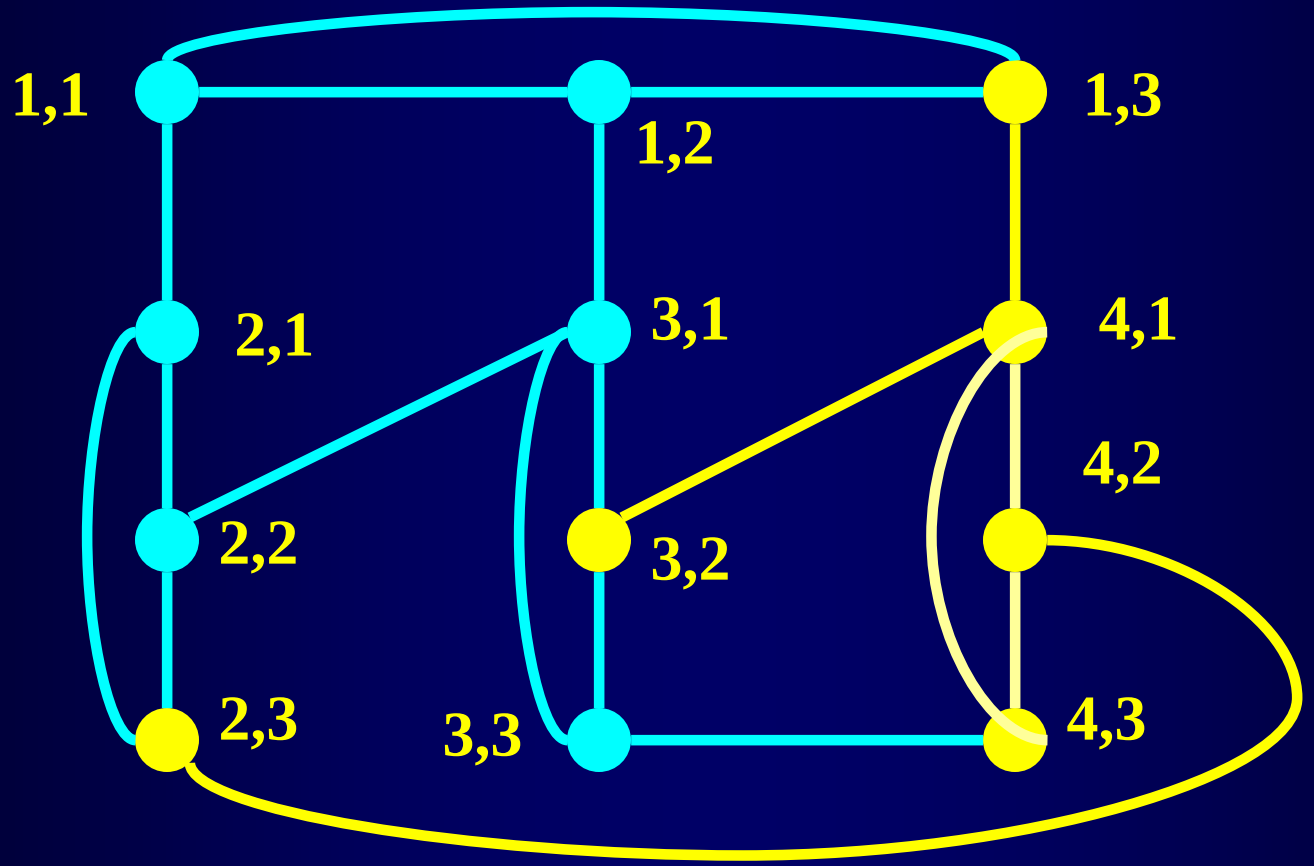
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



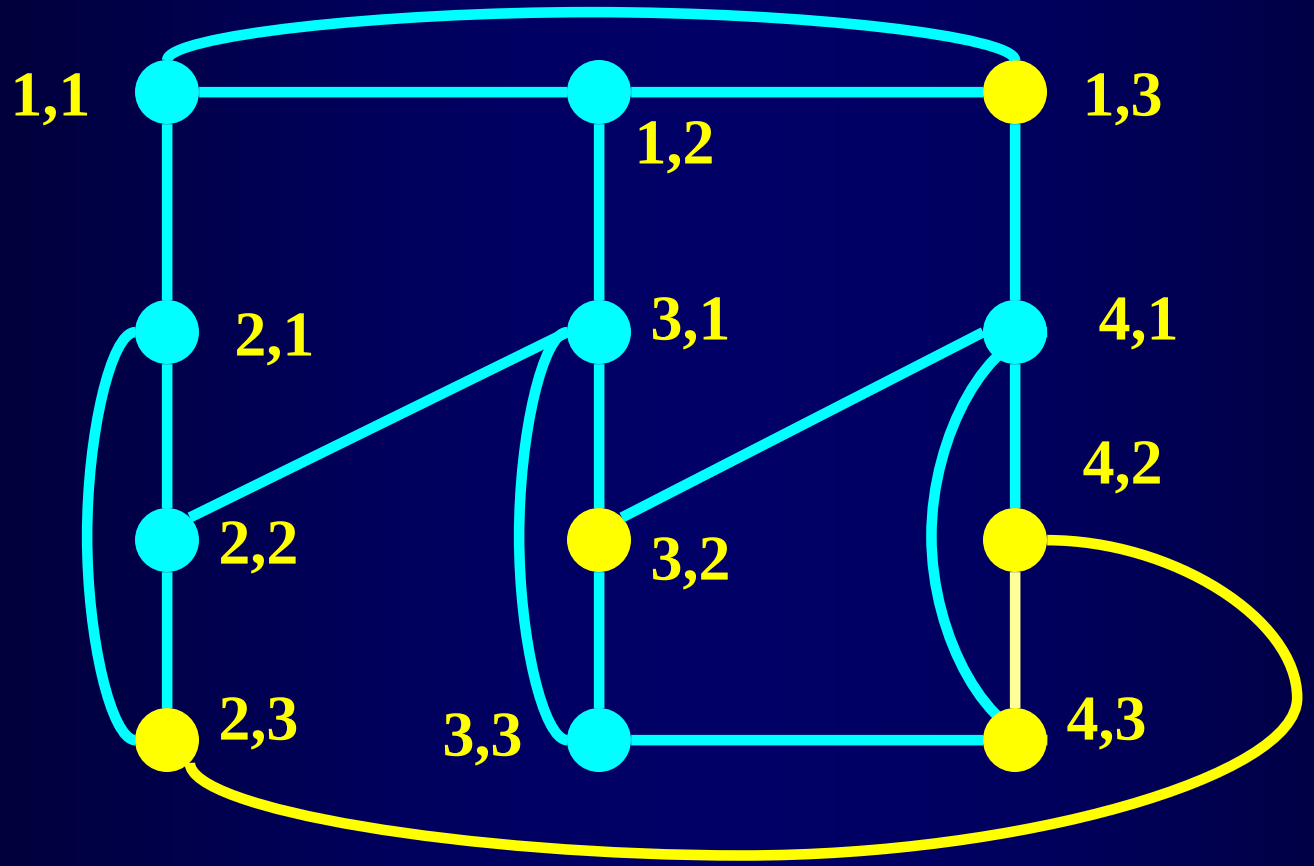
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



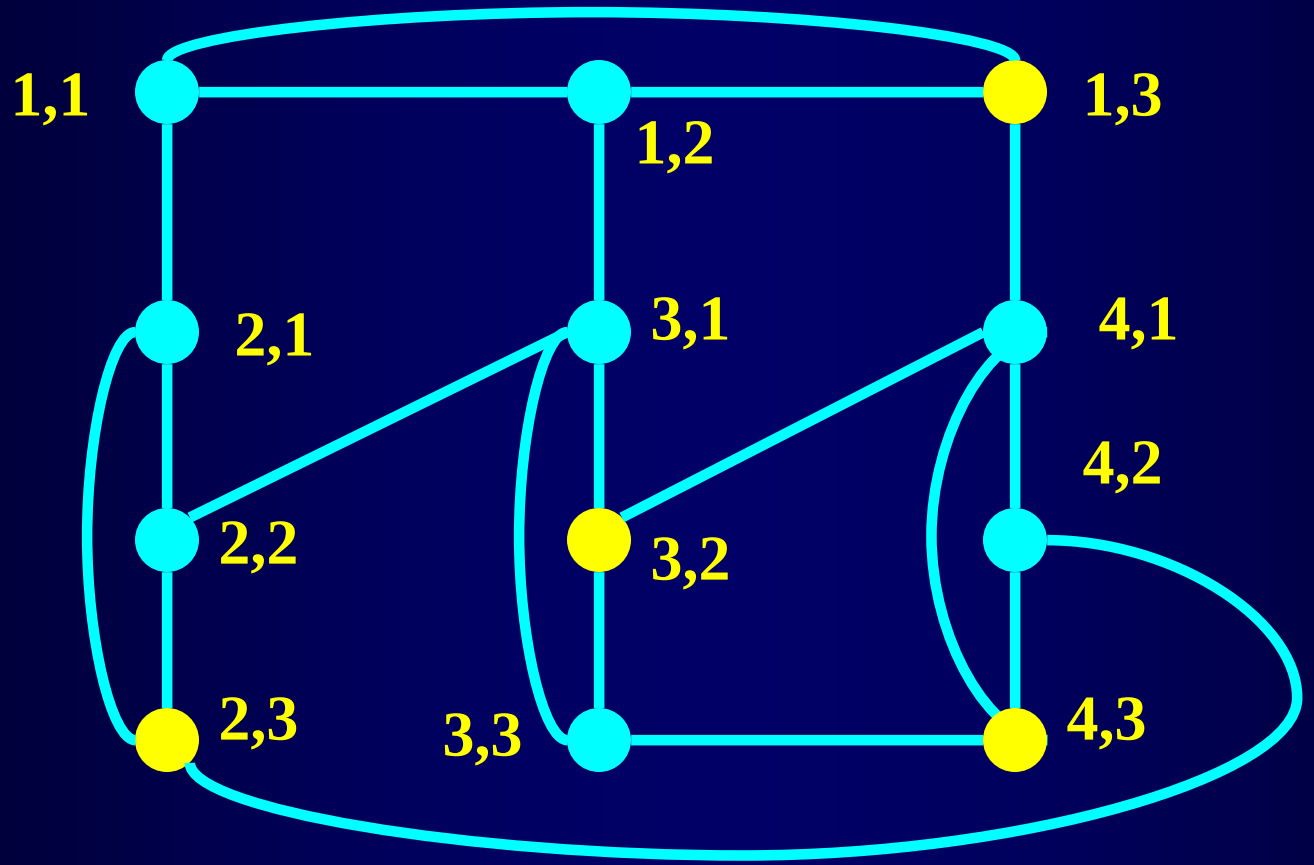
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



