

**Master IAD — Module MGDE**

**Transparents de cours**

**année 2012–2013**

*Christophe Gonzales*



# Inférence dans les réseaux bayésiens

Christophe Gonzales

LIP6 – Université Paris 6, France

## Plan du cours 2

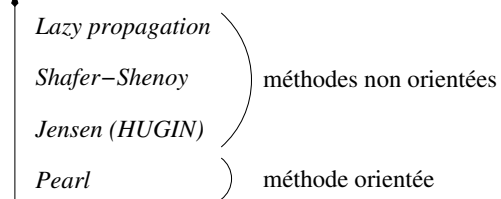
- 1 L'algorithme de Shafer-Shenoy
- 2 D'autres algorithmes d'inférence
- 3 Construction d'un arbre de jonction

## Une présentation unifiée des algorithmes d'inférence

### Définition

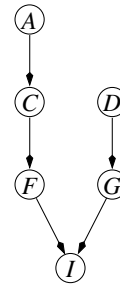
**Inférence** : calculer des probabilités (marginales, a priori, a posteriori, etc.).

### Les algorithmes d'inférence :



## Osons le calcul des probabilités a priori

$$P(A, C, D, F, G, I) = P(A)P(C|A)P(F|C)P(D)P(G|D)P(I|F, G)$$



Calcul de  $P(I)$  ?

## Shafer-Shenoy brut de fonderie

$$P(A, C, D, F, G, I) = P(A)P(C|A)P(F|C)P(D)P(G|D)P(I|F, G)$$

$$P(I) = \sum_G \left( \sum_F \left( \sum_D \left( \sum_C \left( \sum_A P(A, C, D, F, G, I) \right) \right) \right) \right)$$

$$\sum_A P(A, C, D, F, G, I) = \underbrace{\left( \sum_A P(A)P(C|A) \right)}_{P(C)} P(F|C)P(D)P(G|D)P(I|F, G)$$

$$\sum_A P(A, C, D, F, G, I) = P(C)P(F|C)P(D)P(G|D)P(I|F, G)$$

$$\sum_C \sum_A P(A, C, D, F, G, I) = \underbrace{\left( \sum_C P(C)P(F|C) \right)}_{P(F)} P(D)P(G|D)P(I|F, G)$$

## Dissection du produit de deux probabilités

$$P(A, B|C) = \begin{matrix} a_1 & a_2 \\ \overbrace{\begin{pmatrix} 0,15 & 0,18 \\ 0,15 & 0,12 \end{pmatrix}}^{c_1 \quad c_2} \quad \overbrace{\begin{pmatrix} 0,07 & 0,56 \\ 0,63 & 0,14 \end{pmatrix}}^{c_1 \quad c_2} \end{matrix} b_1 \quad \begin{matrix} a_1 & a_2 \\ \overbrace{\begin{pmatrix} 0,5 & 0,6 \\ 0,5 & 0,4 \end{pmatrix}}^{c_1 \quad c_2} \end{matrix} b_2 \times \begin{matrix} a_1 & a_2 \\ (0,3) & (0,7) \end{matrix} P(A)$$

$P(B|A, C)$

$$P(I, C|B) = \begin{matrix} b_1 & b_2 \\ \overbrace{\begin{pmatrix} 0,48 & 0,08 \\ 0,12 & 0,32 \end{pmatrix}}^{c_1 \quad c_2} \quad \overbrace{\begin{pmatrix} 0,48 & 0,08 \\ 0,12 & 0,32 \end{pmatrix}}^{c_1 \quad c_2} \end{matrix} i_1 \quad \begin{matrix} P(I|C) \\ \overbrace{\begin{pmatrix} 0,8 & 0,2 \\ 0,2 & 0,8 \end{pmatrix}}^{c_1 \quad c_2} \end{matrix} i_2 \times \begin{matrix} P(C) \\ \overbrace{\begin{pmatrix} 0,6 & 0,4 \end{pmatrix}}^{c_1 \quad c_2} \end{matrix}$$

## Shafer-Shenoy graphique (1/6)

### Séquence d'élimination

A C D F G

$$P(A, C, D, F, G, I) = P(A)P(C|A) P(F|C)P(D)P(G|D)P(I|F, G)$$

A AC FC D GD IFG  
 $P(A) P(C|A) P(F|C) P(D) P(G|D) P(I|F, G)$

$$\text{somme sur } A \Rightarrow P(C) = \sum_A P(A)P(C|A)$$

## Shafer-Shenoy graphique (2/6)

### Séquence d'élimination

A C D F G

$$P(C, D, F, G, I) = P(C)P(F|C) P(D)P(G|D)P(I|F, G)$$



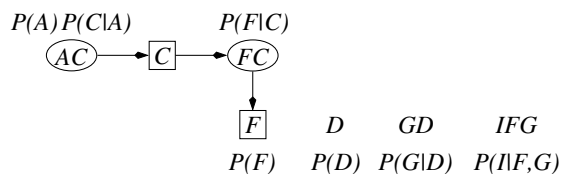
$$\text{somme sur } C \Rightarrow P(F) = \sum_C P(C)P(F|C)$$

## Shafer-Shenoy graphique (3/6)

### Séquence d'élimination

A C D F G

$$P(D, F, G, I) = P(F) P(D)P(G|D) P(I|F, G)$$



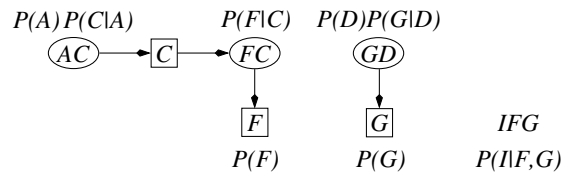
$$\text{somme sur } D \Rightarrow P(G) = \sum_D P(D)P(G|D)$$

## Shafer-Shenoy graphique (4/6)

### Séquence d'élimination

A C D **F** G

$$P(D, F, G, I) = P(F)P(I|F, G)P(G)$$



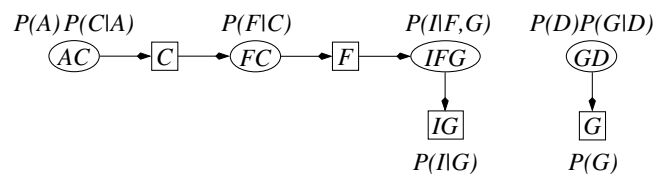
$$\text{somme sur } F \Rightarrow P(I|G) = \sum_F P(F)P(I|F, G)$$

## Shafer-Shenoy graphique (5/6)

### Séquence d'élimination

A C D F **G**

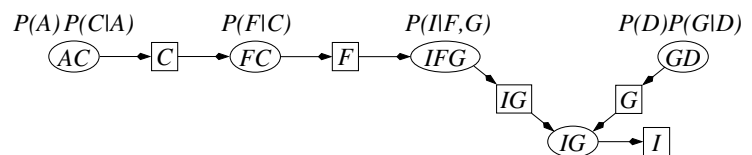
$$P(D, F, G, I) = P(I|G)P(G)$$



$$\text{somme sur } G \Rightarrow P(I) = \sum_G P(G)P(I|G)$$

## Shafer-Shenoy graphique (6/6)

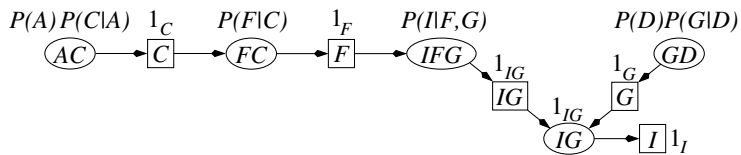
### Le graphe final obtenu par Shafer-Shenoy



### Algorithme de Shafer-Shenoy

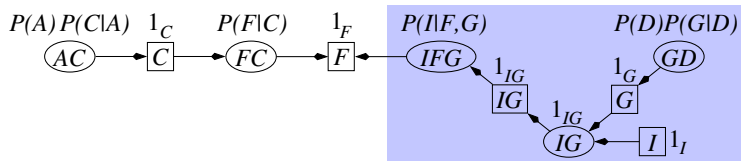
- 1 Se donner une séquence d'élimination des nœuds  
 $\Rightarrow$  *join tree*,
- 2 propager les impacts dans le sens des flèches :
  - dans les ellipses (cliques), on effectue des multiplications,
  - dans les rectangles (séparateurs), on effectue des additions (projections).

### Quelques remarques sur Shafer-Shenoy (1/3)

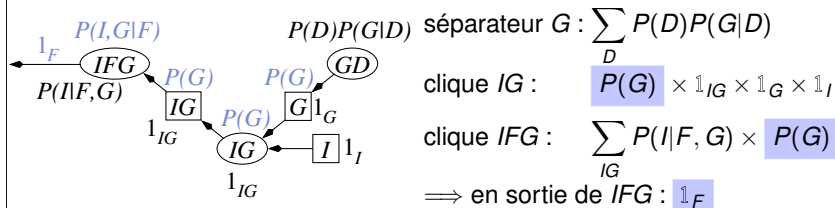


- Loi jointe = produit des fonctions des cliques et des séparateurs :  $P(A, C, D, F, G, I) = P(A)P(C|A)P(F|C)P(D)P(G|D)P(I|F, G)$ .
- Élimination récursive des cliques et séparateurs « externes »  $\implies$  le produit des fonctions des cliques et des séparateurs restants = loi jointe des variables restantes.

### Quelques remarques sur Shafer-Shenoy (2/3)

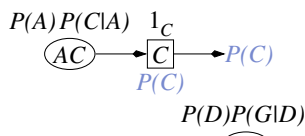
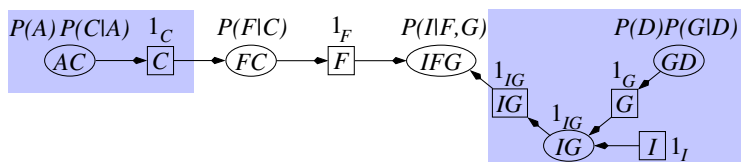


$$P(A) \times P(C|A) \times P(F|C) \times 1_C \times 1_F = P(A, C, F).$$



séparateur  $G$  :  $\sum_D P(D)P(G|D)$   
 clique  $IG$  :  $P(G) \times 1_{IG} \times 1_G \times 1_I$   
 clique  $IFG$  :  $\sum_{IG} P(I|F, G) \times P(G)$   
 $\implies$  en sortie de  $IFG$  :  $1_F$

### Quelques remarques sur Shafer-Shenoy (3/3)

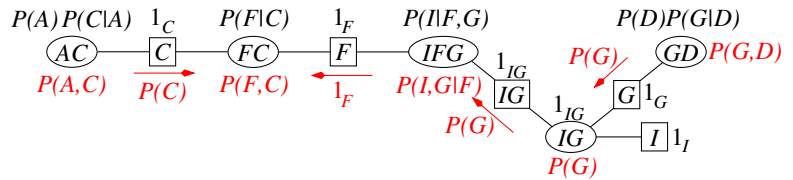
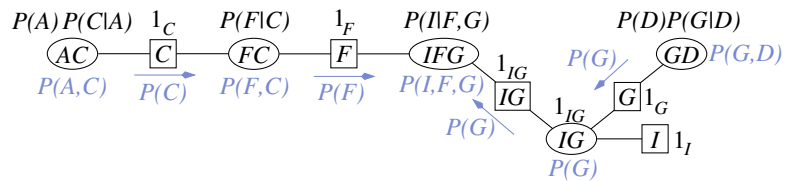


Produit des cliques et séparateurs restants :  $P(C) P(F|C) 1_F P(I|F, G) P(G) = P(I, C, F, G)$

**Conclusion :** on peut utiliser le même graphe pour calculer toutes les probabilités marginales

## C'est pas le deux en un, mais le tout en deux (1/3)

En bleu : les calculs de  $P(I, F, G)$ , en rouge, ceux de  $P(F, C)$



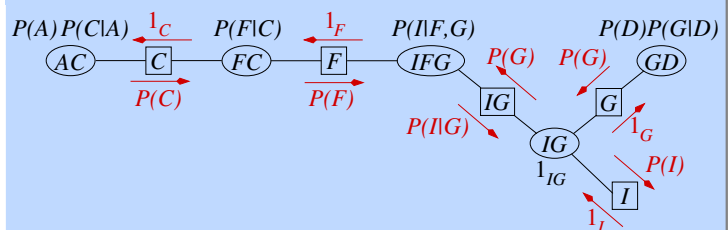
## C'est pas le deux en un, mais le tout en deux (2/3)

### Algorithme de Shafer-Shenoy

- 1 Chaque séparateur contient deux messages initialisés à 1, un en direction de chaque clique voisine.
- 2 Chaque nœud du join tree envoie des messages vers ses voisins en respectant les deux règles suivantes :
  - 1 avant d'envoyer un message vers son voisin X, le nœud Y attend que tous ses autres voisins lui aient envoyé leur message.
  - 2 le message d'un nœud Y vers son voisin X est le produit de tous les messages reçus par Y, à l'exception de celui envoyé par X, et de la table stockée par Y, le tout marginalisé sur X (c'est-à-dire sommé sur les variables de  $Y \setminus X$ ).

## C'est pas le deux en un, mais le tout en deux (3/3)

### Algorithme de Shafer-Shenoy

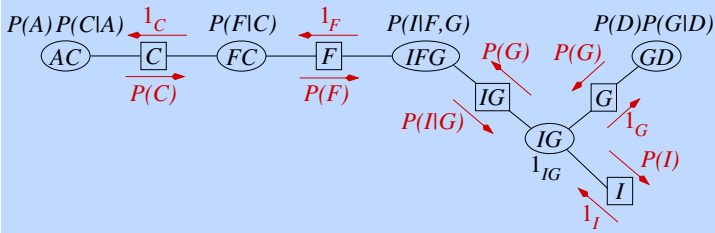


À la fin de l'algorithme, pour tout nœud X, le produit de la table stockée en X par l'ensemble des messages envoyés à X est la probabilité jointe des variables de X.



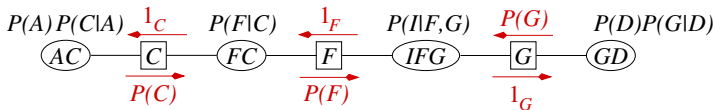
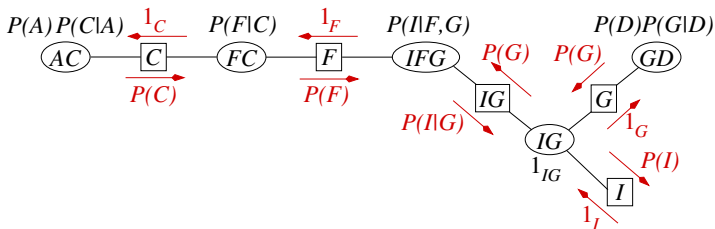
# Shafer-Shenoy : ce qu'il faut retenir

## Algorithme de Shafer-Shenoy



- cliques = ellipses, séparateurs = rectangles
- algorithme par envoi de messages
- opérateurs = + sur les séparateurs, × sur les cliques

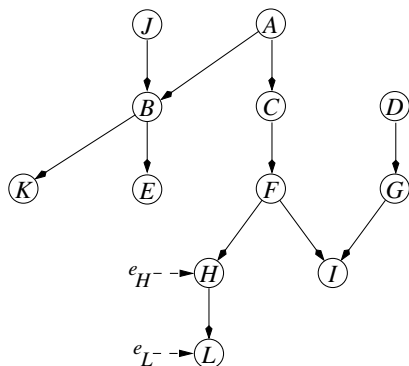
## Les arbres de jonction



arbre de jonction : suppression des cliques incluses dans d'autres cliques

## Soyons observateur : les probabilités a posteriori

Nous observons des informations  $e_H$  sur  $H$  et  $e_L$  sur  $L$ .  
 On veut connaître  $P(I|e_H, e_L)$   
 ⇒ on va calculer  $P(I, e_H, e_L)$  car c'est plus simple



## Quelles sont ces informations ?

### Teneur des informations

informations = observations :


$e_L = \ll L \text{ ne peut plus prendre les valeurs } l_1 \text{ et } l_4 \gg$

*Entrée de ces observations* :  $P(e_L|L)$

## Vous avez dit calcul de $P(e_L|L)$ ? (1/2)

observation :  $e_L = \ll \text{Le capteur m'indique que } L \text{ ne peut plus prendre les valeurs } l_1 \text{ et } l_4 \gg$

$$P(e_L|L) = \begin{array}{c|c} 0 & l_1 \\ 1 & l_2 \\ 1 & l_3 \\ 0 & l_4 \end{array}$$

 dimension de  $P(e_L|L) = |L|$

## Vous avez dit calcul de $P(e_L|L)$ ? (2/2)

observation :  $e_L = \ll \text{Le capteur m'indique que } L \text{ a pris la valeur } l_2. \text{ Mais je n'ai pas totalement confiance en ce capteur. Je pense qu'il y a 90\% de chances pour que } L = l_2, \text{ mais il y a également 10\% de chances que } L = l_1 \text{ ou que } L = l_3. \gg$

$$P(e_L|L) = \begin{array}{c|c} 0.1 & l_1 \\ 0.9 & l_2 \\ 0.1 & l_3 \\ 0 & l_4 \end{array}$$

## Hypothèses et conséquences (1/4)

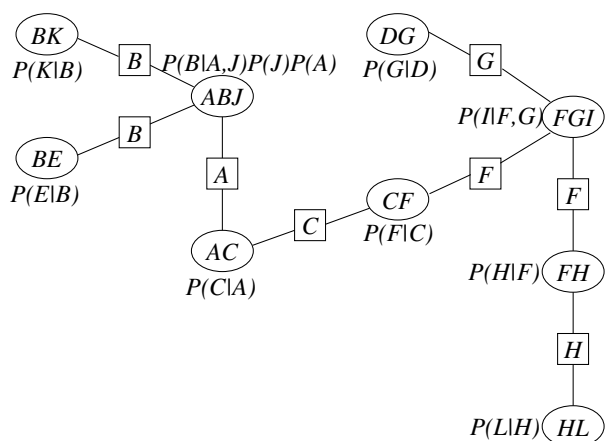
### Hypothèse

Toute information  $e_X$  sur un nœud  $X$  est indépendante du reste du réseau bayésien conditionnellement à  $X$ .