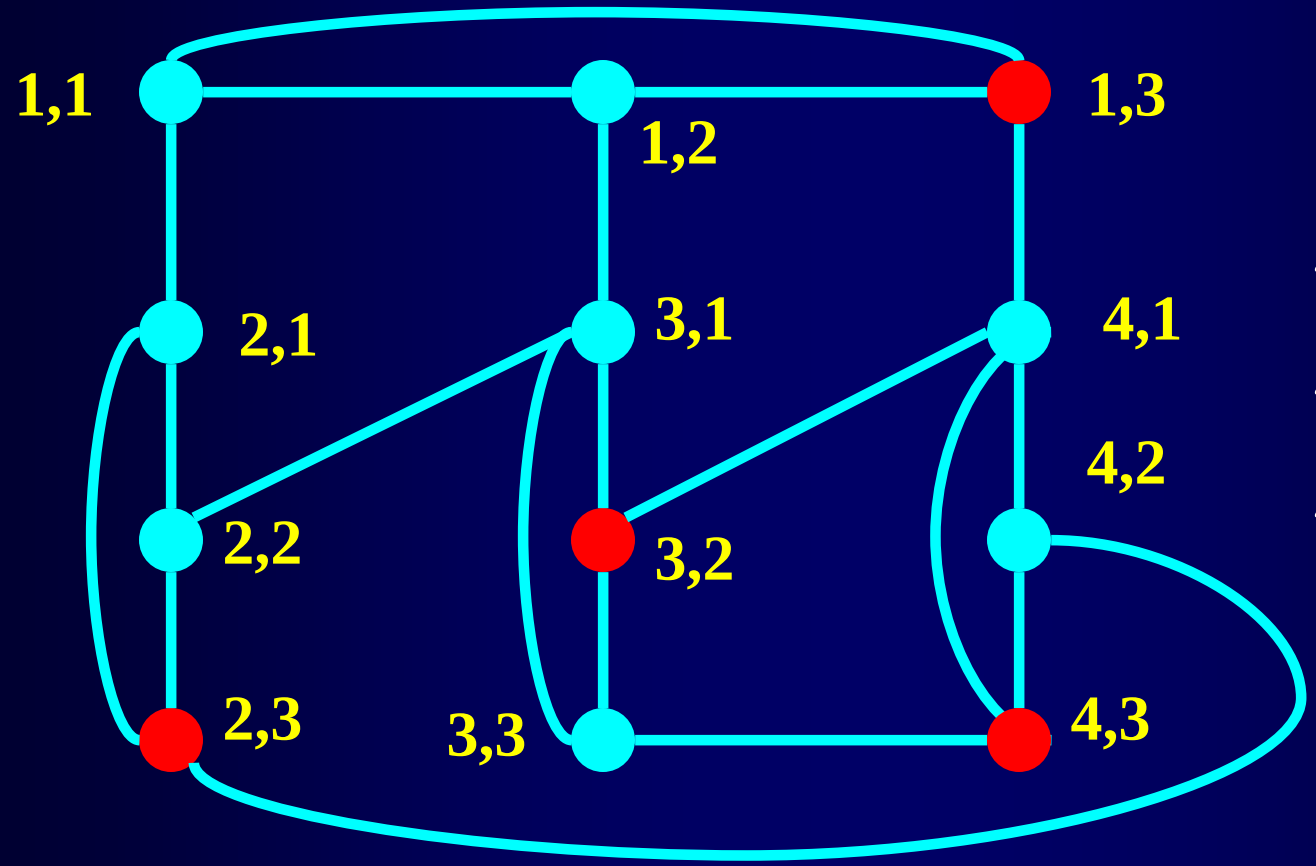
Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



Exemple : $F = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5)$

K=8



$x_3 = V$

$x_4 = V$

$x_5 = F$

OUI $\Rightarrow$ OUI :

- On choisit un littéral vrai par clause, le reste dans le transversal.
- Pas de contradiction, donc c'est un transversal.

OUI $\Leftarrow$ OUI :

- S'il existe un transversal, il doit contenir au moins deux (exactement !) sommets de chaque triangle.
- Si les littéraux hors transversale sont vrais, alors la formule est satisfaite.
- Peut pas y avoir de la contradiction.

Un autre exemple (feuille 3, exo 1)

Nom : Somme minimum des carrés (SMC)
Instance : A un ensemble fini d'entiers, $k, J \in \mathbb{N}$
Question : Existe-t'il une partition de $A = A_1 \cup \ldots \cup A_k$ telle que : $\sum_{i=1}^{k} \left(\sum_{a \in A_i} a\right)^2 \leq J$

SMC ∈ NP

Machine non-déterministe :

- **On parcourt l'ensemble A et on attribue chaque entier à un des ensembles A_k.**
- **On vérifie de manière déterministe qu'on a bien une partition de A, puis que la somme des carrés dans chaque ensemble ne dépasse pas J.**

NP-difficulté

Partition $\propto$ SMC

Soit A une instance de Partition. Pour SMC on utilisera le même ensemble, avec k=2 et $J=S^2/2$ où S est la somme des éléments de A.

Oui $\Rightarrow$ Oui

L'ensemble A admet une partition en deux parties de somme égales, A_1 et A_2, ainsi la somme des éléments de chaque ensemble est S/2, donc le carré est $S^2/4$ et donc la somme des carrés est

$$2\ S^2/4 = S^2/2 = J.$$

Oui ⇐ Oui

Soient A_1 et A_2 une partition qui vérifie

$$\sum_{i=1}^{k} \left(\sum_{a \in A_i} a \right)^2 \leq J$$

Soit $S/2+x$ la somme des éléments de A_1 et ainsi la somme des éléments de A_2 est $S/2-x$.

La somme des carrés est donc

$(S/2+x)^2+(S/2-x)^2 = S^2/4+Sx+x^2+S^2/4-Sx+x^2 =$

$S^2/2+2x^2$.

Si $S^2/2+2x^2 \leq S^2/2 = J$ c'est que $x^2 \leq 0$, donc $x=0$ et on avait une partition en deux parties égales.

On vient de montrer que le problème Partition est équivalent au SMC avec $k = 2$, donc un cas particulier du problème SMC avec k quelconque.

Exercice

Mettre le problème suivant dans la forme usuelle et prouver sa NP-complétude

Chevaliers de la table ronde :

Etant donnés *n* chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ?

Enoncé

Nom : Chevaliers de la table ronde (CTR)

Données : Un ensemble de *n* chevaliers et une relation entre les chevaliers, nommé « chevaliers féroces ennemis », donné par liste de paires. Une table ronde de *n* places.

Question : Peut-on placer les chevaliers à la table de telle manière que paire de féroces ennemis ne soit pas côte à côte ?

CTR $\in$ NP

Une prétendue solution est une liste des chevaliers assis au places 1, 2, …, *n*.

Sa taille est polynomiale.

En temps polynomiale on peut vérifier que (*i*, *i*+1) ne sont pas des féroces ennemis (pour *i* = 1, 2, *n*-1), ni (*n*, 1).

Ainsi CTR $\in$ NP.

CTR est NP-difficile

On réduit Cycle Hamiltonien à CTR.

La transformation : chaque sommet du graphe devient un chevalier et si deux sommets ne sont pas reliés alors les chevaliers correspondants sont des ennemis féroces. La taille de la table sera le nombre de sommets dans le graphe.

C’est une transformation polynomiale.

OUI ⇒ OUI

Si le graphe admet un cycle hamiltonien, alors on a une suite $x_1, x_2, \ldots, x_n$, tel que (x_1, x_2) ne sont pas ennemis, (x_2, x_3) non plus et ainsi de suite (x_{n-1}, x_n) non plus et (x_n, x_1) non plus.

On peut donc conclure que l'ordre $x_1, x_2, \ldots, x_n$ est une solution pour CRT, donc la réponse est OUI.

OUI $\Leftarrow$ OUI

Si on a un ordre pour assoir les chevaliers x_1, x_2, …, x_n, alors dans le graphe il y a une arête entre les sommets x_1 et x_2, et aussi une arête entre x_2 et x_3 et ainsi de suite entre x_{n-1} et x_n aussi et entre x_n et x_1 aussi.

On peut donc conclure que x_1, x_2, …, x_n est un cycle hamiltonien dans le graphe, donc la réponse à CycleHam est OUI.

On modélise notre problème par un graphe :

- **les sommets – les chevaliers**
- **les arêtes – une arête entre deux sommets si les chevaliers correspondants ne seont pas de féroces ennemis**

Rappels
Flots

Un réseau est un graphe orienté G(V,E) avec pour chaque arc (u,v) une capacité $c(u,v) \geq 0$.

Deux nœuds spéciaux : s – source; t – puits

Pour tout sommet x du réseau il existe un chemin de s vers x et un de x vers t.

Un flot dans G est une fonction f : E $\rightarrow$ R :

- **contrainte de capacité**

  pour tout arc e : $0 \leq f(e) \leq c(e)$

- **conservation des flots (lois de Kirchhoff)**

La somme du flot de l'ensemble des arcs entrants est égal à la somme du flot des arcs sortants.

Doit être vérifié pour tous les sommets, sauf la source et le puits.

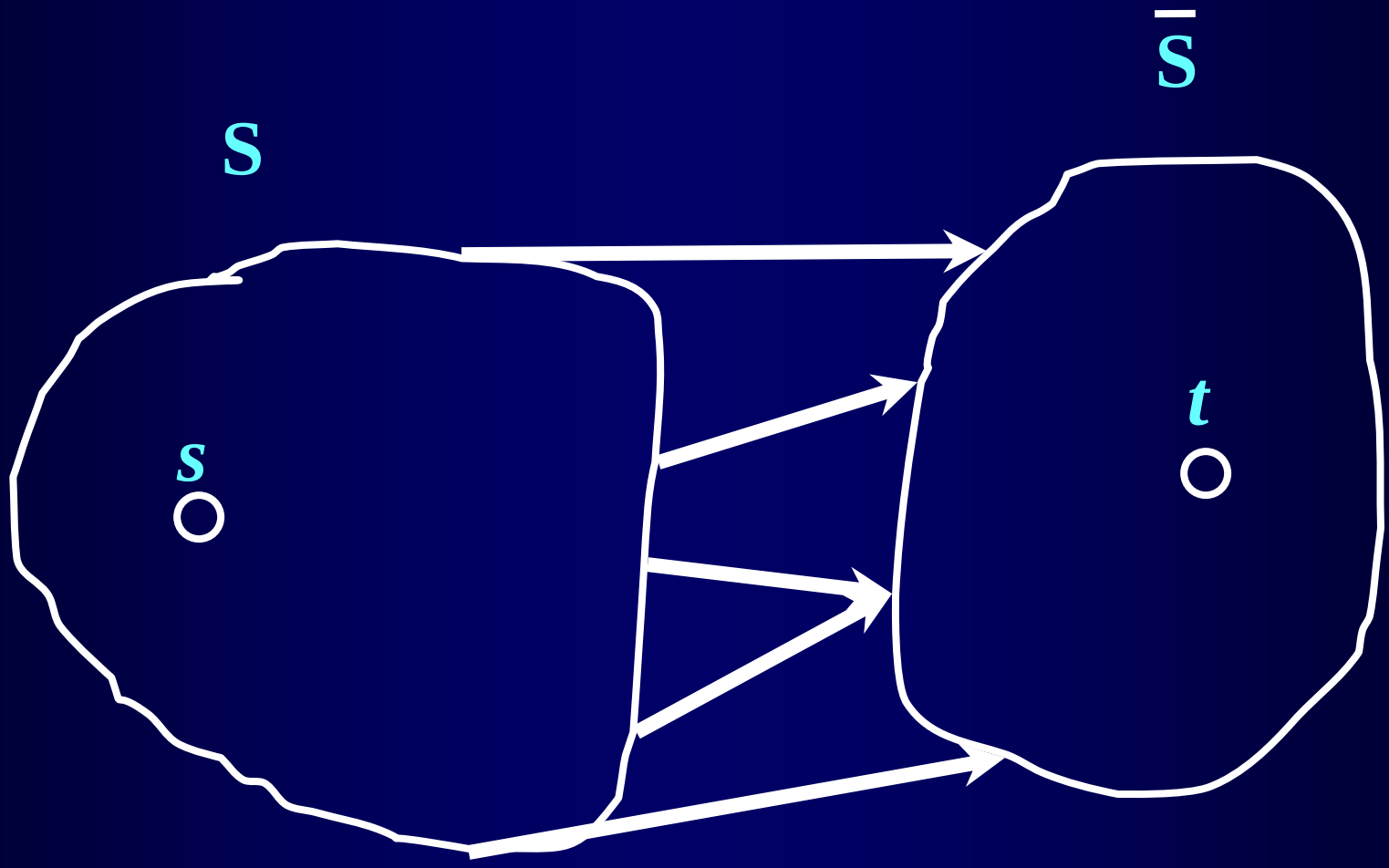
La valeur d'un flot

$$|f| = \Sigma_{e \in \mathrm{In}(t)} f(e) - \Sigma_{e \in \mathrm{Out}(t)} f(e)$$

Le problème du flot maximum :

trouver un flot de valeur maximum.