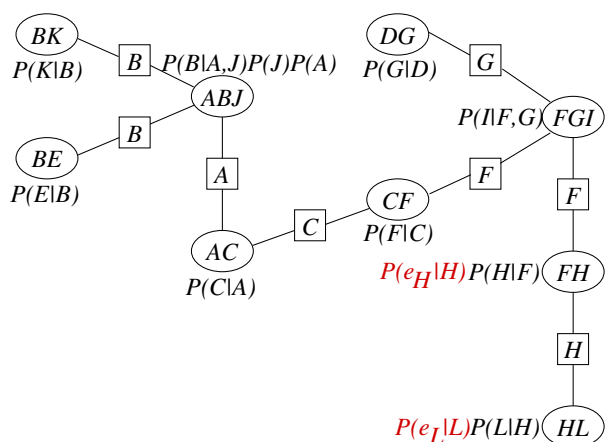
### Conséquence

$$\begin{aligned}
 P(A, B, C, D, E, F, G, H, I, J, K, L, e_H, e_L) &= \\
 &P(e_H|A, B, C, D, E, F, G, H, I, J, K, L, e_L) \times \\
 &P(e_L|A, B, C, D, E, F, G, H, I, J, K, L) \times \\
 &P(A, B, C, D, E, F, G, H, I, J, K, L) \\
 &= P(e_H|H)P(e_L|L)P(A, B, C, D, E, F, G, H, I, J, K, L)
 \end{aligned}$$

## Hypothèses et conséquences (2/4)



## Hypothèses et conséquences (3/4)



L'algorithme de Shafer-Shenoy permet donc de calculer dans la clique  $FGI$  :  $P(F, G, I, e_H, e_L)$

$$\Rightarrow P(I, e_H, e_L) = \sum_{F,G} P(F, G, I, e_H, e_L)$$

$$\Rightarrow P(I|e_H, e_L) = \frac{P(I, e_H, e_L)}{P(e_H, e_L)}$$

$$\text{Or } P(e_H, e_L) = \sum_I P(I, e_H, e_L)$$

$$\text{Donc } P(I|e_H, e_L) = \frac{P(I, e_H, e_L)}{\sum_I P(I, e_H, e_L)}$$

## Résumé sur l'algorithme de Shafer-Shenoy

### Algorithme de Shafer-Shenoy

Pour calculer une proba *a posteriori*  $P(Y|e)$  :

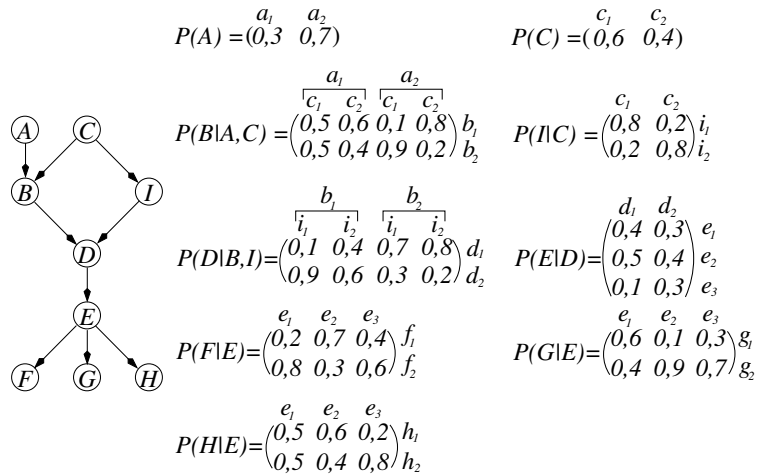
- 1 construire l'arbre de jonction
- 2 insérer les probas conditionnelles du réseau bayésien dans les cliques
- 3 insérer des tables contenant uniquement des 1 dans les séparateurs
- 4 pour chaque information  $e_X$ , calculer  $P(e_X|X)$  et insérer cette proba dans une clique contenant  $X$
- 5 envoyer les messages dans l'arbre de jonction
- 6 choisir une clique contenant  $Y$ , faire le produit de sa table de proba par tous les messages qui lui ont été envoyés, puis sommer sur toutes les variables  $\neq Y \Rightarrow P(Y, e)$
- 7 normaliser  $P(Y, e) \Rightarrow P(Y|e)$

## Petit guide pratique sur Shafer-Shenoy

### Quelques références

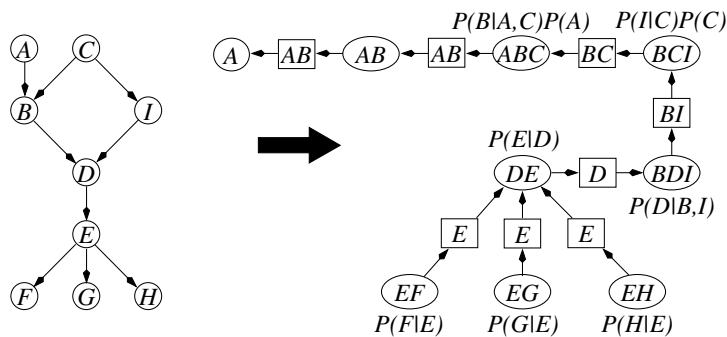
- 1 **G. Shafer (1996)** *Probabilistic expert systems*, Society for Industrial and Applied Mathematics.
- 2 **P.P. Shenoy, G. Shafer (1990)** *Axioms for probability and belief-function propagation*, *Uncertainty in Artificial Intelligence* 4, pp.169–198.
- 3 **P.P. Shenoy (1997)** *Binary join trees for computing marginals in the Shenoy-Shafer architecture*, *International Journal of Approximate Reasoning* 17, pp.1–25.
- 4 **V. Lepar, P.P. Shenoy (1998)** *A Comparison of Lauritzen-Spiegelhalter, Hugin and Shenoy-Shafer Architectures for Computing Marginals of Probability Distributions*, *Proceedings of UAI-98*, pp.328–337.

## Exemple à l'usage des étudiants studieux (1/9)



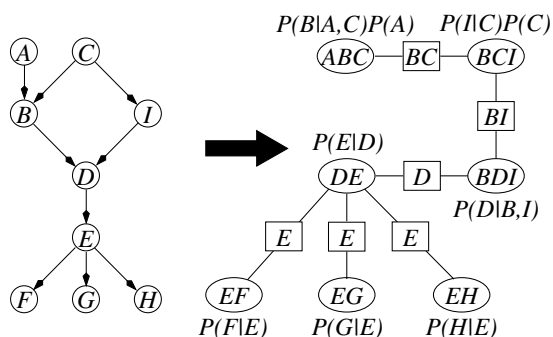
## Exemple à l'usage des étudiants studieux (2/9)

Séquence d'élimination : F, H, G, E, D, I, C, B, A

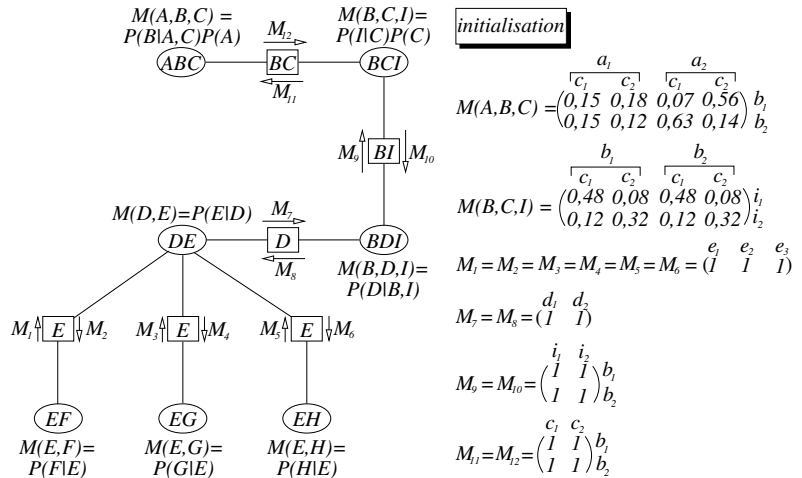


## Exemple à l'usage des étudiants studieux (3/9)

Séquence d'élimination : F, H, G, E, D, I, C, B, A



## Exemple à l'usage des étudiants studieux (4/9)



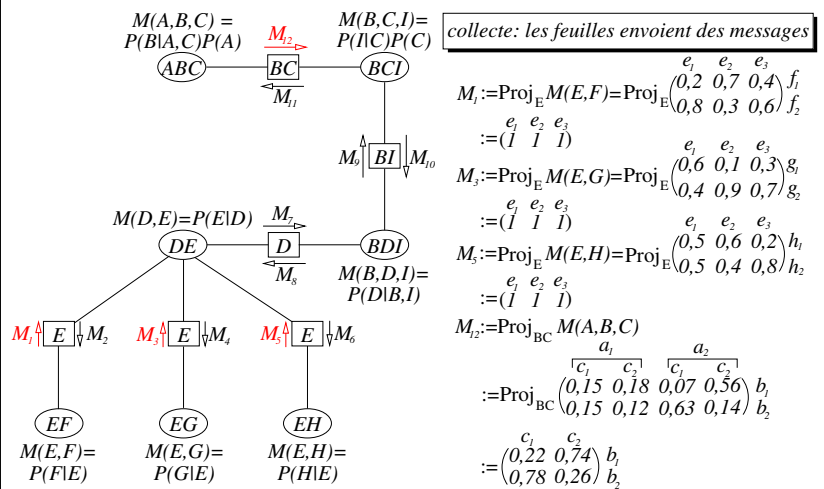
## Rappel : dissection du produit de deux probabilités

$$M(A,B,C) = \begin{pmatrix} \overbrace{c_1}^{a_1} & \overbrace{c_2}^{a_2} \\ 0,15 & 0,18 & 0,07 & 0,56 \end{pmatrix} \begin{matrix} b_1 \\ b_2 \end{matrix} = \begin{pmatrix} \overbrace{c_1}^{a_1} & \overbrace{c_2}^{a_2} \\ 0,5 & 0,6 & 0,1 & 0,8 \end{pmatrix} \begin{matrix} b_1 \\ b_2 \end{matrix} \times \begin{pmatrix} a_1 & a_2 \\ 0,3 & 0,7 \end{pmatrix} P(A)$$

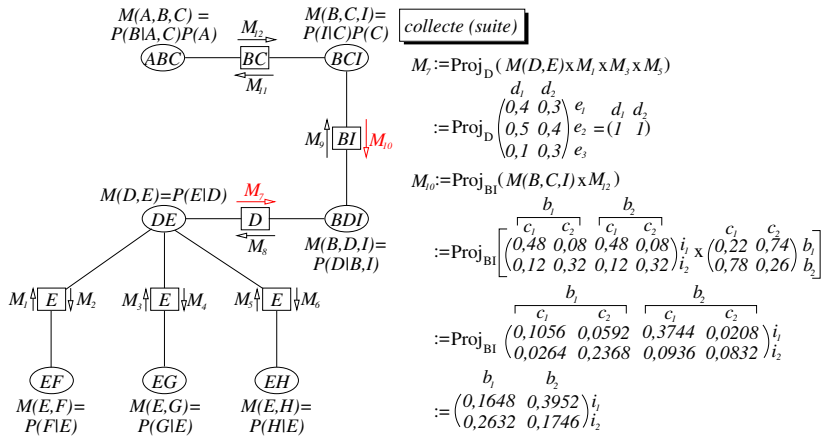
$P(B|A,C)$

$$M(B,C,I) = \begin{pmatrix} \overbrace{b_1}^{c_1} & \overbrace{b_2}^{c_2} \\ 0,48 & 0,08 & 0,48 & 0,08 \end{pmatrix} \begin{matrix} i_1 \\ i_2 \end{matrix} = \begin{pmatrix} \overbrace{c_1}^{P(I|C)} & \overbrace{c_2}^{P(C)} \\ 0,8 & 0,2 \\ 0,2 & 0,8 \end{pmatrix} \begin{matrix} i_1 \\ i_2 \end{matrix} \times \begin{pmatrix} c_1 & c_2 \\ 0,6 & 0,4 \end{pmatrix}$$

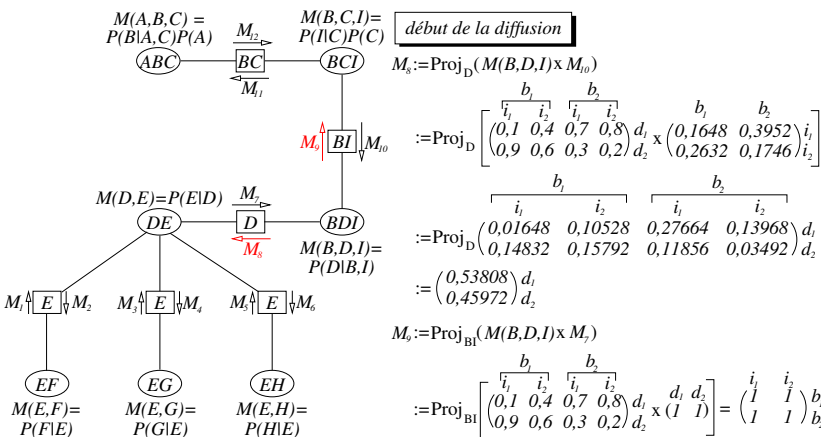
## Exemple à l'usage des étudiants studieux (5/9)



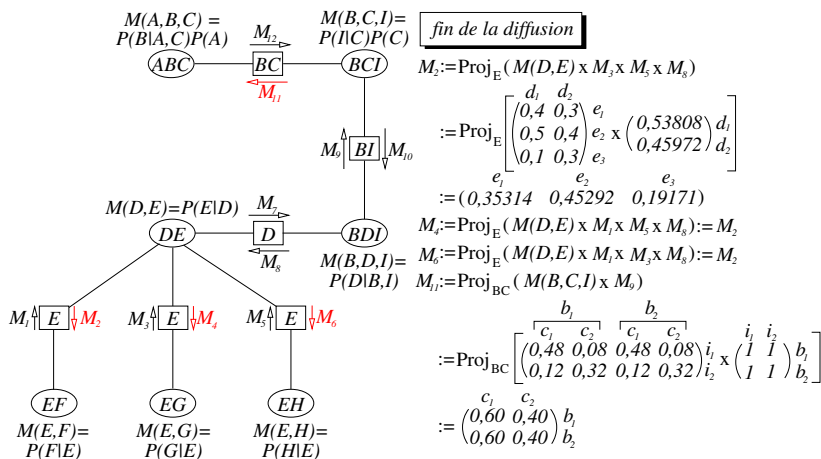
## Exemple à l'usage des étudiants studieux (6/9)



## Exemple à l'usage des étudiants studieux (7/9)



## Exemple à l'usage des étudiants studieux (8/9)



$$\begin{aligned}
 M_{11} \times M(A, B, C) &= \begin{pmatrix} c_1 & c_2 \\ 0,60 & 0,40 \\ 0,60 & 0,40 \end{pmatrix} b_1 \times \begin{pmatrix} a_1 & a_2 \\ c_1 & c_2 \\ 0,15 & 0,18 & 0,07 & 0,56 \\ 0,15 & 0,12 & 0,63 & 0,14 \end{pmatrix} b_2 \\
 &= \begin{pmatrix} c_1 & c_2 \\ 0,090 & 0,072 & 0,042 & 0,224 \\ 0,090 & 0,048 & 0,378 & 0,056 \end{pmatrix} b_1 b_2
 \end{aligned}$$

- 1 Petit exercice de calcul mental : calculez la projection de la matrice ci-dessus sur  $BC$ .
- 2 Petit exercice de déduction : à quoi correspond cette matrice ?
- 3 Ultime exercice à l'usage de l'élite (et des biens nantis) : calculez le produit  $M_{11} \times M_{12}$ .
- 4 Un dernier pour la route : dites « bizarre, bizarre, comme c'est bizarre », et expliquez pourquoi c'est bizarre.

## 2 Existe-t-il d'autres algorithmes d'inférence ?

### De Shafer-Shenoy à Lazy propagation



si  $P(A, B, C, D, E) = P(A)P(B)P(C|A, B)P(D)P(E|C, D)$

alors  $P(B, C, D, E) = \sum_A P(A)P(B)P(C|A, B)P(D)P(E|C, D)$

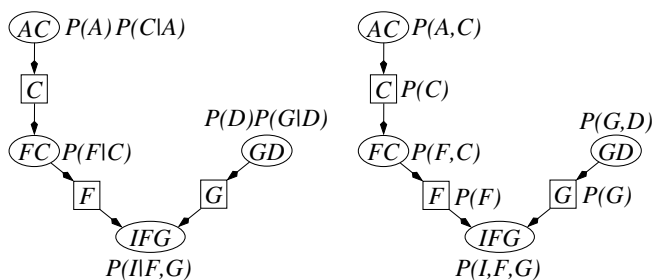
$$= \left( \sum_A P(A)P(C|A, B) \right) P(B)P(D)P(E|C, D)$$

⇒ on a intérêt à ne pas effectuer les produits avant les sommes

#### Lazy propagation

Principe : garder les produits sous forme de listes et n'effectuer les multiplications que lorsque c'est nécessaire.





**Shafer-Shenoy :** élimination de  $A \implies P(C, e_A) = \sum_A P(A, e_A)P(C|A)$

élimination de  $F \implies P(F, C, e_A) = P(F|C)P(C, e_A)$

**Jensen :** élimination de  $F \implies P(F, C, e_A) = \frac{P(F, C)}{P(C)}P(C, e_A)$

## Petit guide sur Jensen et Lazy Propagation

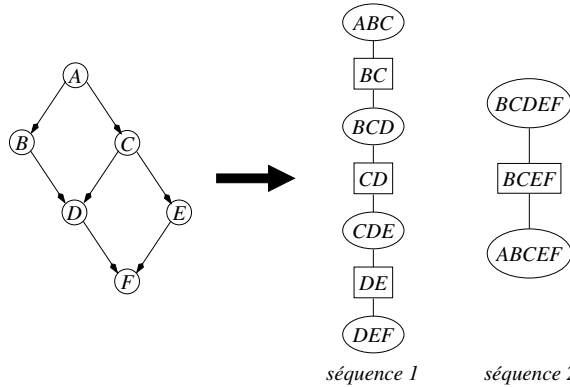
### Quelques références

- 1 S.L. Lauritzen, D.J. Spiegelhalter (1988) *Local computations with probabilities on graphical structures and their application to expert systems(with discussion)*, Journal of the Royal Statistical Society, *Series B*, 50, pp.157–224.
- 2 F.V. Jensen, S.L. Lauritzen, K.G. Olesen (1990) *Bayesian Updating in Causal Probabilistic Networks by Local Computations*, Comp. Stat. Quarterly, 4, pp.269–282.
- 3 A.L. Madsen, F.V. Jensen (1998) *Lazy Propagation in Junction Trees*, Proceedings d'UAI-98.
- 4 A.L. Madsen, F.V. Jensen (1999) *Lazy Propagation : A Junction Tree Inference Algorithm Based on Lazy Evaluation*, Artificial Intelligence, 113, pp.203–245.

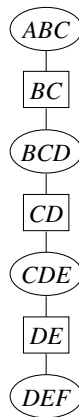
- 3 Comment construire l'arbre de jonction  
ou la triangulation par l'exemple

## Toutes les séquences d'élimination ne sont pas égales

Séquence 1 : A, B, C, F, D, E    Séquence 2 : D, C, A, E, B, F



## De la propriété d'intersection courante



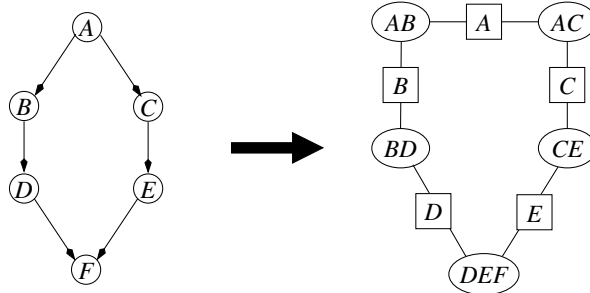
### Propriété d'intersection courante

Soient  $C_1$  et  $C_2$  deux cliques quelconques de l'arbre de jonction et soit  $S = C_1 \cap C_2 \neq \emptyset$ . Alors sur toute chaîne reliant  $C_1$  et  $C_2$ , les cliques et séparateurs contiennent  $S$

### Théorème

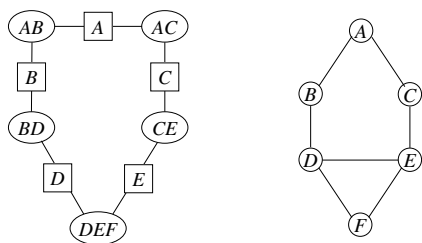
L'algorithme de Shafer-Shenoy fonctionne avec n'importe quel graphe sans cycle vérifiant la propriété d'intersection courante (ce que l'on appelle un *join tree*)

## Un graphe de jonction avec cycles



Si l'on n'y prend garde, des cycles peuvent exister bien que la propriété d'intersection courante soit vérifiée

## Graphe de jonction et triangulation (1/2)

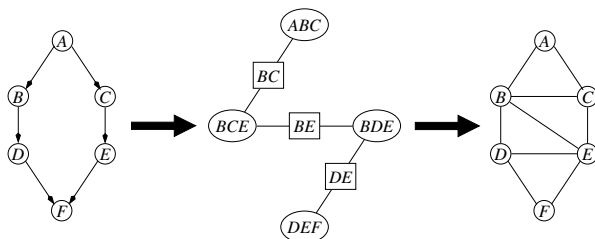


### Triangulation

Un graphe non orienté est triangulé si et seulement si, pour tout cycle de longueur 4 ou plus, il existe une corde, c'est-à-dire une arête reliant deux nœuds non consécutifs du cycle

**Exemple :** le graphe ci-dessus n'est pas triangulé car le cycle A, B, D, E, C, A ne comporte pas de corde

## Graphe de jonction et triangulation (2/2)



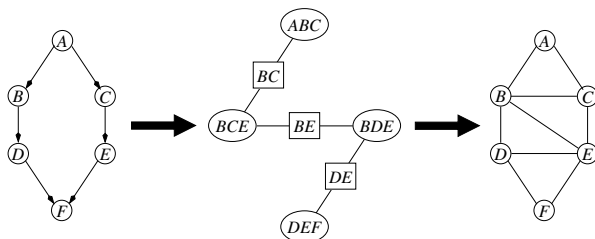
### Proposition

il y a équivalence entre les deux assertions :

- 1 Le graphe de jonction est acyclique
- 2 Le graphe non orienté correspondant est triangulé

⇒ pour trouver un « bon » arbre de jonction, il faut trouver une « bonne » triangulation

## Un peu de morale, ça ne fait pas de mal



### Moralisation

relier tous les parents d'un même nœud, puis supprimer les orientations ⇒ le graphe moral.

⇒ les cliques pourront contenir l'ensemble des probabilités conditionnelles de la décomposition de la loi jointe

## Recherche des triangulations optimales (1/2)

### Proposition (Rose 1970)

Un graphe non orienté est triangulé si et seulement si l'application des deux règles suivantes permet d'éliminer tous les nœuds  $X_i$  du graphe sans rajouter une seule arête :

- 1 on rajoute des arêtes entre tous les voisins du nœud  $X_i$  que l'on veut éliminer (on forme une clique)
- 2 on supprime  $X_i$  ainsi que les arêtes qui lui sont adjacentes du graphe

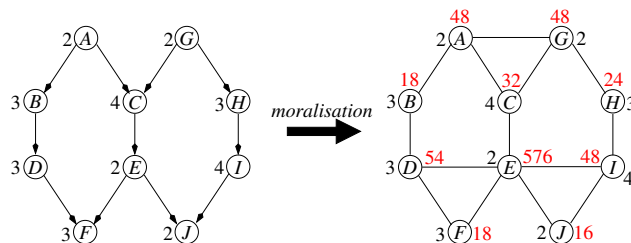
⇒ pour créer un join tree, il suffit de partir d'un graphe non orienté et d'appliquer, avec une certaine séquence d'élimination, les deux points ci-dessus

**Mellouli (87)** : Tout join tree « optimal » peut être construit à partir d'une séquence d'élimination

## Recherche des triangulations optimales (2/2)

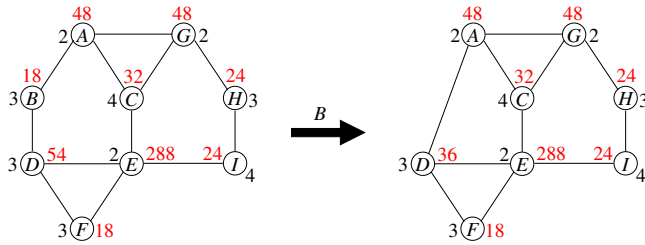
- **Arnborg et al. (87)** : trouver la triangulation optimale est NP-difficile ⇒ essayer de trouver des heuristiques
- **Kjærulff (90)** : un algorithme glouton rapide et efficace :  
Soit un graphe non orienté (moral)  $G = (X, E)$ ,  $X = \{X_1, \dots, X_n\}$ 
  - 1 Associer à chaque  $X_i$  un « poids » égal au produit des modalités de  $X_i$  et de ses voisins
  - 2 éliminer le nœud  $X_i$  dont le poids est minimal (i.e., relier tous ses voisins de manière à former une clique  $C_i$  puis éliminer  $X_i$  et ses arêtes adjacentes)
  - 3 mettre à jour les poids des nœuds restants⇒ les  $C_i$  sont les cliques (ellipses) du join tree
- **van den Eijkhof & Bodlaender (2002)** : “safe reductions”  
⇒ élimination de variables avec garantie d'optimalité
- Autres algorithmes : Becker & Geiger (96) ; Shoiket & Geiger (87)

## Exemple de création de join tree (1/5)



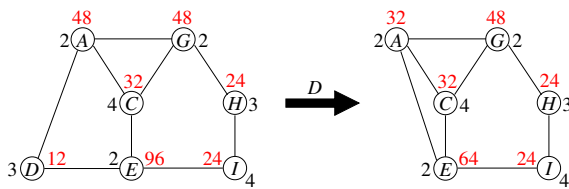
Variable à éliminer :  $J$  ⇒ clique  $EIJ$

## Exemple de création de join tree (2/5)



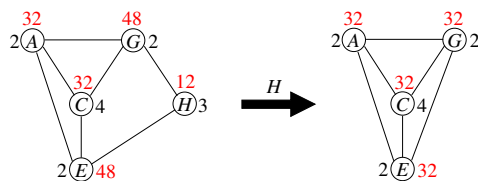
première variable à éliminer :  $B \Rightarrow$  clique  $ABD$   
 deuxième variable à éliminer :  $F \Rightarrow$  clique  $DEF$

## Exemple de création de join tree (3/5)



première variable à éliminer :  $D \Rightarrow$  clique  $ADE$   
 deuxième variable à éliminer :  $I \Rightarrow$  clique  $EHI$

## Exemple de création de join tree (4/5)



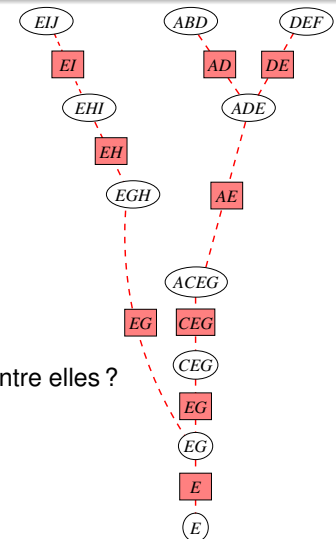
- première variable à éliminer :  $H \Rightarrow$  clique  $EGH$
- deuxième variable à éliminer :  $A \Rightarrow$  clique  $ACEG$
- puis les autres variables peuvent être éliminées dans n'importe quel ordre puisqu'elles appartiennent toutes à la même clique :
  - $C \Rightarrow$  clique  $CEG$
  - $G \Rightarrow$  clique  $EG$
  - $E \Rightarrow$  clique  $E$

## Exemple de création de join tree (5/5)

Ensemble des cliques selon leur ordre de création (avec la variable dont l'élimination a créé la clique) :

$EIJ$  (J),  $ABD$  (B),  $DEF$  (F),  $ADE$  (D),  
 $EHI$  (I),  $EGH$  (H),  $ACEG$  (A),  
 $CEG$  (C),  $EG$  (G),  $E$  (E)

**Problème** : comment relier les cliques entre elles ?



## Des cliques vers l'arbre d'élimination (1/2)

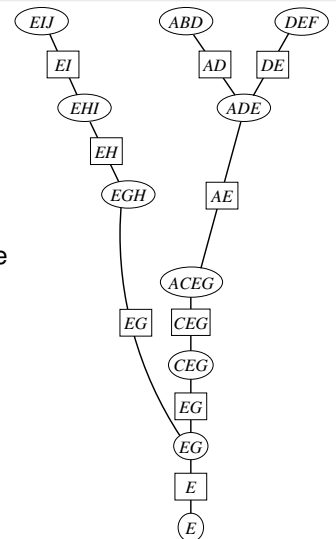
### Définition de l'arbre d'élimination

- Soit  $\sigma : \{1, \dots, n\} \mapsto \{1, \dots, n\}$  la permutation telle que les variables  $X_i$  sont éliminées dans l'ordre  $X_{\sigma(1)}, \dots, X_{\sigma(n)}$
- Pour tout  $i$ , soit  $D_{\sigma(i)}$  la clique créée au moment où  $X_{\sigma(i)}$  est éliminée
- Arbre d'élimination : graphe  $\mathcal{G} = (\mathcal{D}, \mathcal{E})$ , où :
  - $\mathcal{D} = \{D_{\sigma(i)} : i \in \{1, \dots, n\}\}$ ,
  - $\mathcal{E} = \{(D_{\sigma(i)}, D_{\sigma(j)}) : 1 \leq i < n, j = \min\{k \neq i : X_{\sigma(k)} \in D_{\sigma(i)}\}\}$

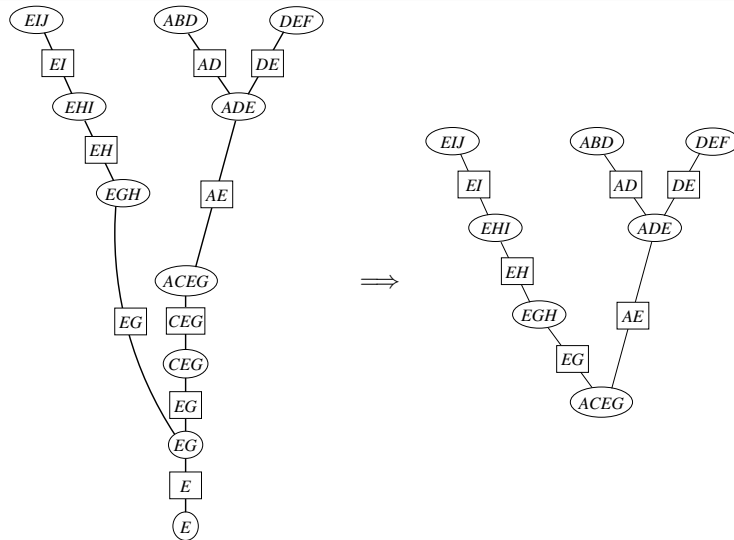
$\implies$  Si l'on trie les nœuds  $X_i$  à l'intérieur des cliques selon leur ordre d'élimination, alors :  
 on relie  $D_{\sigma(i)} = \{X_{\sigma(i)}, X_{\sigma(j)}, \dots\}$  à  $D_{\sigma(j)}$ .