Une coupe



La capacité d'une coupe est la somme des capacités des arcs orientés dans le sens source vers puits de la coupe

Etant donné une coupe la valeur du flot peut être défini comme le somme des flots sur les arcsde la coupe dans le sens source vers puits moins la somme des flots dans le sens inverse

Ainsi, pour tout flot et toute coupe, la valeur du flot est bornée par la capacité de la coupe. En cas d'égalité on a un flot maximum.

Une chaîne augmentante est un chaîne entre *s* et *t* sur laquelle on peut augmenter le flot, c.a.d. une chaîne telle que

- **les arcs "avant" (orientés dans le sens source vers puits) le flot est inférieur à la capacité**
- **les arcs "arrière" (orientés dans le sens puits vers source) le flot est non nul**

Augmenter le flot : augmenter sur les arcs avant et diminuer sur les arcs arrière

Le théorème Flot-max Coupe-min

Soit *f* un flot dans un réseau G(V,E) de source *s* et puits *t*. Les conditions suivantes sont équivalentes :

1. ***f* est un flot maximum**
2. **le réseau résiduel G_f ne contient pas de chaîne augmentante**
3. **$|f| = c(S,S')$ pour une coupe (S,S') de G**

Méthode Ford & Fulkerson

tant qu'il existe une chaîne augmentante

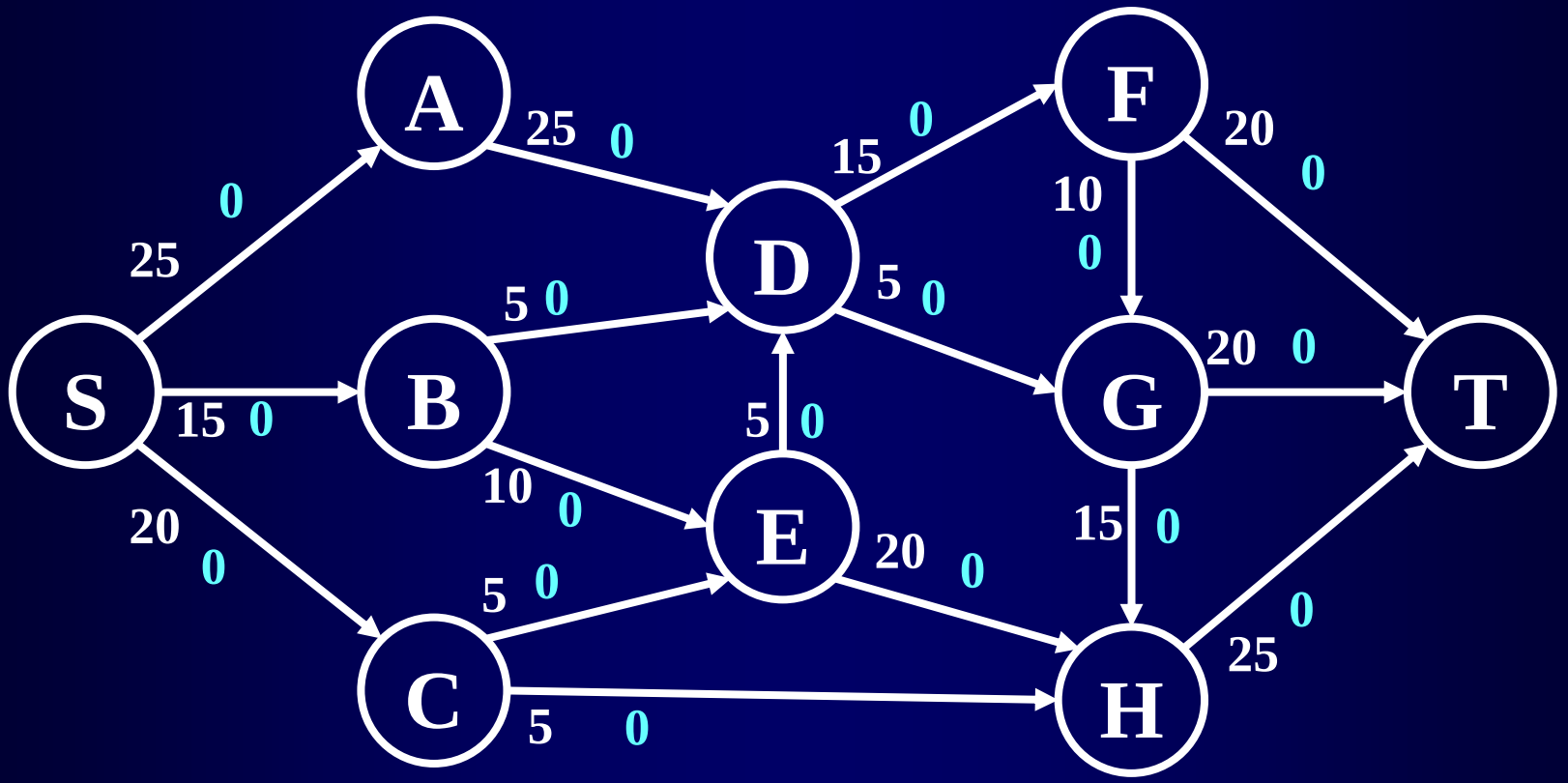
faire

- **chercher une chaîne augmentante**
- **augmenter le flot sur la chaîne**

fait

Exemple (feuille 4, exo 1)

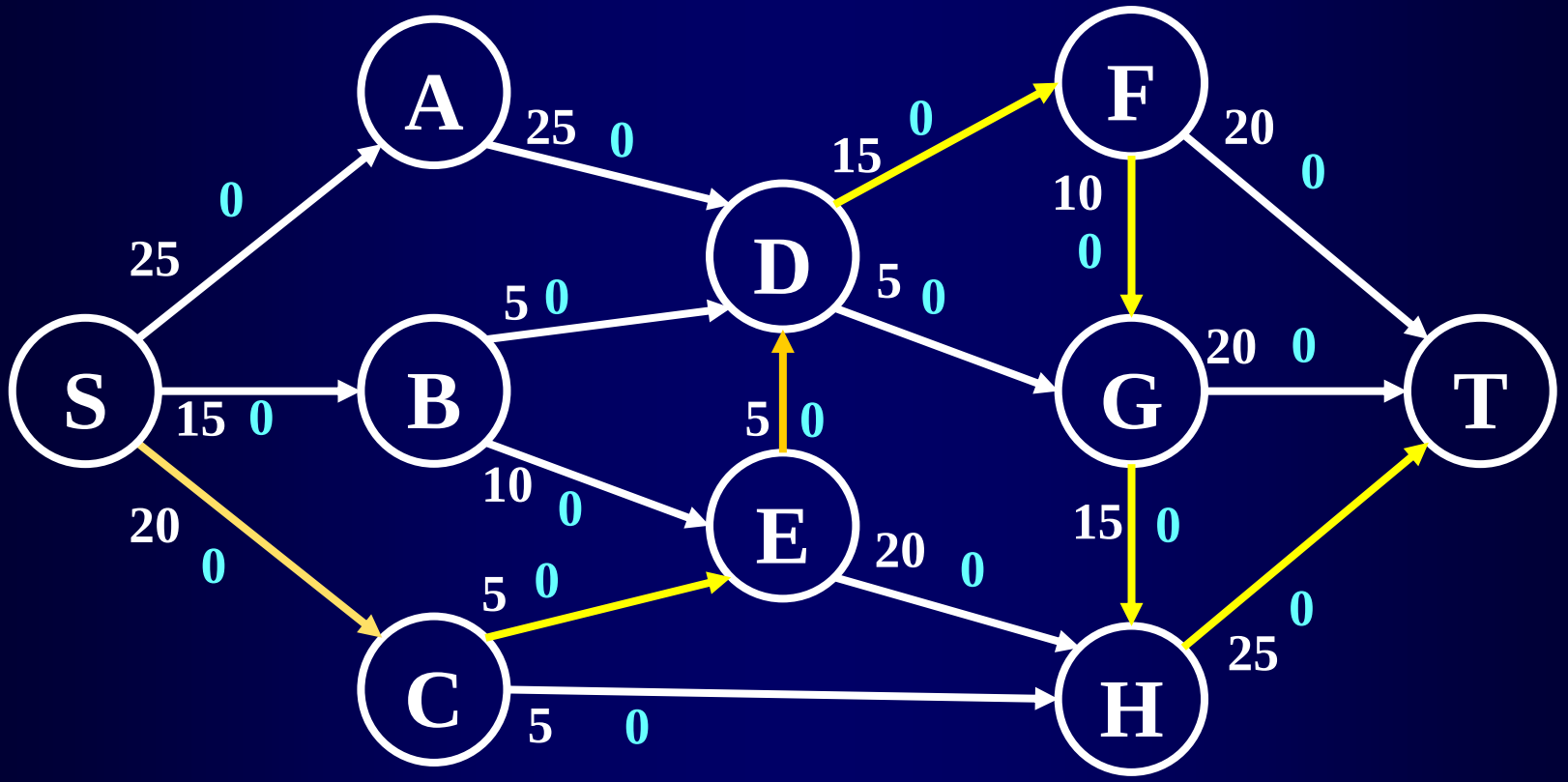
Valeur du flot = 0



Flot initial (nul)

c f

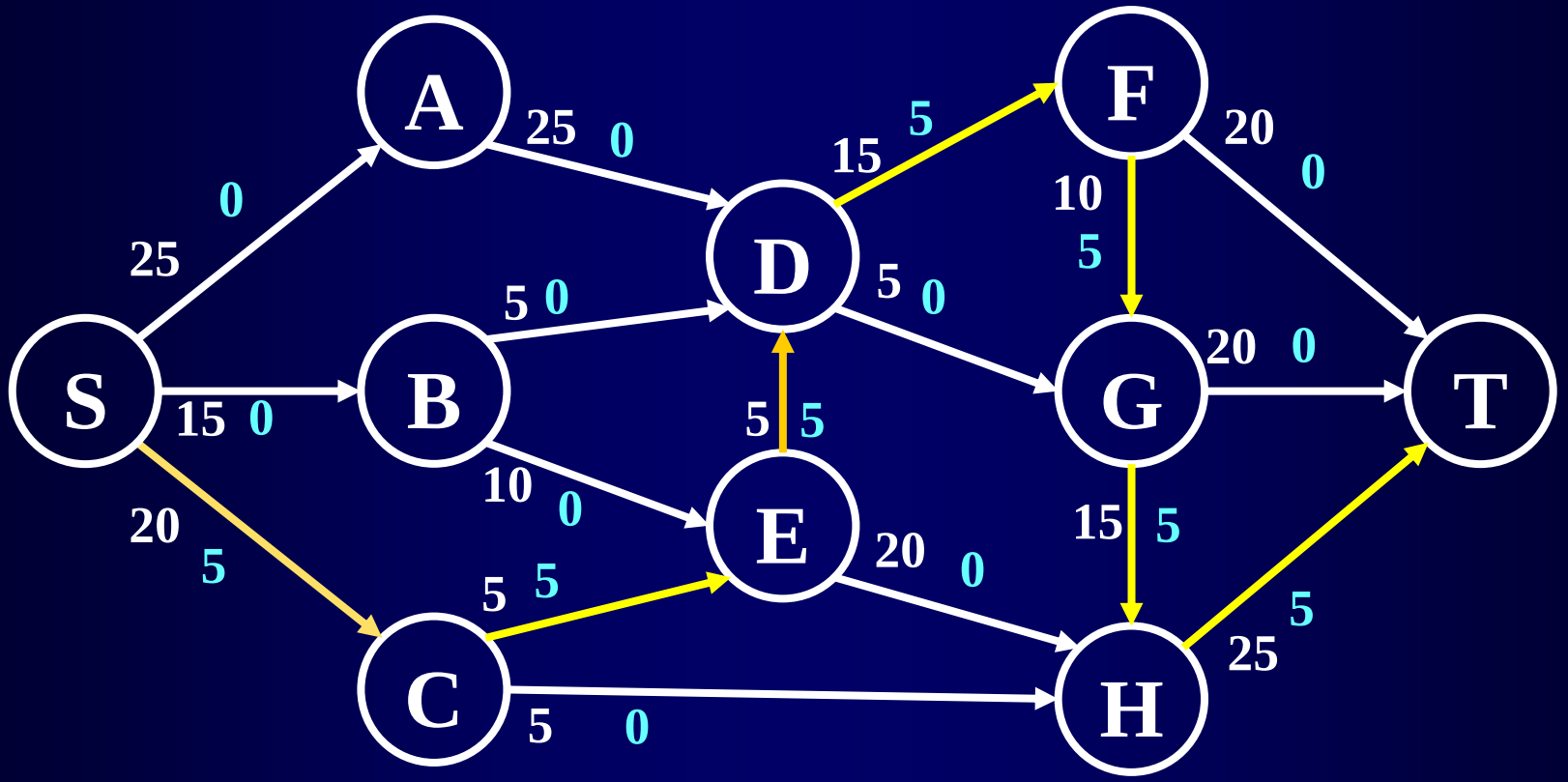
Valeur du flot = 0



Chaîne augmentante

c f

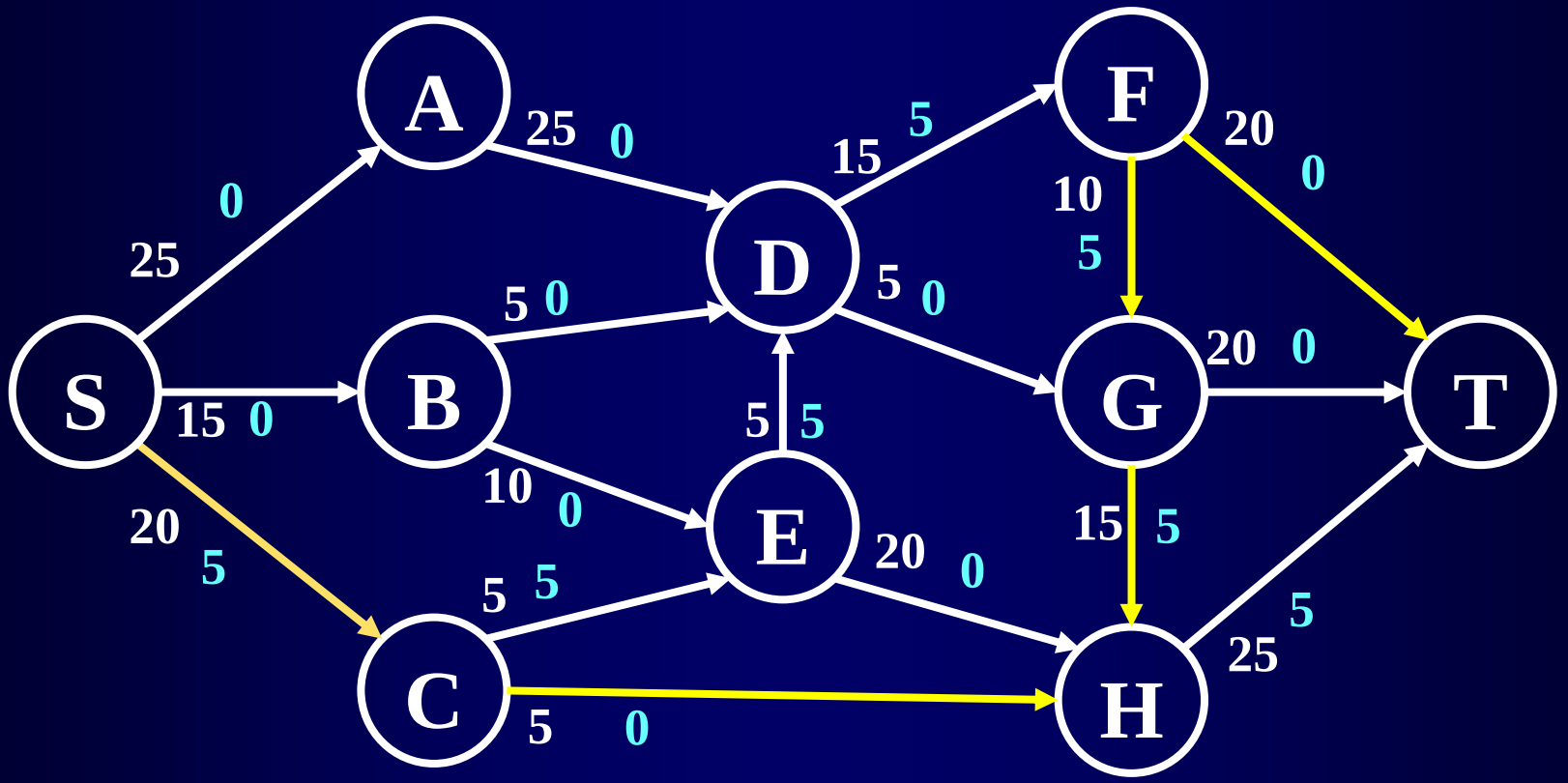
Valeur du flot = 5



Augmentation

c f

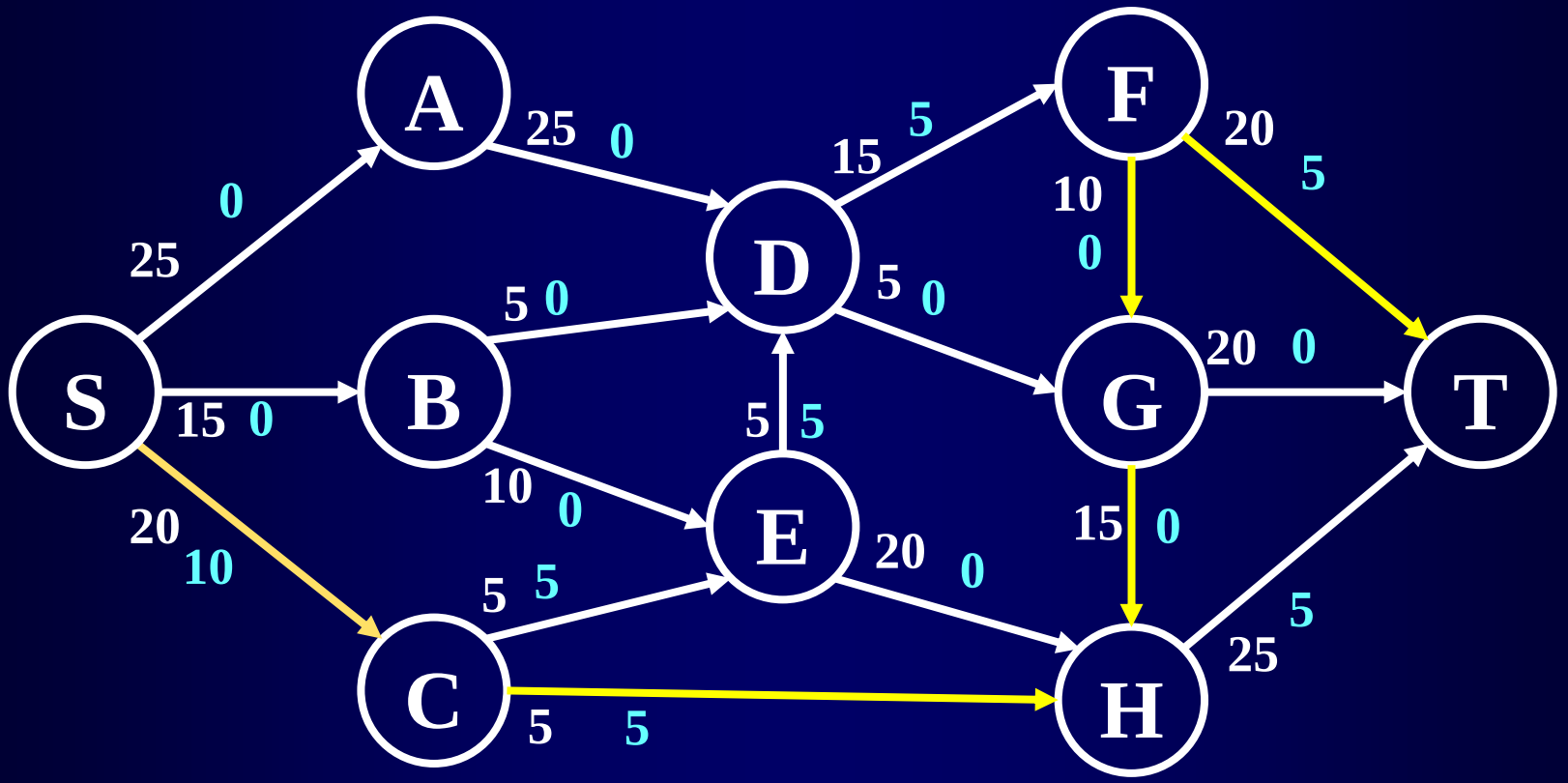
Valeur du flot = 5



Chaine augmentante

c f

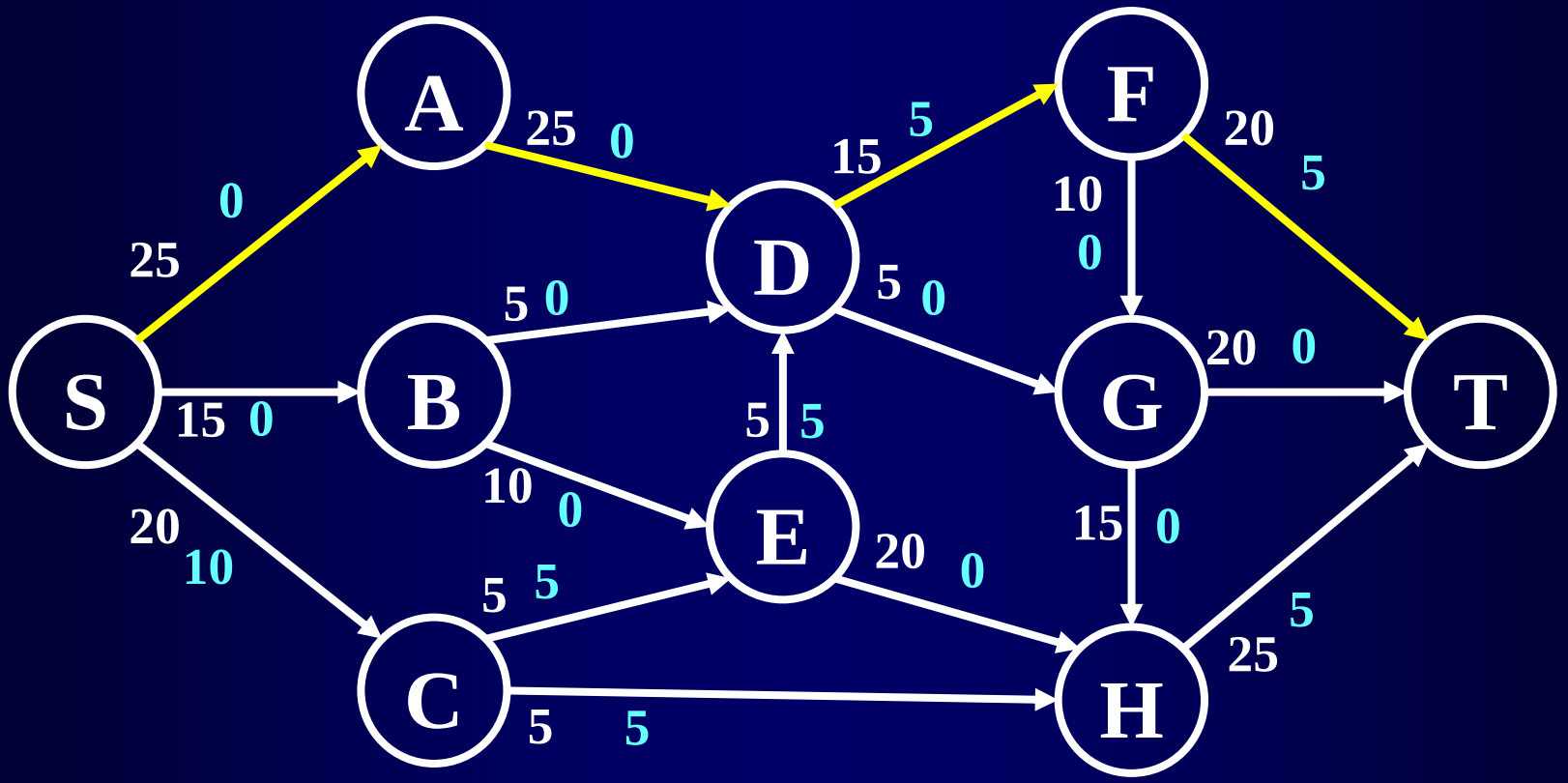