## Des cliques vers l'arbre d'élimination (2/2)

- Arbre d'élimination :  $\mathcal{G} = (\mathcal{D}, \mathcal{E})$ , où :
  - $\mathcal{D} = \{D_{\sigma(i)} : i \in \{1, \dots, n\}\}$ ,
  - $\mathcal{E} = \{(D_{\sigma(i)}, D_{\sigma(j)}) : 1 \leq i < n, j = \min\{k \neq i : X_{\sigma(k)} \in D_{\sigma(i)}\}\}$
- Ensemble des cliques selon leur ordre de création (avec la variable dont l'élimination a créé la clique) :  
 $EIJ$  (J),  $ABD$  (B),  $DEF$  (F),  $ADE$  (D),  
 $EHI$  (I),  $EGH$  (H),  $ACEG$  (A),  
 $CEG$  (C),  $EG$  (G),  $E$  (E)



## De l'arbre d'élimination vers l'arbre de jonction (1/2)



## Propriétés de l'arbre d'élimination

### Propriétés

$\mathcal{D} = \{D_{\sigma(i)} : i \in \{1, \dots, n\}\},$

$\mathcal{E} = \{(D_{\sigma(i)}, D_{\sigma(j)}) : 1 \leq i < j, j = \min\{k \neq i : X_{\sigma(k)} \in D_{\sigma(i)}\}\}$

- 1 L'arbre d'élimination est un arbre
- 2 Il vérifie la propriété d'intersection courante
- 3 Soit  $D_{\sigma(j)}$  un enfant de  $D_{\sigma(i)}$ , alors  $|D_{\sigma(j)}| \geq |D_{\sigma(i)}| - 1$
- 4 Soient  $D_{\sigma(i)}$  et  $D_{\sigma(j)}$  les parents de  $D_{\sigma(k)}$ , alors  $D_{\sigma(i)} \not\subset D_{\sigma(j)}$  et  $D_{\sigma(j)} \not\subset D_{\sigma(i)}$
- 5 Soit  $D_{\sigma(j)}$  un enfant de  $D_{\sigma(i)}$ , alors  $D_{\sigma(j)} \subset D_{\sigma(i)} \iff |D_{\sigma(j)}| = |D_{\sigma(i)}| - 1$
- 6 Soit  $D_{\sigma(j)}$  un enfant de  $D_{\sigma(i)}$  tel que  $D_{\sigma(j)} \not\subset D_{\sigma(i)}$ , alors il n'existe pas d'ancêtre  $D_{\sigma(k)}$  de  $D_{\sigma(i)}$  tel que  $D_{\sigma(j)} \subset D_{\sigma(k)}$

## De l'arbre d'élimination vers l'arbre de jonction (2/2)

### Algorithme pour obtenir un arbre de jonction

- 01 créer l'arbre d'élimination  $\mathcal{G} = (\mathcal{D}, \mathcal{E})$
- 02 marquer à **false** tous les arcs de  $\mathcal{E}$
- 03 **pour**  $i$  variant de  $n$  à 1 **faire**
- 04 **si** il existe  $D_{\sigma(j)}$  parent de  $D_{\sigma(i)}$  tel que l'arc  $(D_{\sigma(j)}, D_{\sigma(i)})$  est non marqué et  $|D_{\sigma(i)}| = |D_{\sigma(j)}| - 1$  **alors**
- 05 **pour** tous les autres parents  $D_{\sigma(k)}$  de  $D_{\sigma(i)}$  **faire**
- 06 créer dans  $\mathcal{G}$  un arc  $(D_{\sigma(k)}, D_{\sigma(j)})$
- 07 marquer cet arc à **true**
- 08 **fait**
- 09 **si**  $D_{\sigma(i)}$  a un enfant  $D_{\sigma(k)}$  **alors**
- 10 créer dans  $\mathcal{G}$  un arc  $(D_{\sigma(j)}, D_{\sigma(k)})$
- 11 **fin**
- 12 supprimer  $D_{\sigma(i)}$  ainsi que ses arcs adjacents
- 13 **fait**

A la fin de l'algorithme ci-dessus,  $\mathcal{G}$  est un arbre de jonction.

## Améliorations de l'algorithme d'élimination

- Au lieu de choisir le nœud à éliminer en fonction du poids, choisir, quand c'est possible, un nœud appartenant à une seule clique (« simplicial rule », bodlaender (02)) ;
- Sous certaines contraintes, choisir un nœud presque simplicial (il manque une seule arête pour former une clique) ;
- Autres règles de réduction optimales (Buddies rule, Extended cube rule...);
- Suppression des arêtes de triangulation superflues (cf. Kjærulff (90)) — nécessite de recalculer la séquence d'élimination (par exemple par maximum cardinality search) ;
- Optimisation de l'arbre de jonction par modification des adjacences (cf. Jensen & Jensen (94)).

## Références bibliographiques (1/2)

### Quelques références

- **Kjærulff, U (1990)** *Triangulation of graphs – Algorithms giving small total state space*, technical report.
- **Kjærulff, U (1991)** *Optimal decomposition of probabilistic networks by simulated annealing*, Statistics and Computing, Vol 2, pp7-17.
- **Becker, A & Geiger, D (1996)** *A sufficiently fast algorithm for finding close to optimal junction trees*, Proceedings d'UAI-96.
- **van den Eijkhof, F & Bodlaender A (2002)** *Safe reduction rules for weighted treewidth*, Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science, vol 2573, pp176–185.

## Références bibliographiques (2/2)

### Quelques références

- **Jensen, F.V. & Jensen, F. (1994)** *Optimal junction trees* Proceedings d'UAI-94.
- **Shoikhet, K & Geiger, D (1997)** *Finding optimal triangulations via minimal vertex separators*, Proceedings de AAAI-97.
- **Leimer, H.-G. (1993)** *Optimal decomposition by clique separators*, Discrete Mathematics, vol 113, pp99-123.
- **Olesen, K & Madsen, A (1999)** *Maximal prime decomposition of Bayesian networks*, technical report.
- **Flores, J & Gámez, J & Olesen, K (2003)** *Incremental compilation of Bayesian networks*, Proceedings d'UAI-03.



# Diagrammes d'influence et CP-nets

Christophe Gonzales

LIP6 – Paris 6 Université, France

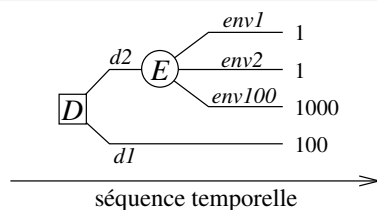
## Plan du cours 3

- 1 Les arbres de décision
- 2 Les diagrammes d'influence
- 3 Les CP-nets

## Les Arbres de décision

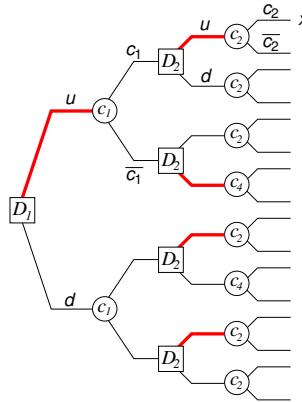
### Exemple

- enveloppe 1 contient 100 €
  - enveloppe 2 choisie parmi une pile de 100 enveloppes dont 3 contiennent 1000 € et 97 contiennent 1 €
- ⇒  $\begin{cases} \text{enveloppe 1} = 100 \text{ €} \\ \text{enveloppe 2} = 3 \text{ chances sur } 100 \text{ d'avoir } 1000 \text{ € et} \\ \quad 97 \text{ chances sur } 100 \text{ d'avoir } 1 \text{ €} \end{cases}$



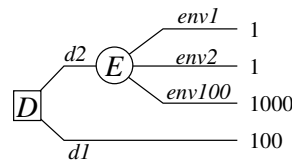
carrés = décisions    ronds = nœuds de chance

### Décisions séquentielles (1/3)



$x = \text{conséquence si } D_1 = u, C_1 = c_1, D_2 = u, C_2 = c_2$

### Décisions séquentielles (2/3)



⇒ la décision optimale est celle dont la moyenne des utilités des conséquences est la plus élevée

$d_1 \equiv \text{loterie } L_1 = \langle 100, 1 \rangle$

$d_2 \equiv \text{loterie } L_2 = \langle 1, 0.97; 1000, 0.03 \rangle$

$EU(d_1) = EU(L_1) = 100$

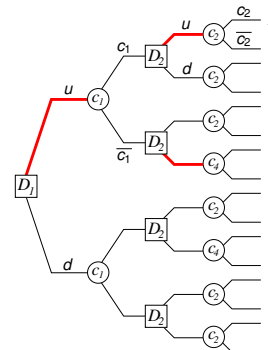
$EU(d_2) = EU(L_2) = 0,97 \times 1 + 0,03 \times 1000 = 30,97$

⇒ décision optimale selon EU :  $d_1$

### Décisions séquentielles (3/3)

#### Stratégie

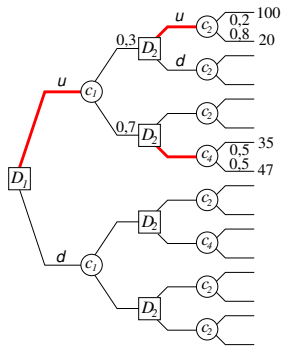
Une stratégie de décision = la sélection en **tout** sommet de décision  $D$  de l'arbre accessible compte tenu des décisions prises précédemment, d'une décision  $d$  appartenant à l'ensemble des décisions réalisables de ce sommet.



## Décisions séquentielles (3/3)

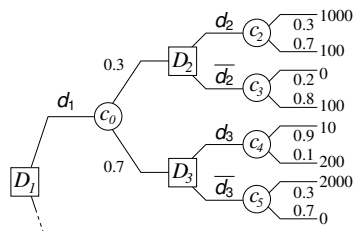
### Stratégie et espérance d'utilité

- À toute stratégie correspond une loterie
- critère d'optimalité = espérance max sur les loteries



loterie =  $\langle 20, 0.24; 35, 0.35; 47, 0.35; 100, 0.06 \rangle$

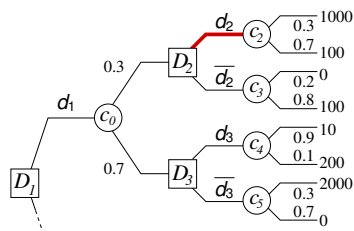
## Calculs dans un arbre de décision (1/5)



$$\begin{aligned} \text{Stratégie } S_1 &= \langle \langle D_1 = d_1, D_2 = d_2, D_3 = d_3 \rangle \rangle \\ &\equiv \langle 10, 0.7 \times 0.9; 100, 0.3 \times 0.7, 200, 0.7 \times 0.1, 1000, 0.3 \times 0.3 \rangle \\ E(S_1) &= 0.3 \times [0.7 \times 100 + 0.3 \times 1000] + 0.7 \times [0.9 \times 10 + 0.1 \times 200] \\ &= 0.3 \times E(\langle \langle D_2 = d_2 \rangle \rangle) + 0.7 \times E(\langle \langle D_3 = d_3 \rangle \rangle) \end{aligned}$$

$$\begin{aligned} \text{Stratégie } S_2 &= \langle \langle D_1 = d_1, D_2 = d_2, D_3 = \bar{d}_3 \rangle \rangle \\ &\equiv \langle 0, 0.7 \times 0.7; 100, 0.3 \times 0.7, 1000, 0.3 \times 0.3, 2000, 0.7 \times 0.3 \rangle \\ E(S_2) &= 0.3 \times [0.7 \times 100 + 0.3 \times 1000] + 0.7 \times [0.7 \times 0 + 0.3 \times 2000] \\ &= 0.3 \times E(\langle \langle D_2 = d_2 \rangle \rangle) + 0.7 \times E(\langle \langle D_3 = \bar{d}_3 \rangle \rangle) \end{aligned}$$

## Calculs dans un arbre de décision (2/5)

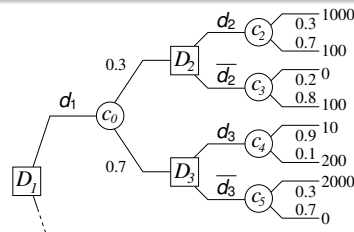


$$\begin{aligned} \text{Stratégie } S_1 &= \langle \langle D_1 = d_1, D_2 = d_2, D_3 = d_3 \rangle \rangle \\ E(S_1) &= 0.3 \times E(\langle \langle D_2 = d_2 \rangle \rangle) + 0.7 \times E(\langle \langle D_3 = d_3 \rangle \rangle) \end{aligned}$$

$$\begin{aligned} \text{Stratégie } S_2 &= \langle \langle D_1 = d_1, D_2 = d_2, D_3 = \bar{d}_3 \rangle \rangle \\ E(S_2) &= 0.3 \times E(\langle \langle D_2 = d_2 \rangle \rangle) + 0.7 \times E(\langle \langle D_3 = \bar{d}_3 \rangle \rangle) \end{aligned}$$

⇒ calculer  $E(\langle \langle D_2 = d_2 \rangle \rangle)$  une seule fois, stocker le résultat en  $D_2$  et le réutiliser pour toute stratégie contenant  $D_2 = d_2$

## Calculs dans un arbre de décision (3/5)



**Problème :** Doit-on stocker en  $D_2$  les 2 espérances  $E(\langle\langle D_2 = d_2 \rangle\rangle)$  et  $E(\langle\langle D_2 = \bar{d}_2 \rangle\rangle)$  ?

Soit  $S_1 = \langle\langle D_1 = d_1, \dots, D_2 = d_2 \rangle\rangle$  et  $S_2 = \langle\langle D_1 = d_1, \dots, D_2 = \bar{d}_2 \rangle\rangle$   
deux stratégies ne différant que par la décision  $D_2$

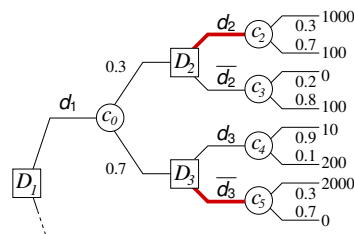
Alors  $E(S_1) - E(S_2) = E(\langle\langle D_2 = d_2 \rangle\rangle) - E(\langle\langle D_2 = \bar{d}_2 \rangle\rangle)$

$\Rightarrow$  Si  $E(\langle\langle D_2 = d_2 \rangle\rangle) \geq E(\langle\langle D_2 = \bar{d}_2 \rangle\rangle)$  alors  $E(S_1) \geq E(S_2)$

$\Rightarrow$  ne conserver que  $E(\langle\langle D_2 = d_2 \rangle\rangle)$  dans le nœud  $D_2$

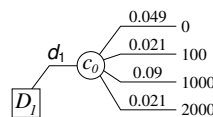
## Calculs dans un arbre de décision (4/5)

$\Rightarrow$  sur les nœuds de décision « terminaux », ne conserver que les meilleures décisions :



$\Rightarrow$  Si on choisit  $D_1 = d_1$ , la sous-stratégie optimale est forcément :  $D_2 = d_2, D_3 = d_3$  qui correspond à la loterie :

$\langle 0, 0.49; 100, 0.21; 1000, 0.09; 2000, 0.21 \rangle$

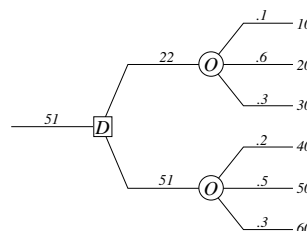


puis réitérer le process...