Valeur du flot = 10



Augmentation

c f

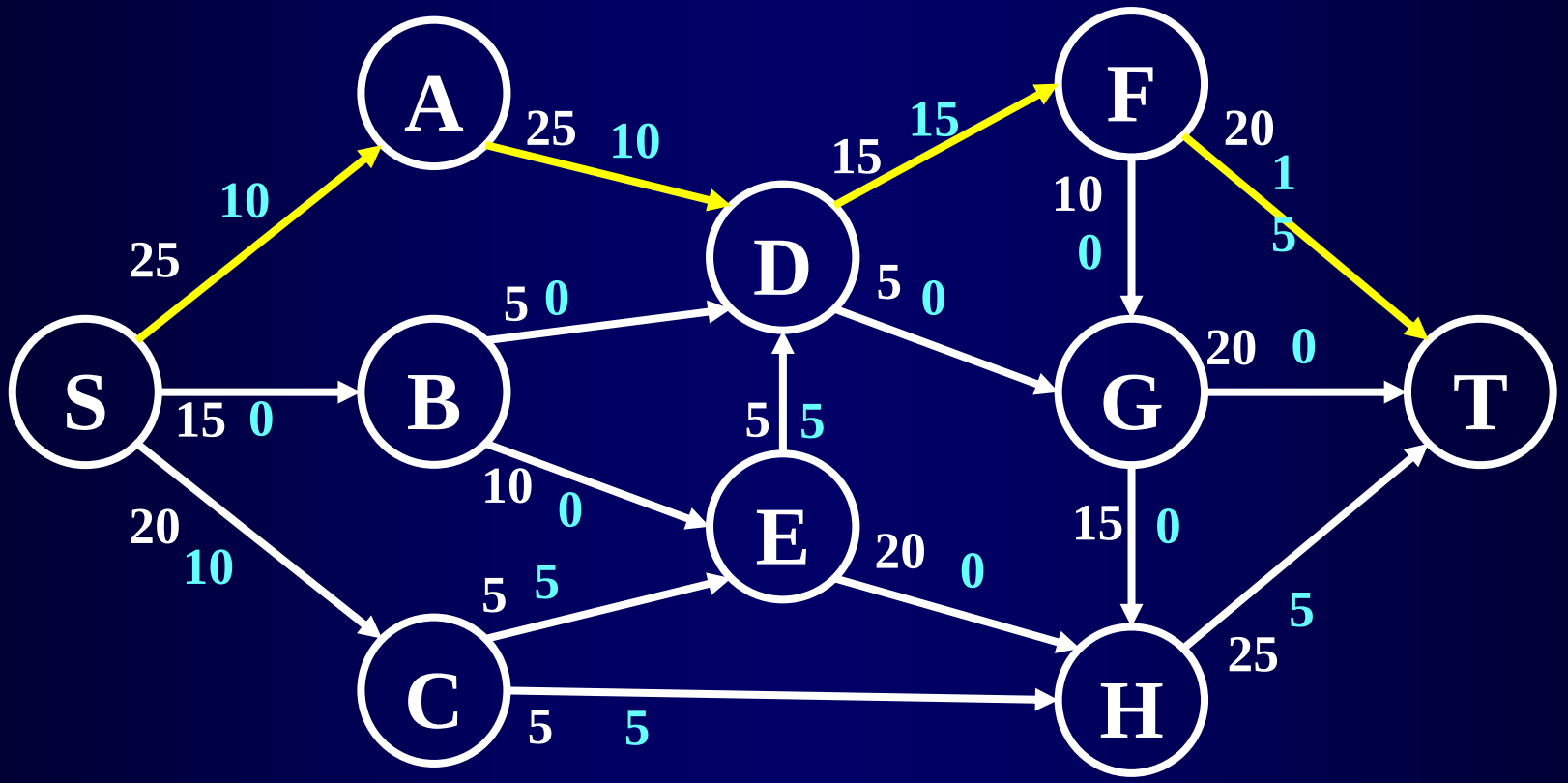
Valeur du flot = 10



Chaine augmentante

c f

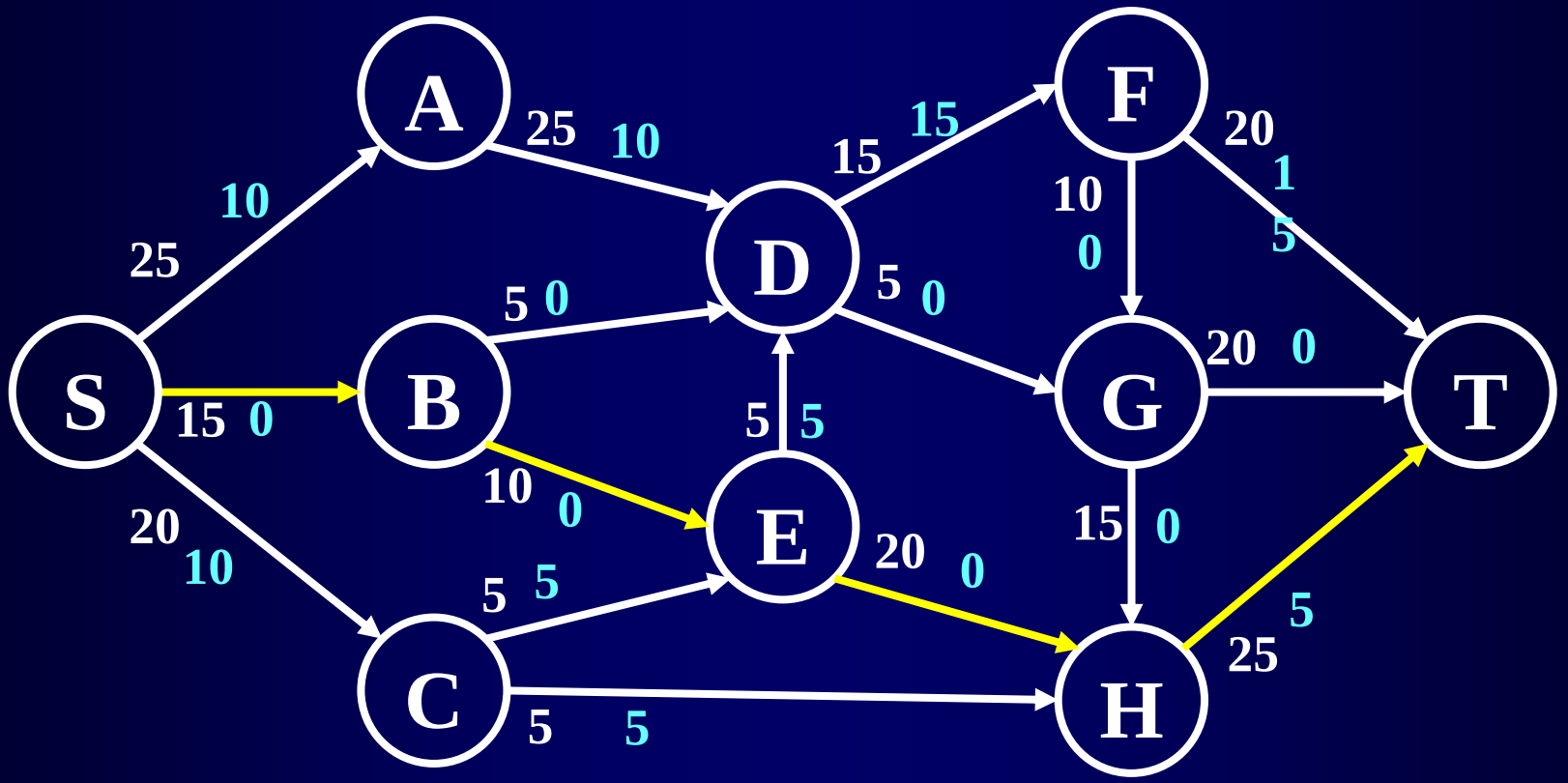
Valeur du flot = 20



Augmentation

c f

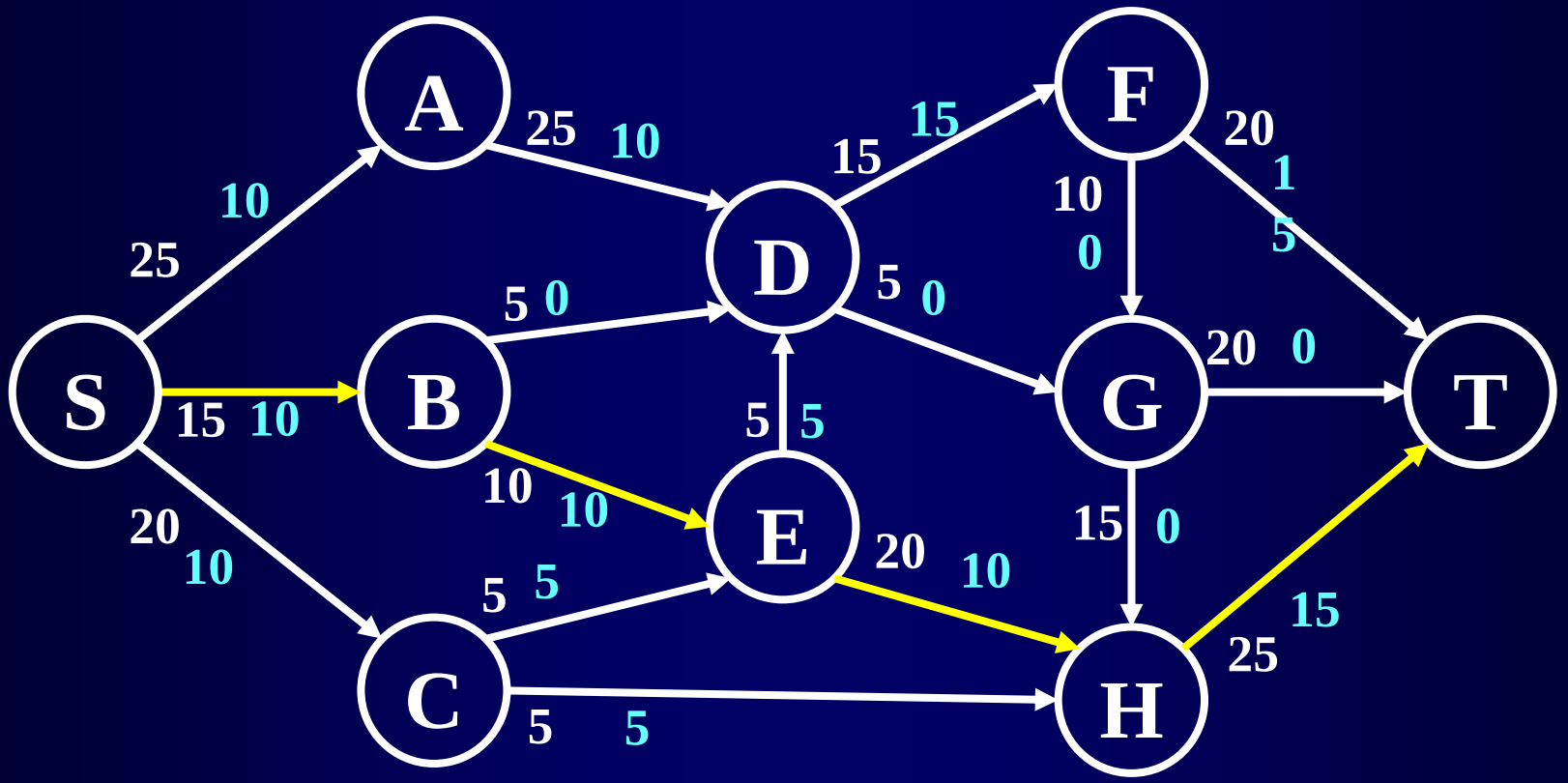
Valeur du flot = 20



Chaine augmentante

c f

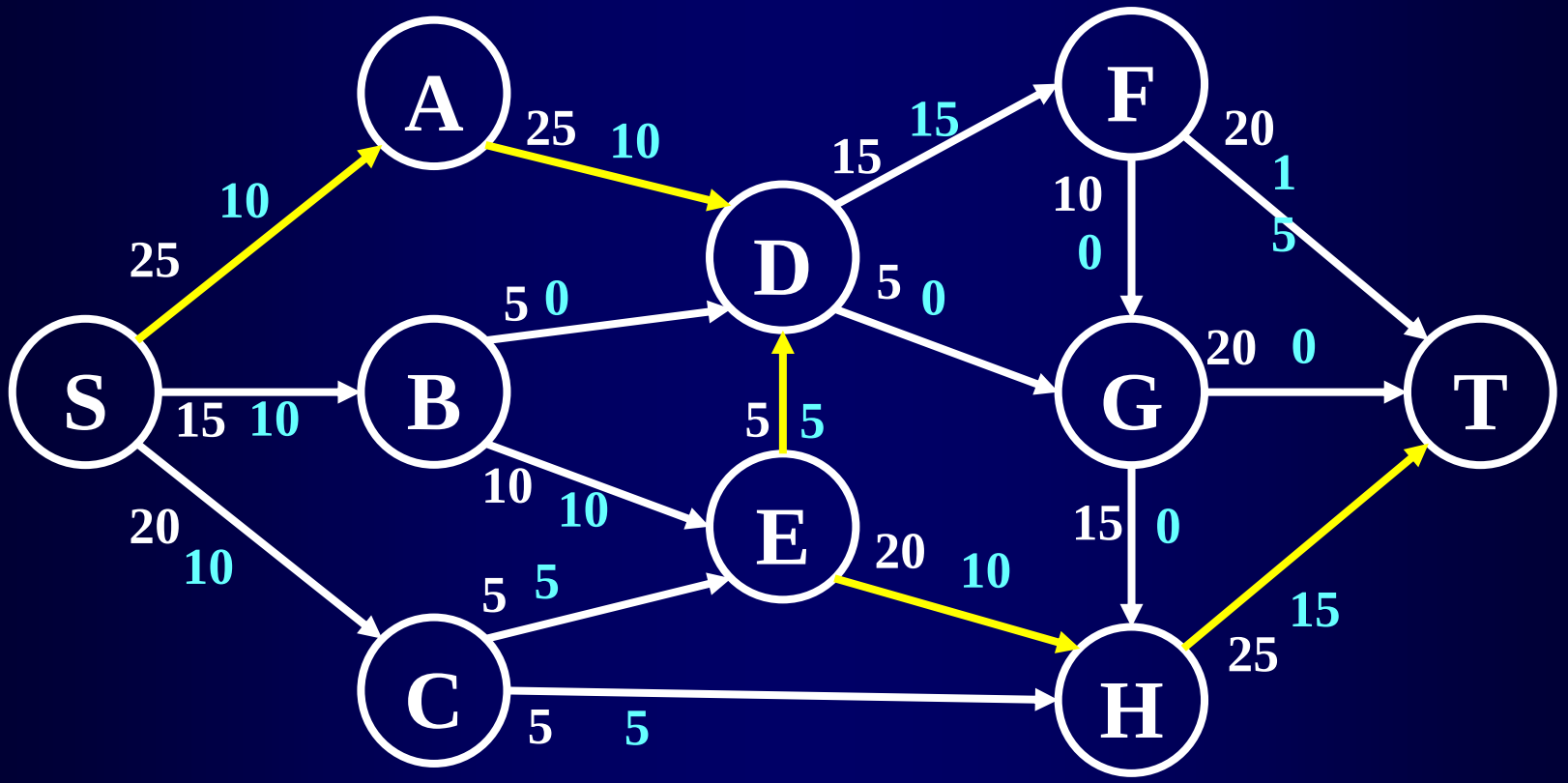
Valeur du flot = 30



Augmentation

c f

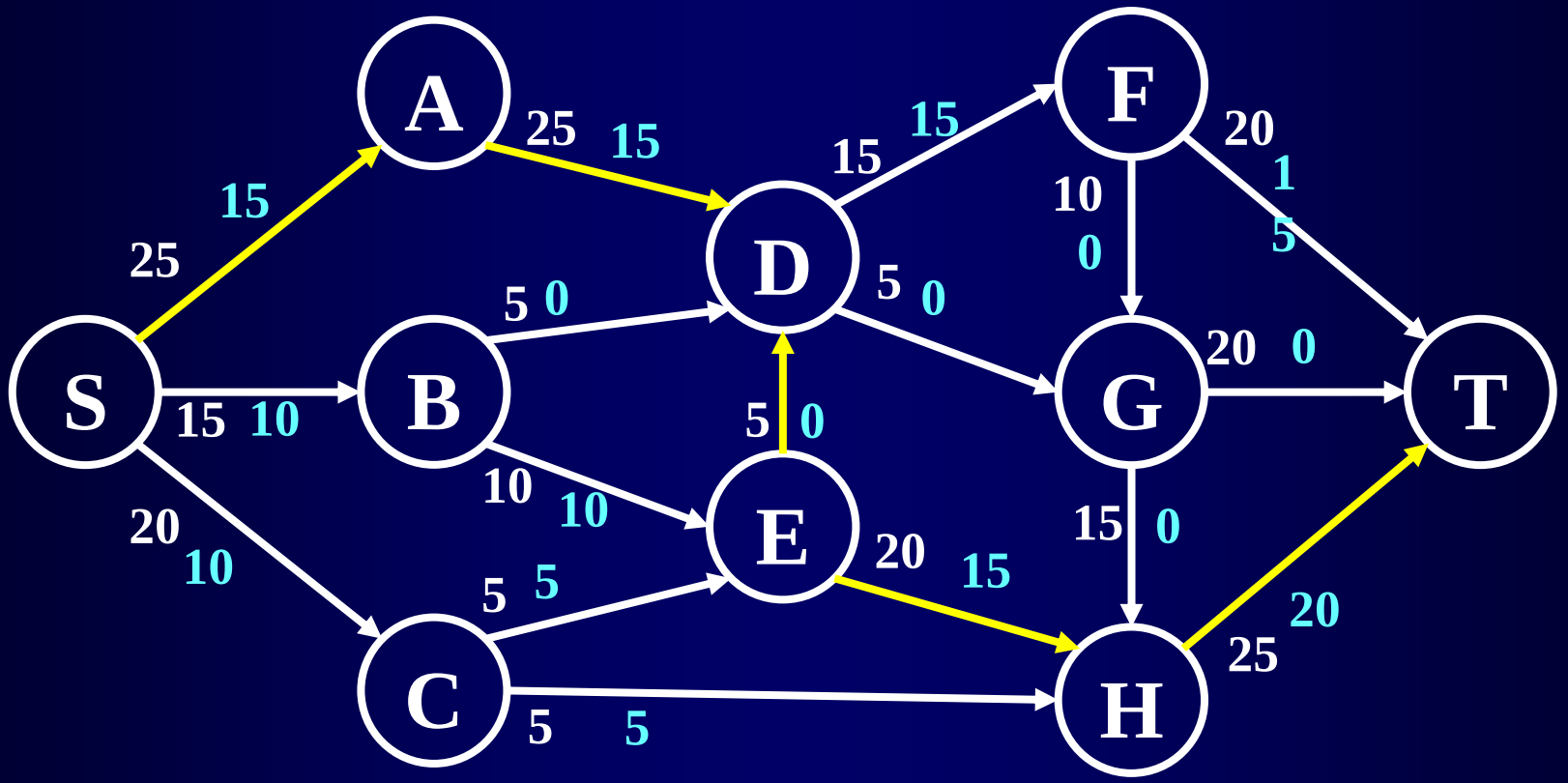
Valeur du flot = 30



Chaine augmentante

c f

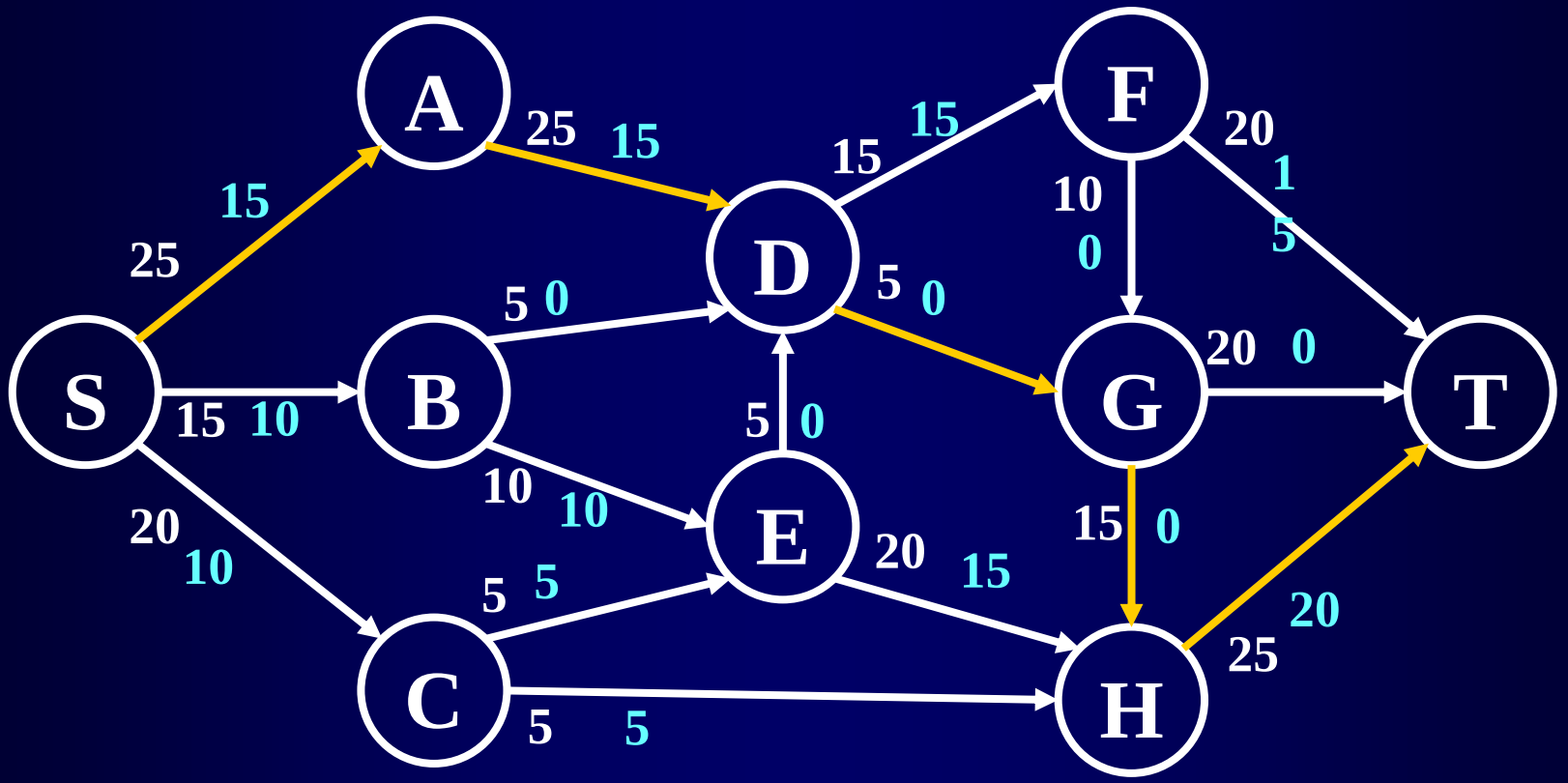
Valeur du flot = 35



Augmentation

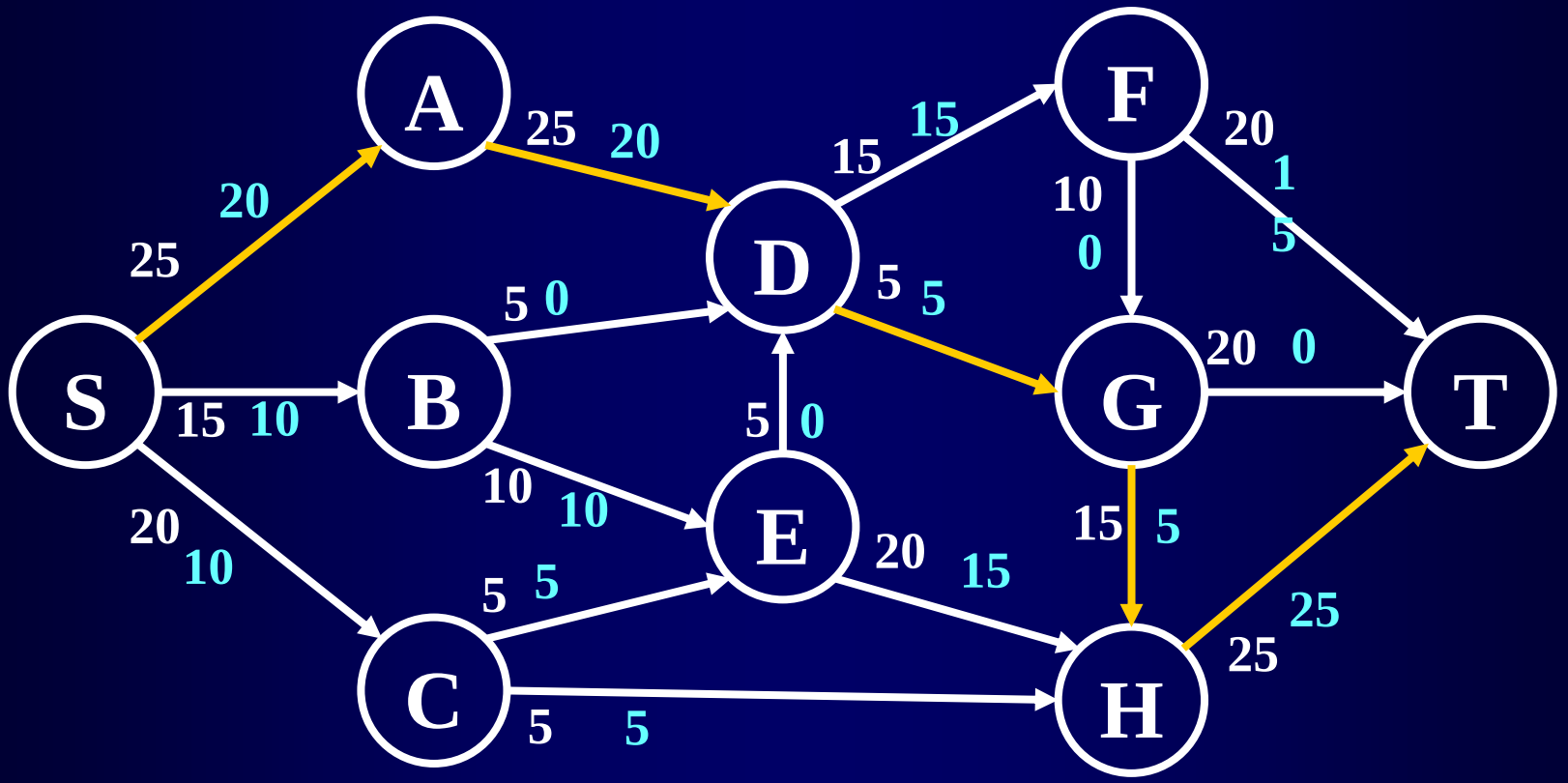