## Calculs dans un arbre de décision (5/5)

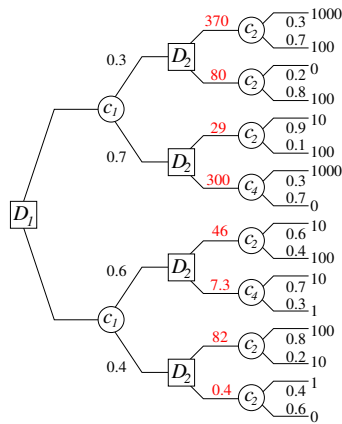
*Règle de calcul dans l'arbre de décision*

- 1 si le nœud est un nœud de chance, on calcule une espérance
- 2 si le nœud est un nœud de décision, on conserve le max



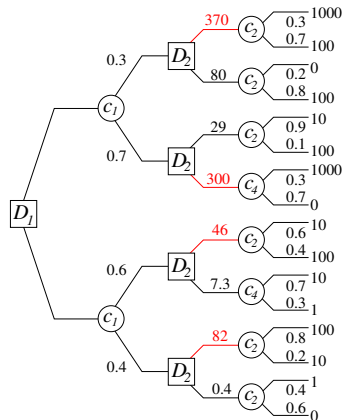
Méthode de calcul = inférence arrière

### Exemple d'inférence dans un arbre de décision (1/4)



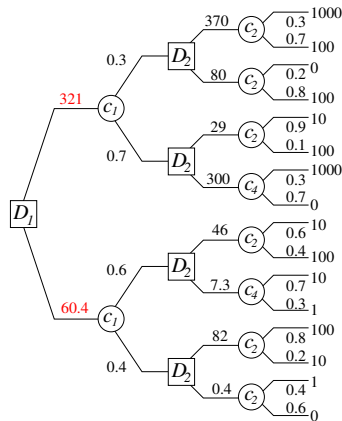
calcul :  $EU(C_2) = \sum_{C_2} P(C_2|D_1, C_1, D_2)u(D_1, C_1, D_2, c_2)$

### Exemple d'inférence dans un arbre de décision (2/4)



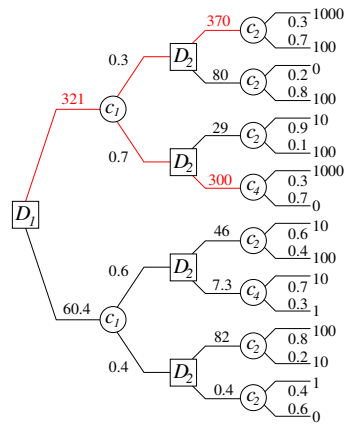
calcul :  $EU(D_2) = \max_{D_2} EU(C_2)$

### Exemple d'inférence dans un arbre de décision (3/4)



calcul :  $EU(C_1) = \sum_{C_1} P(C_1|D_1)EU(D_2)$

## Exemple d'inférence dans un arbre de décision (4/4)



$$\text{calcul : } EU(D_1) = \max_{D_1} EU(C_1)$$

## Exemple d'arbre de décision (1/2)

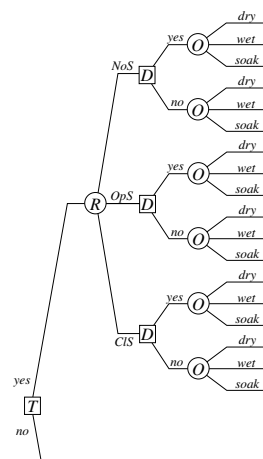
### Exemple classique de Raiffa (1968)

An oil wildcatter must decide either to drill (**yes**) or not drill (**no**). He is uncertain whether the hole is dry (**Dry**), wet (**Wet**) or soaking (**Soak**). At a cost of 10000\$, the wildcatter could take seismic soundings which help determine the geological structure at the site. The soundings will disclose whether the terrain below has no structure (**NoS**), that's bad, or open structure (**OpS**), or closed structure (**CIS**), (which is hopeful).

⇒ deux nœuds de décisions : test (T), forer (D)

⇒ deux nœuds de chance : résultat du test (R), quantité de pétrole (O)

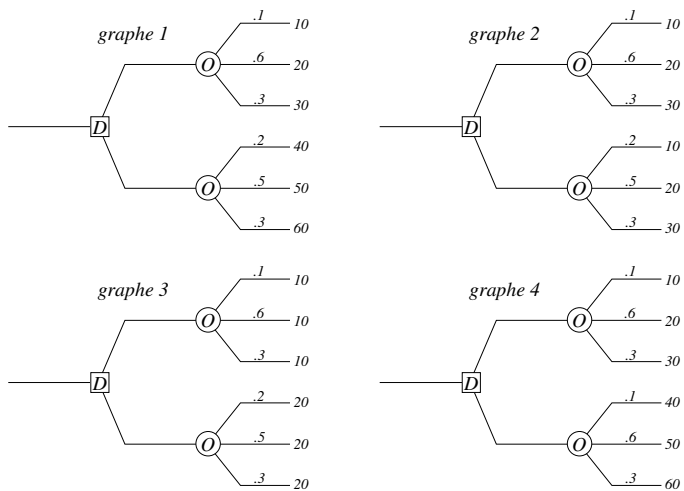
## Exemple d'arbre de décision (2/2)



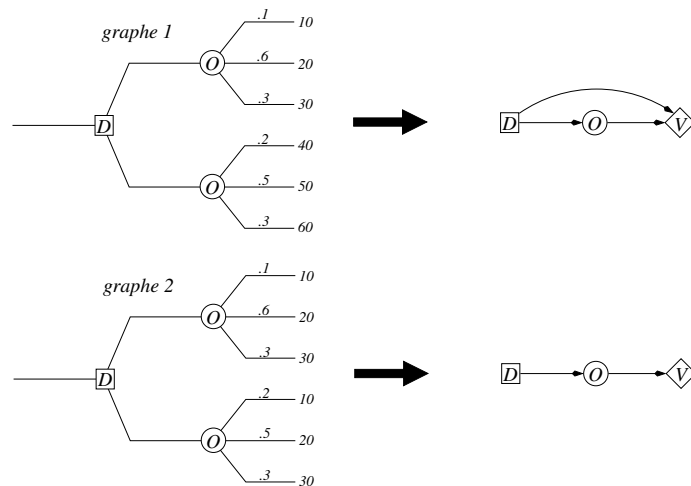
- 1 Les arbres sont trop compliqués pour le décideur  
 $\implies$  on ne voit pas la structure du problème
- 2 Les tables de probabilité dont on a besoin dans l'arbre ne sont pas forcément celles que l'on possède.
- 3 Il y a une explosion combinatoire au niveau des calculs  
 $\implies$  recherche des similitudes dans l'arbre

## 2 Les diagrammes d'influence

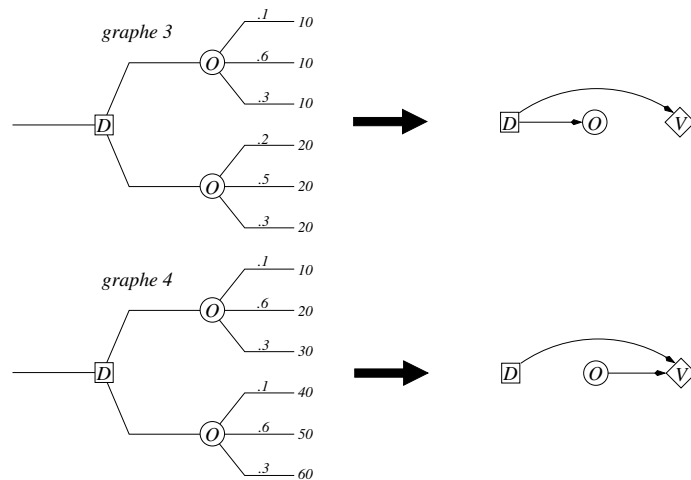
### structure des arbres de décision (1/3)



## structure des arbres de décision (2/3)



## structure des arbres de décision (3/3)



## Diagrammes d'influence et Réseaux de valuation

### Définition : diagramme d'influence

Un diagramme d'influence est un graphe orienté sans circuit (DAG) contenant trois types de nœuds :

- des nœuds de décision (les carrés) ;
- des nœuds de chance (probabilités), les cercles ;
- des nœuds d'utilité (les losanges).

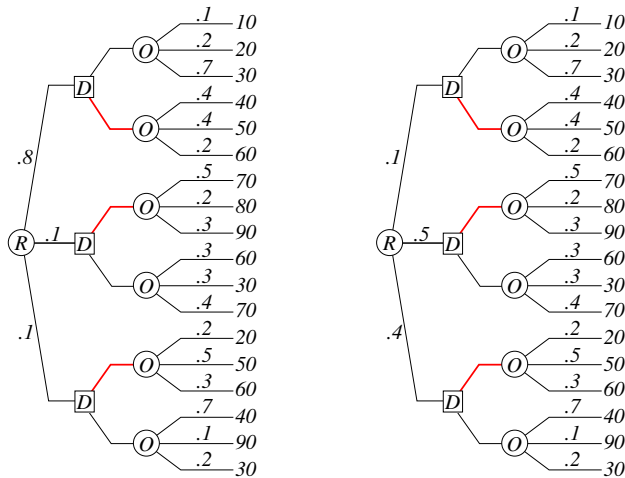
Les arcs vers les nœuds de décision  $D_i$  indiquent les infos connues par le décideur avant que la décision  $D_i$  ne soit prise. Tous les autres arcs indiquent des dépendances probabilistes.

### Définition : réseau de valuation

Un réseau de valuation est un diagramme d'influence dans lequel on a supprimé les arcs entrant dans les nœuds de décision.



## Pourquoi utiliser les réseaux de valuation ?



## Plusieurs nœuds d'utilité (1/5)

Soit un arbre de décision dans lequel on a 4 nœuds de décision  $D_i$ ,  $i \in \{1, 2, 3, 4\}$ , et 4 nœuds de chance  $c_i$ ,  $i \in \{1, 2, 3, 4\}$ .

$$\rho_4 = \max_{D_4} \sum_{c_4} P(c_4 | D_1, c_1, D_2, c_2, D_3, c_3, D_4) u(D_1, c_1, D_2, c_2, D_3, c_3, D_4, c_4)$$

$$\rho_3 = \max_{D_3} \sum_{c_3} P(c_3 | D_1, c_1, D_2, c_2, D_3) \max_{D_4} \sum_{c_4} P(c_4 | D_1, c_1, D_2, c_2, D_3, c_3, D_4) u(D_1, c_1, D_2, c_2, D_3, c_3, D_4, c_4)$$

$$= \max_{D_3} \sum_{c_3} \max_{D_4} \sum_{c_4} P(c_3, c_4 | D_1, c_1, D_2, c_2, D_3, D_4) u(D_1, c_1, D_2, c_2, D_3, c_3, D_4, c_4)$$

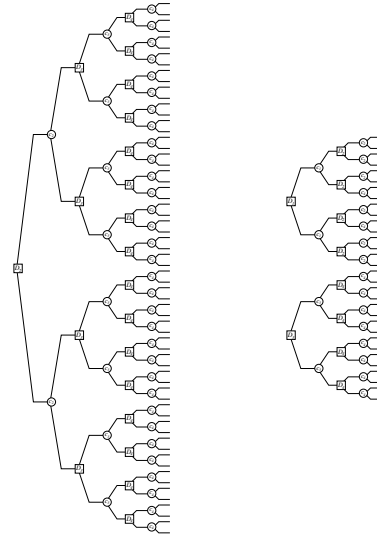
.....

## Plusieurs nœuds d'utilité (2/5)

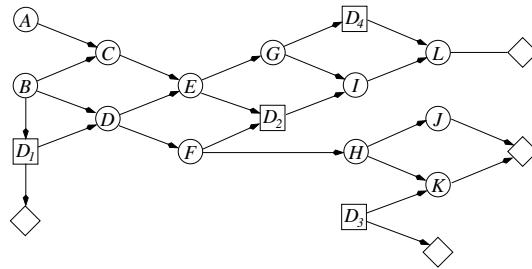
- Si  $u$  se sépare en 2 utilités :  $i = u_1 + u_2$ ,  
 $u_1$  ne dépend que des décisions  $D_1, D_2$ , et de  $c_1, c_2$ ,  
 $u_2$  ne dépend que des décisions  $D_3, D_4$ , et de  $c_3, c_4$ ,
- alors on peut améliorer les calculs.

$$\begin{aligned} \rho_4 &= \max_{D_4} \sum_{c_4} P(c_4 | D_1, c_1, D_2, c_2, D_3, c_3, D_4) u(D_1, c_1, D_2, c_2, D_3, c_3, D_4, c_4) \\ &= \max_{D_4} \sum_{c_4} P(c_4 | D_1, c_1, D_2, c_2, D_3, c_3, D_4) u_1(D_1, c_1, D_2, c_2) + \\ &\quad \max_{D_4} \sum_{c_4} P(c_4 | D_1, c_1, D_2, c_2, D_3, c_3, D_4) u_2(D_3, c_3, D_4, c_4) \\ &= u_1(D_1, c_1, D_2, c_2) + \max_{D_4} \sum_{c_4} P(c_4 | D_1, c_1, D_2, c_2, D_3, c_3, D_4) \\ &\quad u_2(D_3, c_3, D_4, c_4) \end{aligned}$$

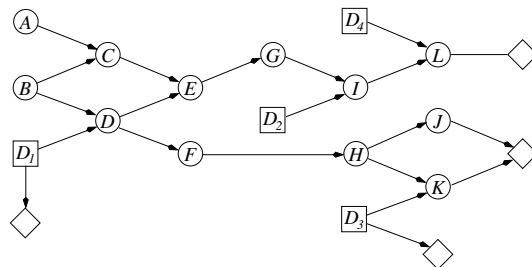
### Plusieurs nœuds d'utilité (3/5)



### Plusieurs nœuds d'utilité (4/5)



### Plusieurs nœuds d'utilité (5/5)



## Détermination de la stratégie optimale

Procédure similaire aux calculs dans les réseaux bayésiens :

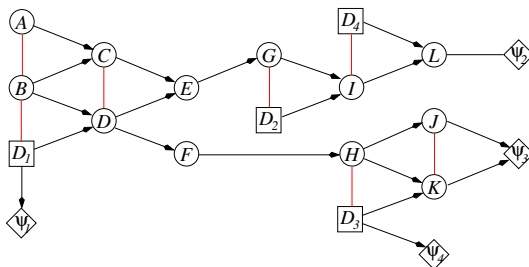
- 1 moraliser le réseau de valuation ;
- 2 supprimer les nœuds d'utilité (les losanges) ;
- 3 trianguler le réseau obtenu ;
- 4 créer un arbre de jonction ;
- 5 effectuer les calculs dans l'arbre de jonction.

**Problème** : contrairement aux réseaux bayésiens, pour la triangulation, la séquence d'élimination des nœuds est partiellement déterminée par des contraintes temporelles.

## La phase de moralisation

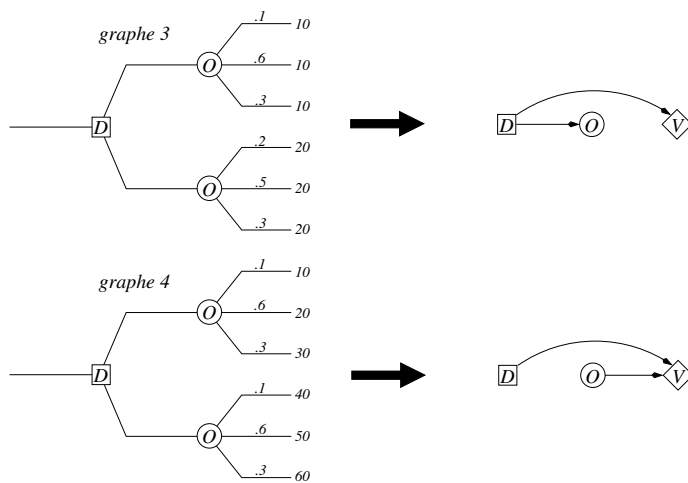
### Moralisation

Relier entre eux tous les parents d'un même nœud (que ce soient des nœuds de chance ou de décision).



⚠ il faut partir du réseau de valuation !!!

## Pourquoi doit-on être moral ?



## La triangulation, ça a du bon (1/2)

### Les contraintes temporelles

Appelons  $D_1, \dots, D_n$  les nœuds de décision  
 $C_1, \dots, C_k$  les nœuds de chance

On peut partitionner  $\{C_1, \dots, C_k\}$  en des ensembles disjoints  $I_0, I_1, \dots, I_n$  tels que  $I_k$  est l'ensemble des nœuds de chance observables entre les décisions  $D_k$  et  $D_{k+1}$ .

Cela induit un ordre partiel « temporel » sur les nœuds :

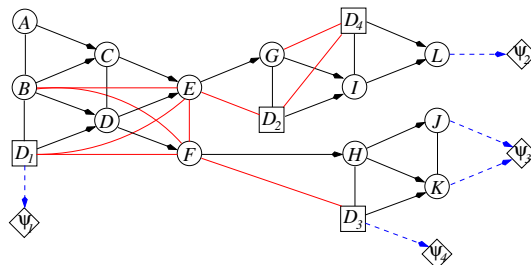
$$I_0 \prec D_1 \prec I_1 \prec \dots \prec D_n \prec I_n.$$

### Triangulation

N'importe quelle séquence d'élimination respectant l'ordre partiel  $\prec$  peut être utilisée (on doit d'abord éliminer les nœuds de  $I_n$ , puis de  $D_n$ , puis de  $I_{n-1}$ , etc.).