c f

Valeur du flot = 35



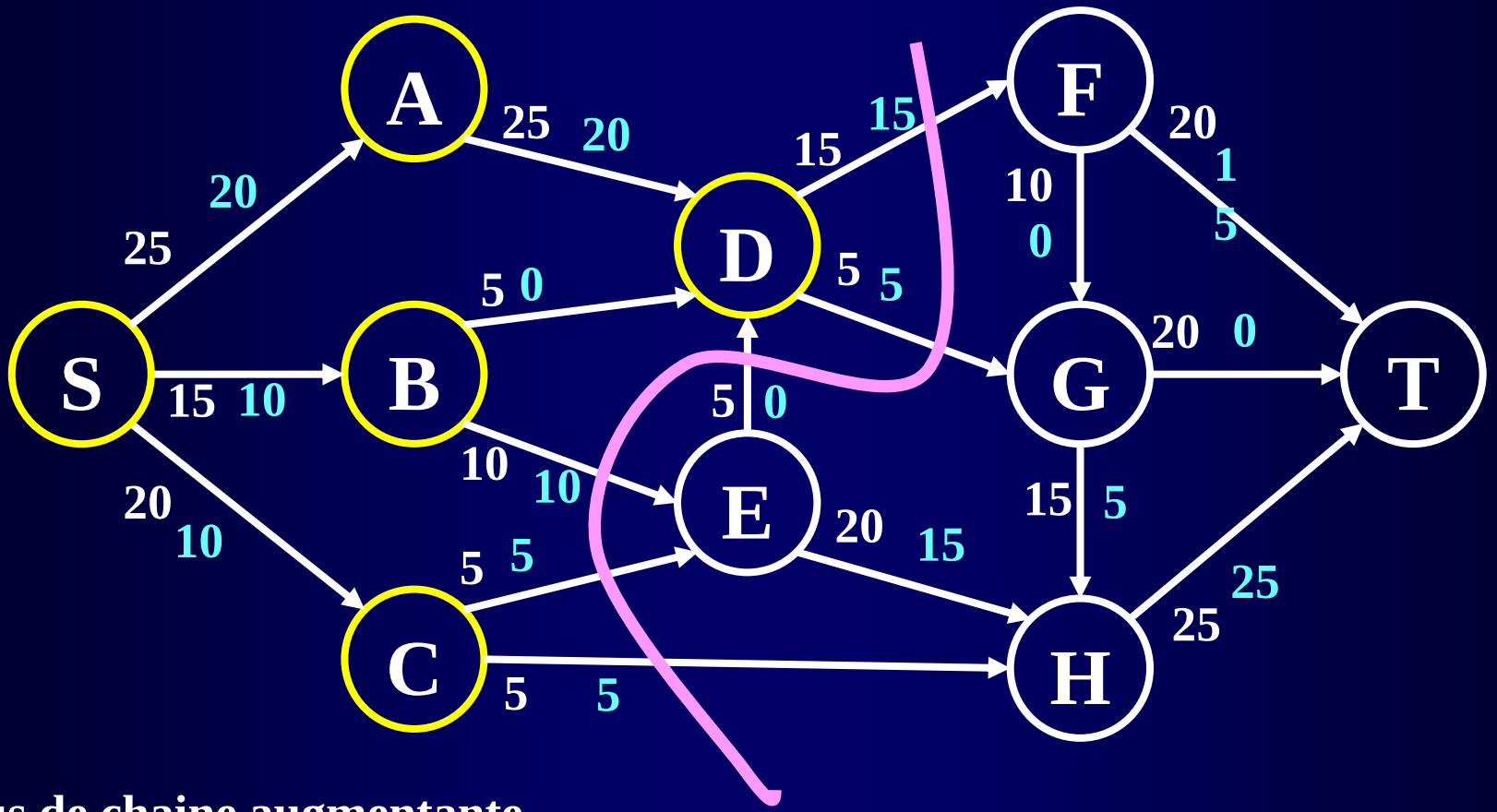
Chaine augmentante

c f



Augmentation **c f**

Valeur du flot = 40



Plus de chaine augmentante

Recherche d'une coupe minimum

c f

Valeur du flot = 40 = capacité de la coupe

Complexité de la méthode :