## La triangulation, ça a du bon (2/2)

- Après moralisation, on élimine les nœuds d'utilité ainsi que l'orientation des arcs.
- Ordre partiel :  $\{B\} \prec D_1 \prec \{E, F\} \prec D_2 \prec \emptyset \prec D_3 \prec \{G\} \prec D_4 \prec \{A, C, D, H, I, J, K, L\}$   
 $\Rightarrow$  ordre d'élimination :  $L, J, K, I, H, A, C, D, D_4, G, D_3, D_2, F, E, D_1, B$



## L'arbre de jonction (1/2)

### Strong junction tree

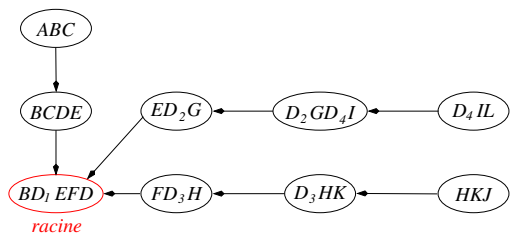
C'est un arbre de jonction (donc vérifiant la propriété d'intersection courante), possédant une clique  $R$  appelée racine telle que pour tout couple  $(C_1, C_2)$  de cliques adjacentes,  $C_1$  étant plus près de  $R$  que  $C_2$ , il existe un ordre sur les nœuds de  $C_2$  respectant l'ordre partiel  $\prec$  tel que les nœuds du séparateur  $C_1 \cap C_2$  précèdent ceux de  $C_2 \setminus C_1$  selon cet ordre.

### Proposition

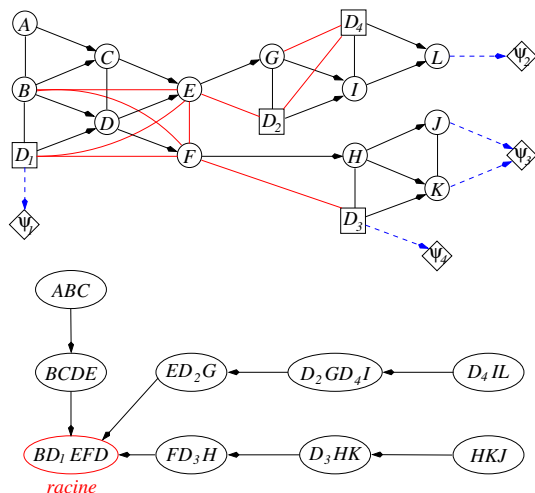
Dans un strong junction tree, les messages locaux à la Jensen ou Shafer-Shenoy permettent de calculer le maximum d'espérance d'utilité.

# L'arbre de jonction (2/2)

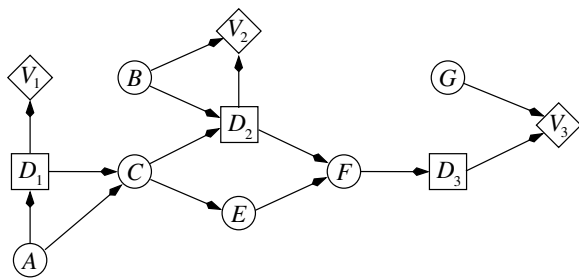
## Strong junction tree



# Arbre de jonction et calculs



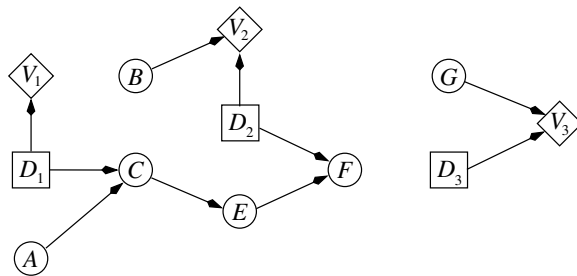
# Exemple de construction d'un strong junction tree (1)



ordre temporel :  $A \prec D_1 \prec \{B, C\} \prec D_2 \prec \{F\} \prec D_3 \prec \{E, G\}$

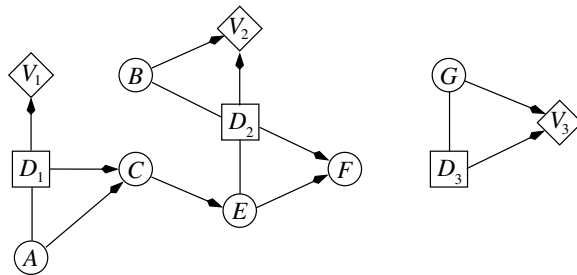
## Exemple de construction d'un strong junction tree (2)

Création du réseau de valuation :



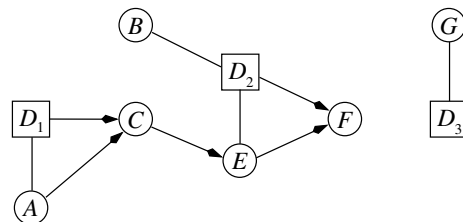
## Exemple de construction d'un strong junction tree (3)

Moralisation du réseau de valuation :



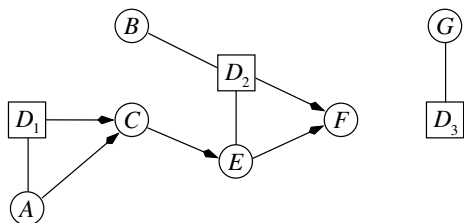
## Exemple de construction d'un strong junction tree (4)

Suppression des nœuds d'utilité :



## Exemple de construction d'un strong junction tree (5)

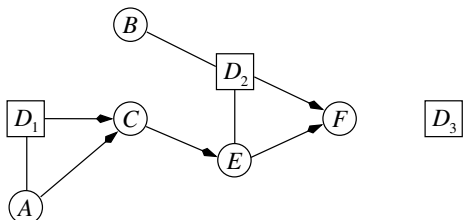
Triangulation : ordre :  $A \prec D_1 \prec \{B, C\} \prec D_2 \prec \{F\} \prec D_3 \prec \{E, G\}$



ordre compatible :  $A \prec D_1 \prec C \prec B \prec D_2 \prec F \prec D_3 \prec E \prec G$

## Exemple de construction d'un strong junction tree (6)

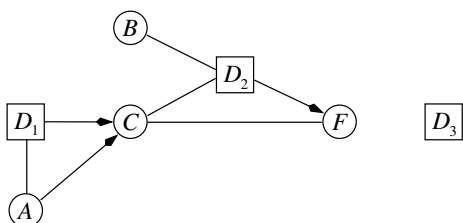
Suppression de  $G$  :



$D_3 G$

## Exemple de construction d'un strong junction tree (7)

Suppression de  $E$  :

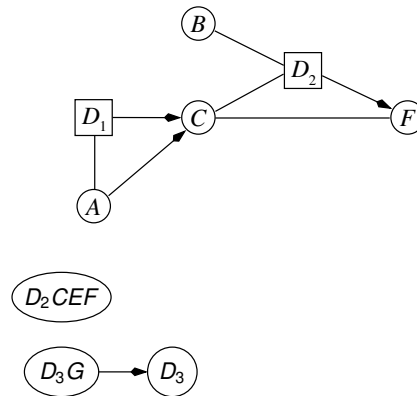


$D_2 CEF$

$D_3 G$

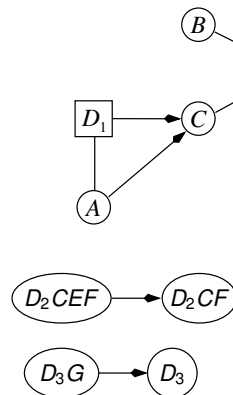
## Exemple de construction d'un strong junction tree (8)

Suppression de  $D_3$  :



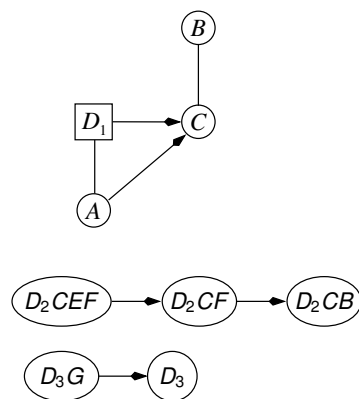
## Exemple de construction d'un strong junction tree (9)

Suppression de  $F$  :



## Exemple de construction d'un strong junction tree (10)

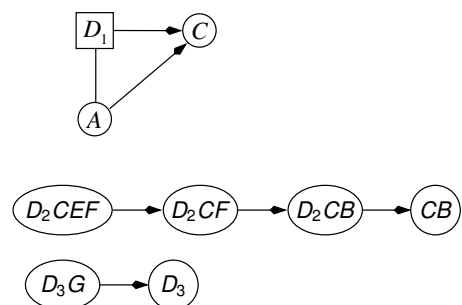
Suppression de  $D_2$  :





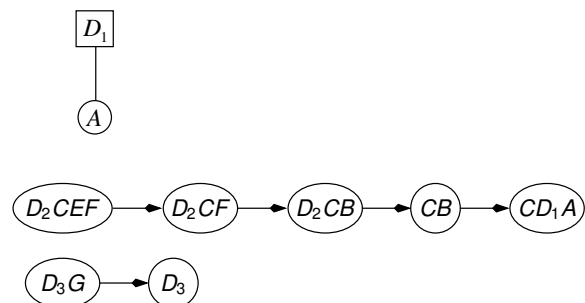
## Exemple de construction d'un strong junction tree (11)

Suppression de  $B$  :



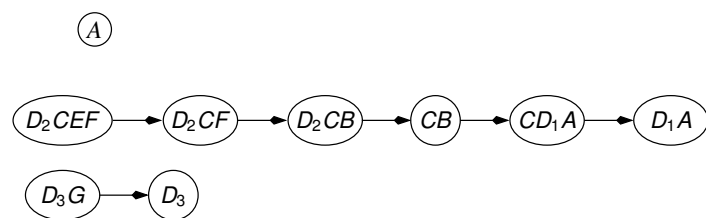
## Exemple de construction d'un strong junction tree (12)

Suppression de  $C$  :



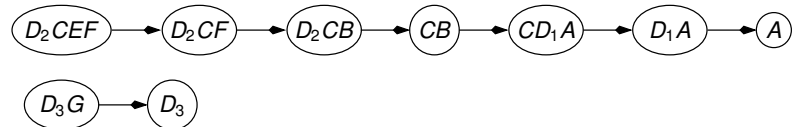
## Exemple de construction d'un strong junction tree (13)

Suppression de  $D_1$  :



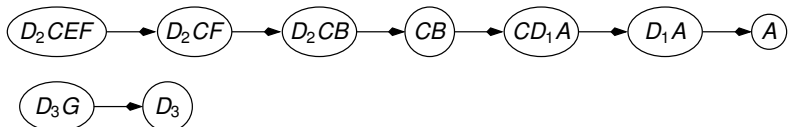
## Exemple de construction d'un strong junction tree (14)

Suppression de A :

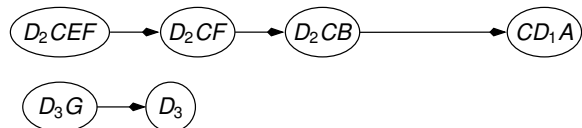


## Exemple de construction d'un strong junction tree (15)

Arbre d'élimination :



Strong junction tree :



## L'asymétrie, ça me scie

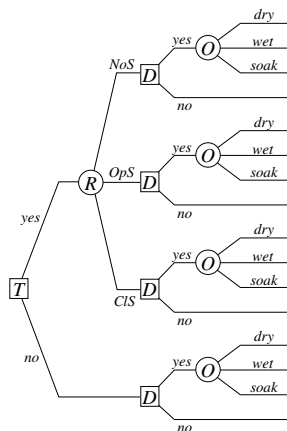
*Exemple classique de Raiffa (1968)*

An oil wildcatter must decide either to drill (yes) or not drill (no). He is uncertain whether the hole is dry (Dry), wet (Wet) or soaking (Soak). At a cost of 10000\$, the wildcatter could take seismic soundings which help determine the geological structure at the site. The soundings will disclose whether the terrain below has no structure (NoS), that's bad, or open structure (OpS), or closed structure (CIS), (which is hopeful).

⇒ deux noeuds de décisions : test (T), forer (D)

⇒ deux noeuds de chance : résultat du test (R), quantité de pétrole (O)

## Les arbres de décisions asymétriques (1/2)



### Asymétrie (définition de Shenoy) :

Dans un arbre de décision, un chemin de la racine vers une feuille est appelé un scénario.

Un problème de décision est asymétrique si le nombre de scénarios est inférieur au produit du nombre d'états de toutes les variables.

## Les arbres de décisions asymétriques (2/2)

⇒ l'asymétrie est encodée de manière « naturelle ».

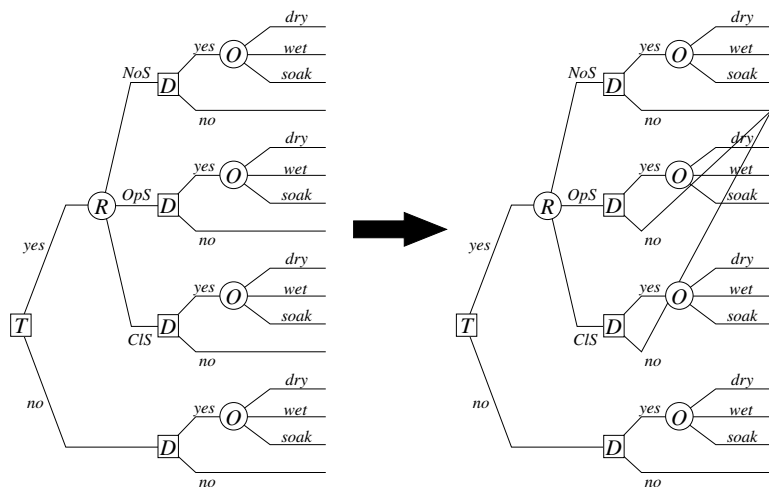
Avantage des arbres des décision : on ne rajoute pas de noeuds ou de valeurs factices (dummy nodes).

Problème : l'asymétrie est gérée de manière globale, pas locale

⇒ il n'est pas facile de voir la structure du problème de décision.

⇒ il n'est pas facile de retrouver les symétries restantes.

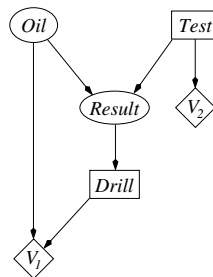
## Coalescence dans les arbres de décision



### Plusieurs manières de traiter les asymétries

- Smith, Holtzman & Matheson (93) :  
Partie graphique : diagramme d'influence classique  
Distributions conditionnelles : décrites à l'aide d'une *distribution tree* (similaire à un arbre de décision).
- Factorisation des distributions conditionnelles (Cowell, Lauritzen, Lauritzen, Spiegelhalter (99)).
- réseaux de valuation asymétriques (Shenoy, Jensen).

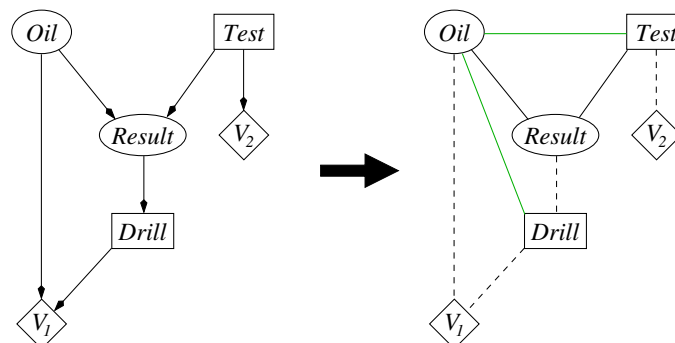
## Factorisation des distributions conditionnelles (1/5)



En principe, le noeud/variable *Result* ne devrait avoir que trois valeurs : *NoS*, *OpS* et *CIS*. Pour symétriser le diagramme, on peut rajouter une quatrième valeur : *NoResult*.

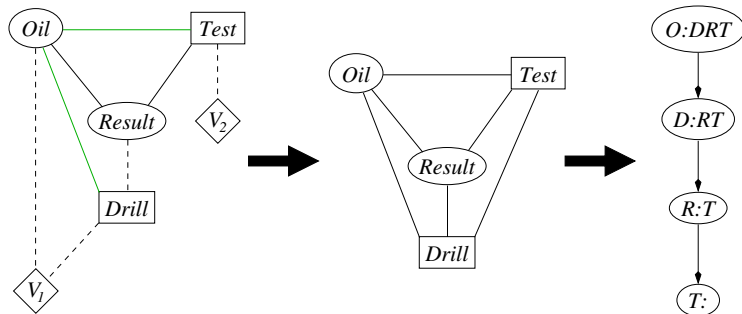
## Factorisation des distributions conditionnelles (2/5)

Construction du strong junction tree : 1/ on élimine les arcs entrant dans les noeuds de décision et on moralise.