$$O(f_{max} |E|)$$

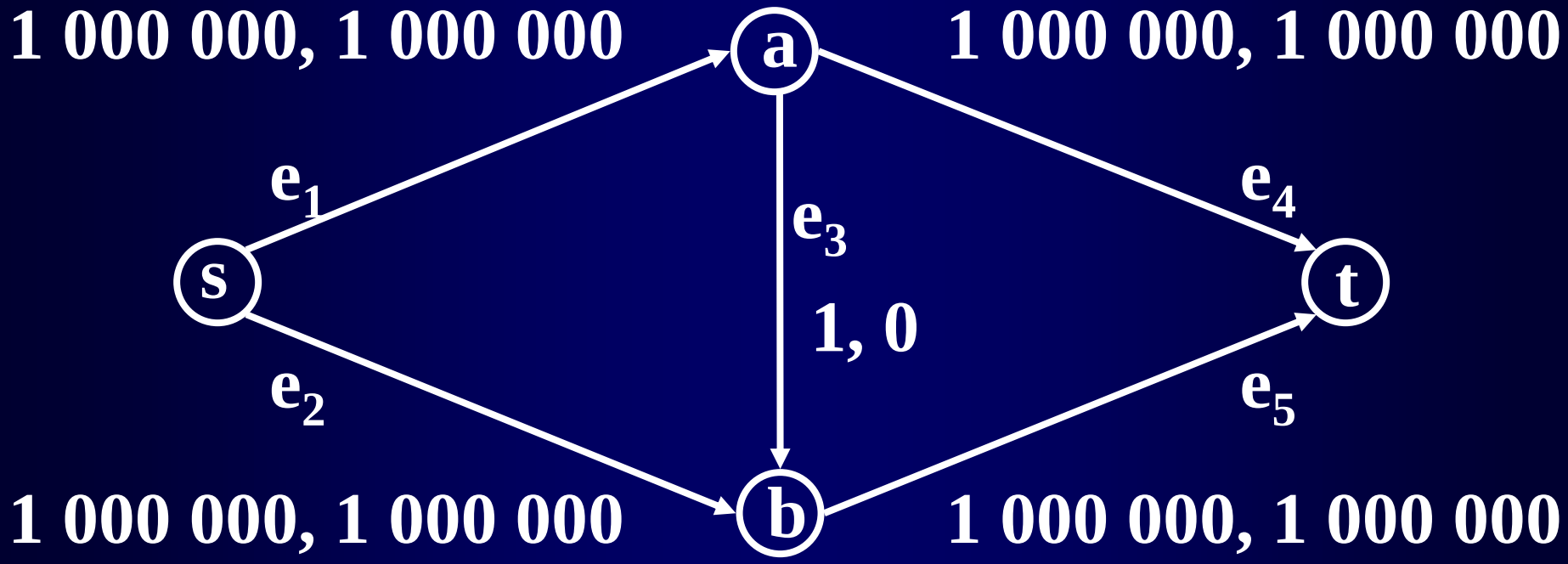
Et même ceci uniquement si les capacités sont entières !

Sinon, l'algorithme peut ne jamais terminer !

Pire, l'algorithme peut converger vers une valeur qui n'est pas le flot maximum !

Solution : choix d'un plus courte chaîne augmentante (due à Edmonds et Karp) $O(n\, m^2)$

2 000 000 augmentations !!!!



c, f

Réseaux avec bornes inf et sup

Dans un cas plus général : toute arête *e* dispose de deux bornes, *b*(*e*) et *c*(*e*) et on exige

$$b(e) \leq f(e) \leq c(e)$$

Une chaine augmentante est une chaine entre *s* et *t* sur laquelle on peut augmenter le flot, c.a.d. une chaîne telle que

- **les arcs "avant" (orientés dans le sens source vers puits) le flot est inférieur au maximum**
- **les arcs "arrière" (orientés dans le sens puits vers source) le flot est supérieur au minimum**

Adaptation de la méthode Ford & Fulkerson :

- **dès qu'on a un flot un flot admissible (qui vérifie les contraintes) on peut utiliser la méthode**
- **avant il faut chercher un flot admissible**

Recherche de flot admissible

On rajoute une "source" $\underline{s}$, et un "puits" $\underline{t}$.

On construit un réseau "classique" avec borne inférieure 0 et borne supérieure $\underline{c}$.

Pour chaque nœud v on rajoute

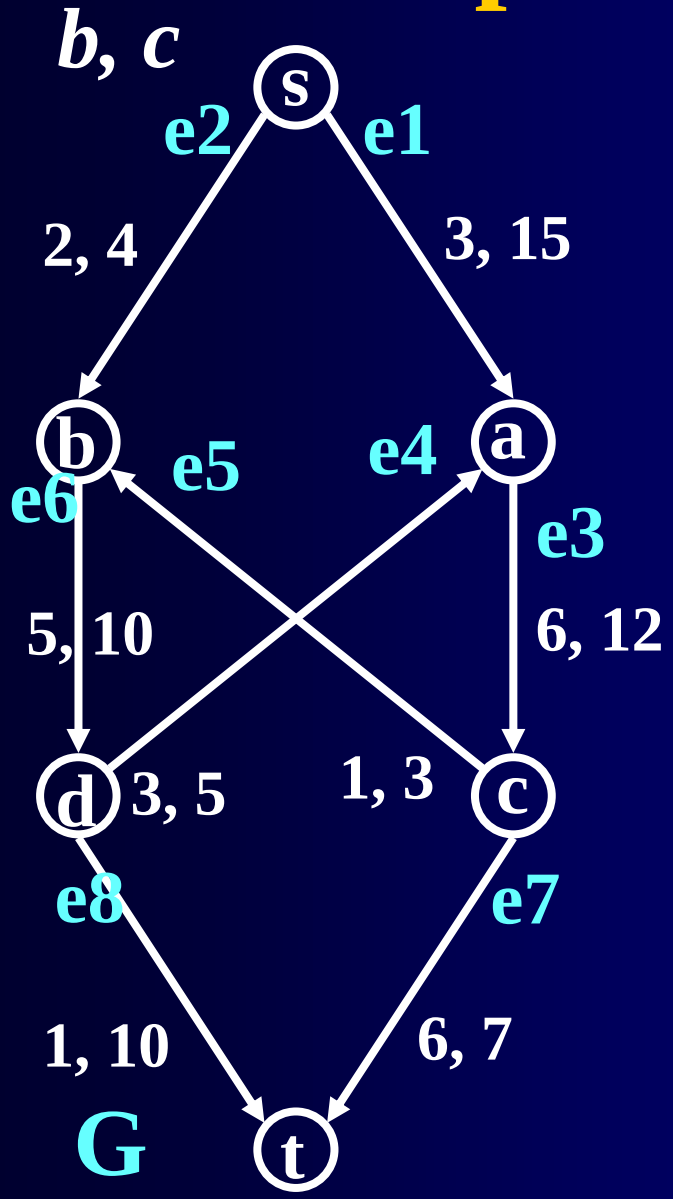
- un arc $(v, \underline{t})$ avec $\underline{c}(v, \underline{t}) = \Sigma_{e \in \mathrm{Out}(v)} b(e)$
- un arc $(\underline{s}, v)$ avec $\underline{c}(\underline{s}, v) = \Sigma_{e \in \mathrm{In}(v)} b(e)$

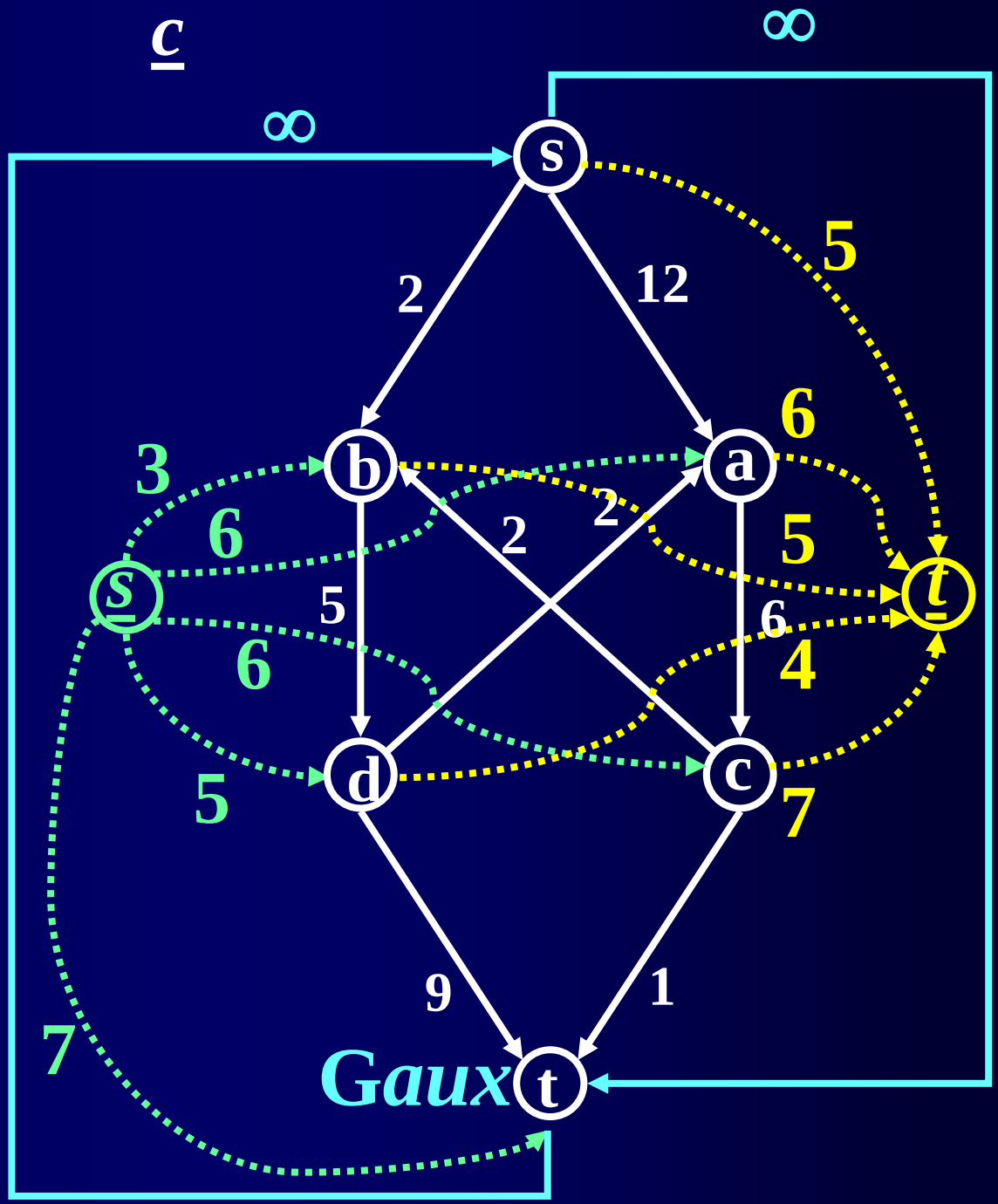
Pour toute les autres arcs on définit

$$\underline{c}(e) = c(e) - b(e)$$

On rajoute deux arcs : $\underline{c}(s, t) = \infty$ et $\underline{c}(t, s) = \infty$.

Exemple


b, c
s
e2
e1
2, 4
3, 15
b
e5
e4
a
e6
e3
5, 10
6, 12
d
3, 5
1, 3
c
e8
e7
1, 10
6, 7
G
t


c
∞
∞
s
5
2
12
6
3
b
a
2
2
5
6
s
5
t
6
6
4
5
d
c
7
9
1
7
Gaux
t

Le réseau original admet un flot admissible si et seulement si le flot maximum du réseau modifié sature toutes les arêtes sortants de *s*.

Remarque : dans ce cas les arcs entrants en *t* sont aussi saturés.