### Factorisation des distributions conditionnelles (3/5)

Triangulation : séquence  $O, D, R, T$



### Factorisation des distributions conditionnelles (4/5)

Dans la clique  $ODRT$ , on stocke la fonction d'utilité  $V_1$  et la proba  $P(R|O, T)$

NoS	OpS	CIS	NoResult	Test	Oil
0,6	0,3	0,1	0	yes	dry
0,3	0,4	0,3	0		wet
0,1	0,4	0,5	0		soak
0	0	0	1	no	dry
0	0	0	1		wet
0	0	0	1		soak

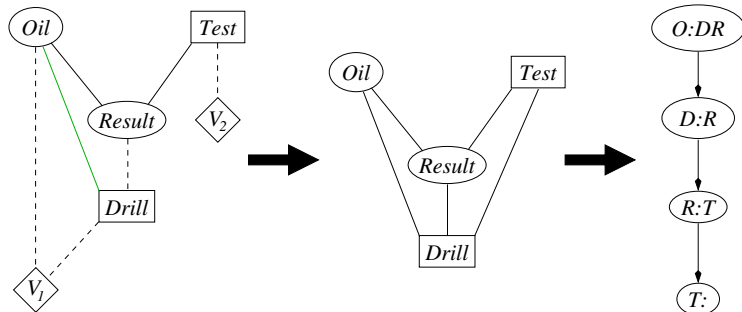
$\Rightarrow P(R|O, T) = f(R, T)g(R, O) :$

T	NoS	OpS	CIS	NoResult	O	NoS	OpS	CIS	NoResult
yes	1	1	1	0	dry	0,6	0,3	0,1	1
wet					wet	0,3	0,4	0,3	1
no	0	0	0	1	soak	0,1	0,4	0,5	1

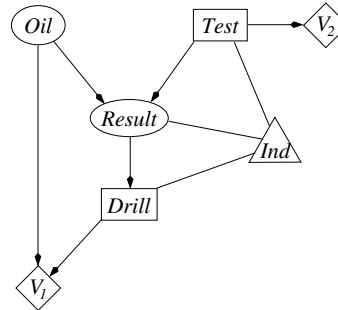
### Factorisation des distributions conditionnelles (5/5)

$\Rightarrow$  on n'a plus besoin de relier les noeuds  $Test$  et  $Oil$  par moralisation.

$\Rightarrow$  le strong junction tree est :



## Les réseaux de valuation asymétriques (Shenoy 1/2)



Le triangle : appelé indicateur par Shenoy (double-triangle chez lui). Il représente la contrainte que le résultat du test n'est connu de la décision *Drill* que si le test a été effectué.

## Les réseaux de valuation asymétriques (Shenoy 2/2)

Table stockée dans le noeud *Ind* : la liste des valeurs des triplets (*Test, Result, Drill*) admissibles.

⇒ {(yes, NoS, yes), (yes, OpS, yes), (yes, CIS, yes),  
(yes, NoS, no), (yes, OpS, no), (yes, CIS, no), (no, NoResult, yes),  
(no, NoResult, no)}

⇒ pour faire la propagation, on utilise l'algorithme de Shafer-Shenoy. On prend en compte l'indicateur en se limitant à ses triplets de valeurs.

## Quelques références bibliographiques (symétrie)

- Howard, R.A. & Matheson, J.E. (1984) *Influence Diagrams*, dans Readings on the Principles and Applications of Decision Analysis, vol 2, Howard, R.A. & Matheson, J.E. (eds), pp.719–762.
- Shachter, R. (1986) *Evaluating Influence Diagrams*, Operations Research, vol. 34, pp.871–882.
- Shenoy, P.P. (1992) *Valuation-based systems for Bayesian decision analysis*, Operations Research, vol 40, pp.463–484.
- Jensen, F. & Jensen, F.V. & Dittmer, S.L. (1994) *From Influence Diagrams to Junction Trees*, UAI-94.
- Cowell, R.G. & Dawid, A.P. & Lauritzen, S.L. & Spiegelhalter, D.J. (1999) *Probabilistic Networks and Expert Systems*, Statistics for Engineering and Information Science, Springer.

## Quelques références bibliographiques (asymétrie)

- Bielza, C & Shenoy, P.P (1999) A comparison of graphical techniques for asymmetric decision problems, *Management Science*, vol 45, pp.1552–1569.
- Shenoy, P.P. (2000) *Valuation network representation and solution of asymmetric decision problems*, *European journal of Operations Research*, vol 121, pp.579–608.
- Cowell, R.G. & Dawid, A.P. & Lauritzen, S.L. & Spiegelhalter, D.J. (1999) *Probabilistic Networks and Expert Systems*, *Statistics for Engineering and Information Science*, Springer.
- Nielsen, T.D. & Jensen, F.V. (1999) *Welldefined Decision Scenarios*, UAI-99.
- Nielsen, T.D. & Jensen, F.V. (2000) *Representing and solving asymmetric Bayesian decision problems*, UAI-2000.

## Modèle graphique pour des préférences ordinales

## Indépendances conditionnelles (1/2)

Rappel : Structure des préférences  $\implies$  indépendances

### Indépendance préférentielle

- $I, J$  = une partition de  $\{1, \dots, n\}$ ,  $I \neq \emptyset$  et  $J \neq \emptyset$
- $Y = \prod_{i \in I} X_i$  et  $Z = \prod_{j \in J} X_j$ .
- $Y$  est préférentiellement indépendant de  $Z$  si et seulement si pour tout  $y_1, y_2 \in Y, z_1, z_2 \in Z$ ,

$$(y_1, z_1) \succ_X (y_2, z_1) \iff (y_1, z_2) \succ_X (y_2, z_2).$$

On dit alors que  $y_1$  est préféré à  $y_2$  *ceteris paribus*.

## Indépendances conditionnelles (2/2)

### Indépendance conditionnelle préférentielle

- $I, J, K =$  une partition de  $\{1, \dots, n\}$ ,  $I \neq \emptyset, J \neq \emptyset, K \neq \emptyset$ .
- $Y = \prod_{i \in I} X_i, Z = \prod_{j \in J} X_j$  et  $T = \prod_{k \in K} X_k$ .
- $Y$  est préférentiellement indépendant de  $Z$  conditionnellement à une valeur  $t \in T$  si et seulement si pour tout  $y_1, y_2 \in Y, z_1, z_2 \in Z$ ,  
 $(y_1, t, z_1) \succsim_X (y_2, t, z_1) \iff (y_1, t, z_2) \succsim_X (y_2, t, z_2)$ .

### Indépendance préférentielle de $Y$ et $Z$ conditionnellement à $T$

$Y$  est préférentiellement indépendant de  $Z$  conditionnellement à  $T$  si et seulement si il l'est pour toute valeur  $t \in T$ .

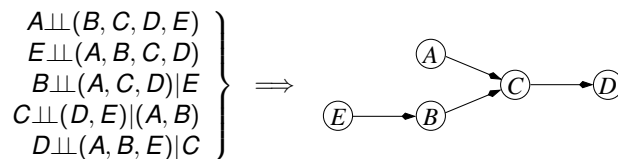
## CP-nets (1/3)

### Définition : CP-net

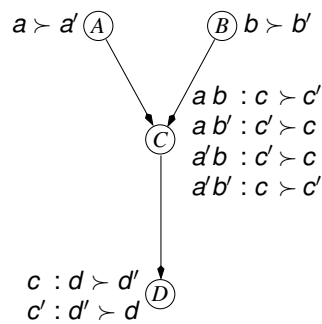
Un CP-net est un graphe (pas forcément acyclique) tel que tout nœud/attribut est préférentiellement indépendant des autres nœuds/attributs conditionnellement à ses parents dans le graphe.

CP = Ceteris Paribus ou « Conditional preferential ».

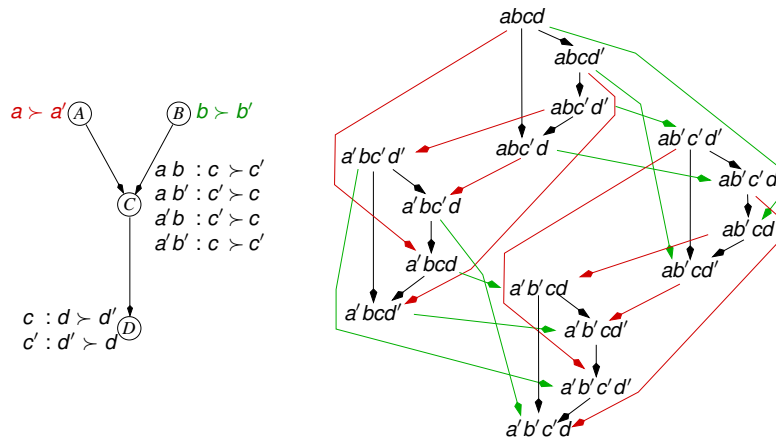
$\implies$  CP-net = notion ordinale



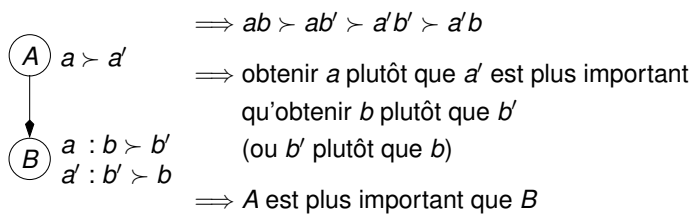
## CP-nets (2/3)







### Importance des parents sur les enfants



Dans un CP-net, les préférences sur les parents ont une priorité plus importante que celles sur les enfants

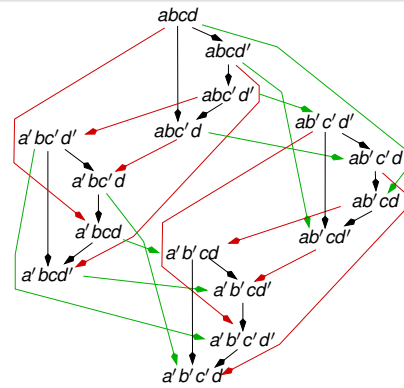
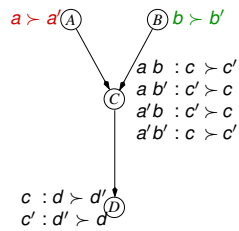
### À quoi servent les CP-nets ?

- optimisation :  $\max_{x \sim x} \{x\}$
- calculs de dominance :  $x \stackrel{?}{\succ} x y$

**Mais** décision dans le risque/l'incertain  $\implies$  diagramme d'influence

- élicitation

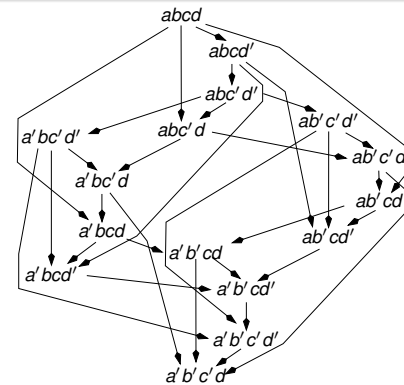
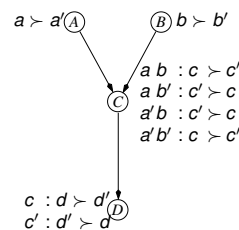
## Optimisation avec les CP-nets



Algorithme pour déterminer l'élément préféré du décideur

- 1 Choisir la meilleure alternative pour les nœuds sans parents
- 2 Pour chaque nœud dont les parents ont été instanciés, choisir la meilleure alternative

## Calcul de dominance avec les CP-nets (1/5)

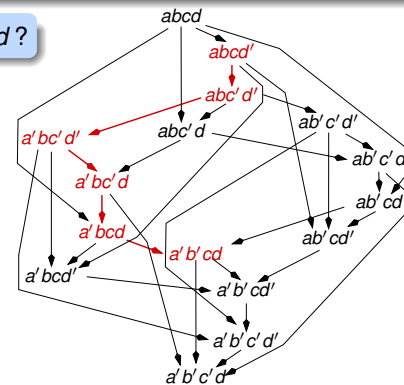
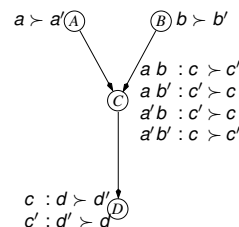


Définition : les flips

Pour tout arc  $X \rightarrow Y$  du graphe des préférences en extension (à droite),  $X$  et  $Y$  ne diffèrent que par la valeur d'un seul attribut. Passer de  $X$  à  $Y$  (ou de  $Y$  à  $X$ ) s'appelle un **flip**.

## Calcul de dominance avec les CP-nets (2/5)

Est-ce que  $abcd' > a'b'cd$  ?



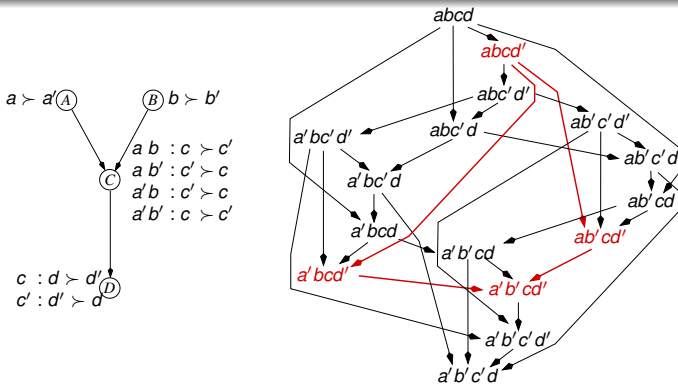
Algorithmes pour déterminer si  $x > y$

- 1 Trouver une « séquence améliorante de flips » permettant de passer de  $y$  à  $x$  = un chemin que l'on remonte de  $y$  vers  $x$
- 2 Trouver une « séquence détériorante de flips » permettant de passer de  $x$  à  $y$  = un chemin que l'on descend de  $x$  vers  $y$

## Calcul de dominance avec les CP-nets (3/5)

### Règle du « suffixe fixé »

- $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n)$ , est-ce que  $x \succ y$  ?
- $x_j = y_j, j = i, \dots, n \implies$  s'il existe une séquence améliorante de flips de  $y$  à  $x$ , il en existe une pour laquelle les valeurs des attributs  $X_j, j = i, \dots, n$ , restent fixées à  $x_j = y_j$ .



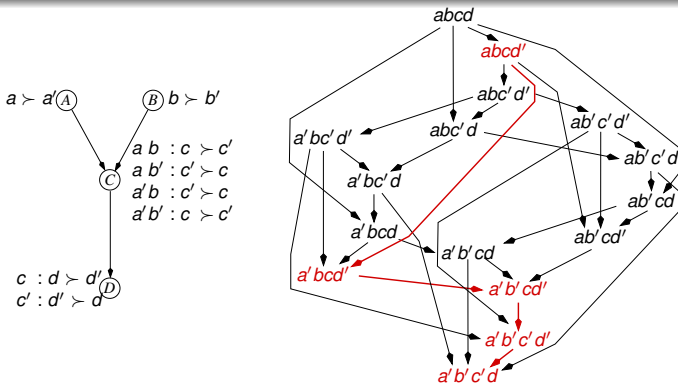
Diagrammes d'influence et CP-nets

79/83

## Calcul de dominance avec les CP-nets (4/5)

### Règle de « l'extension du suffixe »

- $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n)$ , est-ce que  $x \succ y$  ?
- $x_j = y_j, j = i, \dots, n$  et  $\exists$  un flip améliorant de  $y_{i-1}$  à  $y'_{i-1} = x_{i-1} \implies$  si  $\exists$  une séquence améliorante de flips de  $y$  à  $x$ , il en existe une t.q. les valeurs des attributs  $X_j, j = i - 1, \dots, n$ , sont fixées à  $x_j = y_j$ .



Diagrammes d'influence et CP-nets

80/83

## Calcul de dominance avec les CP-nets (5/5)

### Algorithme général de construction de séquences de flips

- 1 appliquer, si possible, la règle du « suffixe fixé »
- 2 appliquer, si possible, la règle de « l'extension du suffixe »
- 3 essayer (avec backtrack possible) de faire des flips (améliorants) avec les attributs les plus bas dans le CP-net
- 4 en conjonction avec 3, faire du pruning

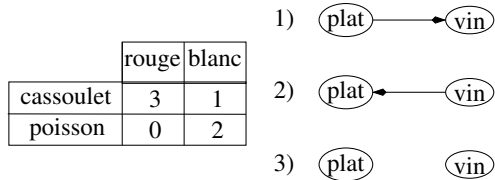
Diagrammes d'influence et CP-nets

81/83

## Inconvénient culinaire des CP-nets

- Repas = Plat de résistance + vin + dessert  
plat  $\in$  {cassoulet, poisson}  
vin  $\in$  {vin rouge, vin blanc}  
dessert  $\in$  {rien, profiterole au chocolat}

- $u(\text{repas}) = u_1(\text{plat}, \text{vin}) + u_2(\text{plat}, \text{dessert})$   
(cassoulet,rouge)  $\succ_X$  (poisson,blanc)  $\succ_X$  (cassoulet,blanc)  $\succ_X$  (poisson,rouge)



$\Rightarrow$  non représentable par un CP-net (compact)

## Biblio sur les modèles graphiques de « préférences »

### Références

- C Boutilier, R Brafman, H Holger et D Poole (1999) "Reasoning with Conditional Ceteris Paribus Preference Statements", Proceedings d'UAI-99
- R Brafman, C Domshlak (2002) "Introducing variable importance tradeoffs into CP-nets", Proceedings d'UAI-02
- F Bacchus et A Grove (1995). "Graphical models for preference and utility", Proceedings d'UAI-95
- C Boutilier, F Bacchus et R Brafman (2001) "UCP-Networks : A directed graphical representation of conditional utilities", Proceedings d'UAI-01

## Modèles graphiques pour les préférences

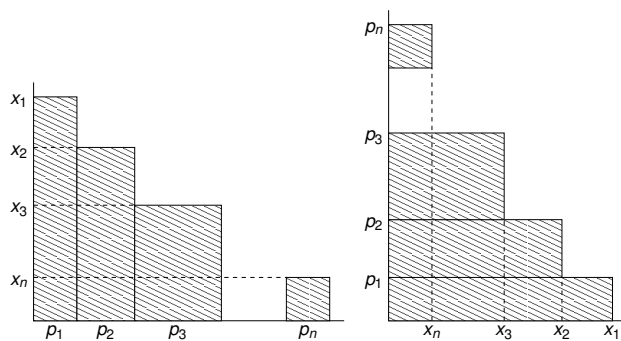
Christophe Gonzales

LIP6 – Paris 6 Université, France

### Dualité probabilités/utilités (1/2)

*Décision dans le risque ou l'incertain*

Espérance d'utilité ou de gain :  $\sum_{i=1}^n p_i x_i$ .



### Dualité probabilités/utilités (2/2)

*Probabilités et réseaux bayésiens*

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)).$$

$$\stackrel{\text{Markov}}{\implies} P(X_1, \dots, X_n) = \prod_{i=1}^n f_i(X_i, Pa(X_i)).$$

$$\implies \log P(X_1, \dots, X_n) = \sum_{i=1}^n g_i(X_i, Pa(X_i))$$

$\implies$  décomposition additive généralisée.

### Dualité probabilités/utilités :

décomposition d'une probabilité jointe  $\implies$  modèle graphique



décomposition d'une utilité  $\implies$  modèle graphique

## Indépendances conditionnelles

- $P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i))$   
 $\implies$  indépendances conditionnelles
- $u(X_1, \dots, X_n) = \sum_{i=1}^n u(X_i | Pa(X_i))$   
 $\implies$  indépendances conditionnelles

indépendances conditionnelles  $\implies$  structure des préférences

### Décision dans le risque/l'incertain

$$\langle P_i, x^i \rangle \succsim \langle Q_i, y^i \rangle \iff \sum_i P_i u(x^i) \geq \sum_i Q_i u(y^i).$$

$\succsim$  : relation de préférences sur les loteries.

$u : \mathcal{X} \mapsto \mathbb{R}$  : utilité sur les conséquences.

$\mathcal{X}$  : espace des conséquences.

$\mathcal{X} = X_1 \times X_2 \times \dots \times X_n$ .

$X_i$  = attribut.

$\succsim_{\mathcal{X}}$  : relation de préférences sur les conséquences.

## Espace des conséquences = produit cartésien

### Structure des conséquences

- $\mathcal{X}$  = produit cartésien  $X_1 \times \dots \times X_n$
- choisir un menu  
 $\implies$  menu = (plat principal, boisson, dessert)

### Exemple

- $\mathcal{X}$  espace des menus  $x = (x_1, x_2, x_3)$ ,
- $x_1 \in X_1 = \{\text{viande (V), poisson (P)}\}$ ,
- boisson  $x_2 \in X_2 = \{\text{vin rouge (R), vin blanc (B)}\}$ ,
- dessert  $x_3 \in X_3 = \{\text{cake (C), sorbet (S)}\}$ .

## Structure des préférences (1/5)

- $x_1 \in X_1 = \{\text{viande (V), poisson (P)}\}$ ,
- boisson  $x_2 \in X_2 = \{\text{vin rouge (R), vin blanc (B)}\}$ ,
- dessert  $x_3 \in X_3 = \{\text{cake (C), sorbet (S)}\}$ .

### Décomposition additive

- préférences lexicographiques :  
 $V \succ_{X_1} P$ , puis  $R \succ_{X_2} B$ , puis  $C \succ_{X_3} S$
  - $(V, R, C) \succ_X (V, R, S) \succ_X (V, B, C) \succ_X (V, B, S) \succ_X (P, R, C) \succ_X (P, R, S) \succ_X (P, B, C) \succ_X (P, B, S)$
- $\succ_X$  est représentable par une fonction d'utilité additive  $\sum_{i=1}^3 u_i(x_i)$ .

La décomposition la plus simple et la plus efficace

## Structure des préférences (2/5)

### Définition : utilité additive

$u : X \mapsto \mathbb{R}$  est une utilité additive ssi il existe des  $u_i : X_i \mapsto \mathbb{R}$  telles que  $u(x_1, \dots, x_n) = \sum_{i=1}^n u_i(x_i)$ . Donc :

$$(x_1, \dots, x_n) \succsim_X (y_1, \dots, y_n) \iff \sum_{i=1}^n u_i(x_i) \geq \sum_{i=1}^n u_i(y_i).$$

### conséquence :

$$\begin{aligned} (x_1, x_2) \succsim_X (y_1, x_2) &\iff u_1(x_1) + u_2(x_2) \geq u_1(y_1) + u_2(x_2) \\ &\iff u_1(x_1) \geq u_1(y_1) \\ &\iff u_1(x_1) + u_2(y_2) \geq u_1(y_1) + u_2(y_2) \\ &\iff (x_1, y_2) \succsim_X (y_1, y_2) \end{aligned}$$

décomposition additive  $\implies$  forme d'« indépendance » entre les attributs

## Structure des préférences (3/5)

- $x_1 \in X_1 = \{\text{viande (V), poisson (P)}\}$ ,
- boisson  $x_2 \in X_2 = \{\text{vin rouge (R), vin blanc (B)}\}$ ,
- dessert  $x_3 \in X_3 = \{\text{cake (C), sorbet (S)}\}$ .

### Décomposition additive plus « sophistiquée »

- priorités :  
 haute priorité accordée à l'accord entre le vin et le plat,  
 puis la viande est préférée au poisson,  
 puis le cake est préféré au sorbet
  - $(V, R, C) \succ_X (V, R, S) \succ_X (P, B, C) \succ_X (P, B, S) \succ_X (V, B, C) \succ_X (V, B, S) \succ_X (P, R, C) \succ_X (P, R, S)$
- $\succ_X$  est représentable par une utilité additive  $u_{1,2}(x_1, x_2) + u_3(x_3)$

Pas d'attribut en commun entre  $u_{1,2}$  et  $u_3$

## Structure des préférences (4/5)

- repas :  $X_1 = \text{plat}$     $X_2 = \text{vin}$

(viande, vin rouge)  $\succsim_X$  (poisson, vin rouge)  $\not\Leftarrow$   
(viande, vin blanc)  $\succsim_X$  (poisson, vin blanc)

$\Rightarrow (x_1, x_2) \succsim_X (y_1, x_2) \not\Leftarrow (x_1, y_2) \succsim_X (y_1, y_2)$

$\Rightarrow$  préférences non représentables par une utilité additive

- Explication : dépendance entre les attributs

## Structure des préférences (5/5)

- $x_1 \in X_1 = \{\text{viande } (V), \text{poisson } (P)\}$ ,
- boisson  $x_2 \in X_2 = \{\text{vin rouge } (R), \text{vin blanc } (B)\}$ ,
- dessert  $x_3 \in X_3 = \{\text{cake } (C), \text{sorbet } (S)\}$ .

### Décomposition additive généralisée

- haute priorité accordée à l'accord entre le vin et le plat, puis la viande est préférée au poisson, puis le cake est préféré au sorbet si le plat est du poisson mais c'est l'inverse si le plat est de la viande

- $(V, R, S) \succ_X (V, R, C) \succ_X (P, B, C) \succ_X (P, B, S) \succ_X$   
 $(V, B, S) \succ_X (V, B, C) \succ_X (P, R, C) \succ_X (P, R, S)$

$\succsim_X$  est représentable par une utilité GAI-décomposable  $u_{1,2}(x_1, x_2) + u_{1,3}(x_1, x_3)$ .

L'attribut  $X_1$  appartient à  $u_{1,2}$  et  $u_{1,3}$

## Décomposition additive généralisée (GAI)

### Définition

- $\mathcal{X} = X_1 \times X_2 \times \dots \times X_n$
- $Z_1, \dots, Z_k \subseteq N = \{1, \dots, n\}$  tels que  $N = \cup_{i=1}^k Z_i$
- $X_{Z_i} = \prod_{j \in Z_i} X_j$
- $X_{Z_1}, \dots, X_{Z_k}$  sont GAI-indépendants pour  $\succsim_X$  ssi il existe des fonctions  $u_i : X_{Z_i} \mapsto \mathbb{R}$  telles que

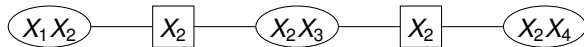
$$u(x) = \sum_{i=1}^k u_i(x_{Z_i}), \forall x = (x_1, \dots, x_n) \in \mathcal{X}.$$

### Exemple

- $u(x_1, x_2, x_3, x_4) = u_1(x_1) + u_2(x_2, x_3) + u_3(x_3, x_4)$
- dépendance entre les attributs



- $u(x_1, x_2, x_3, x_4) = u_1(x_1, x_2) + u_2(x_2, x_3) + u_3(x_2, x_4)$



ellipse = clique      rectangle = séparateur

- *Propriété d'intersection courante* :  
Pour tout couple de cliques  $(C_1, C_2)$  avec une intersection  $S$  non vide,  $S$  est contenu dans toute clique et tout séparateur sur la chaîne entre  $C_1$  et  $C_2$
- Similaire aux arbres de jonction pour les réseaux bayésiens

## Principaux problèmes relatifs aux préférences

- *Élicitation de préférences*
- *Prise de décision*
  - 1 *optimisation globale* : trouver l'élément préféré dans l'ensemble de conséquences  $\mathcal{X}$ .
  - 2 *optimisation sous contraintes* : trouver l'élément préféré conditionnellement à des valeurs de certains attributs.
  - 3 *comparaisons d'éléments* : pour un couple  $(x, y) \in \mathcal{X} \times \mathcal{X}$ , trouver si  $x$  est préféré à  $y$ , ou bien l'inverse.

Problème : en pratique,  $\mathcal{X}$  est un espace combinatoire  
 $\Rightarrow$  explosion combinatoire (en temps et mémoire)

## Élicitation de préférences

## Questions : élicitation dans le risque/l'incertain

- décision  $d \iff$  loterie  $\langle p_1, x^1; \dots; p_n, x^n \rangle$ 
  - $x^i$  : conséquence ou résultat
  - $p_i$  : probabilité que la décision  $d$  ait pour conséquence  $x^i$
- le décideur est maximisateur d'espérance d'utilité :  
 $\langle P_i, x^i \rangle \succsim \langle Q_i, y^i \rangle \iff \sum_i P_i u(x^i) \geq \sum_i Q_i u(y^i)$
- $\langle 1, z \rangle \sim \langle p, y; 1-p, x \rangle$   

$$z \sim \begin{array}{l} p \text{---} y \\ \swarrow \quad \searrow \\ \quad \quad x \\ \uparrow \quad \downarrow \\ 1-p \end{array} \implies u(z) = pu(y) + (1-p)u(x)$$

$\implies$  l'utilité sur les résultats est unique à une transformation affine strictement positive près

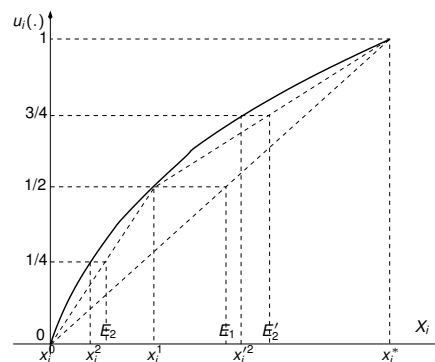
## Élicitation d'une utilité additive (1/2)

- *Élicitation dans le risque/l'incertain* : demander  $p$  tel que  
 $\langle 1, z \rangle \sim \langle p, y; 1-p, x \rangle$   

$$z \sim \begin{array}{l} p \text{---} y \\ \swarrow \quad \searrow \\ \quad \quad x \\ \uparrow \quad \downarrow \\ 1-p \end{array} \implies u(z) = pu(y) + (1-p)u(x)$$
- *Alternative* : demander  $z$  tel que :  $\langle 1, z \rangle \sim \langle \frac{1}{2}, y; \frac{1}{2}, x \rangle$   
 $\implies u(z) = \frac{1}{2}u(y) + \frac{1}{2}u(x)$   
 $\implies u(z) - u(y) = u(x) - u(z)$

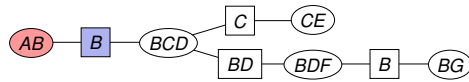
## Élicitation d'une utilité additive (2/2)

$$u(x_1, x_2) = u_1(x_1) + u_2(x_2)$$



on pose  $u_i(x_i^0) = 0$  et  $u_i(x_i^*) = 1$

## Élicitation de la sous-utilité d'une clique (1/3)



$$u_1(A, B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

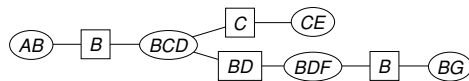
Pour toute valeur fixée  $b$  de  $B$ , utilité additive :

$$u_1^b(A) + [u_2(C, E) + u_3^b(C, D) + u_4^b(D, F) + u_5^b(G)]$$

- *Élicitation dans le certain* : construire  $u_1(\cdot)$  en utilisant des séquences standards
- *Élicitation dans le risque/l'incertain* : demander  $p$  tel que  $\langle 1, z \rangle \sim \langle p, y; 1 - p, x \rangle$

$$z \sim \begin{array}{l} p \text{ } y \\ \swarrow \quad \searrow \\ 1-p \text{ } x \end{array} \implies u(z) = pu(y) + (1-p)u(x)$$

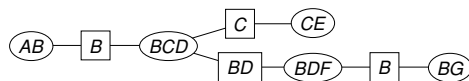
## Élicitation de la sous-utilité d'une clique (2/3)



Connaissant  $u_1(A, b^0)$  et  $v_1(A, b)$ ,

comment construire  $u_1$  sur  $A \times \{b^0, b\}$  ?

## Unicité des utilités GAI-décomposables



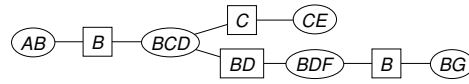
$$u_1(A, B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

### Proposition

- $(a^0, b^0, c^0, d^0, e^0, f^0, g^0)$  un n-uplet choisi arbitrairement
- solvabilité + essentialité + connexité des séparateurs  $\implies$  l'utilité GAI-décomposable  $u(\cdot)$  telle que :
  - $u_1(a^0, b) = 0 \forall b$
  - $u_2(c, e^0) = 0 \forall c$
  - $u_5(b, g^0) = 0 \forall b$
  - $u_4(b, d, f^0) = 0 \forall b, d$
  - $u_3(b^0, c^0, d^0) = 0$

est unique à une transformation linéaire strictement positive près.

## Élicitation de la sous-utilité d'une clique (3/3)



Connaissant  $u_1(A, b^0)$  et  $v_1(A, b)$ , comment construire  $u_1$  sur  $A \times \{b^0, b\}$  ?

Proposition  $\implies u_1(a^0, b^0) = v_1(a^0, b) = 0$   
 $\implies$  trouver  $k$  tel que  $u_1(A, b) = k \times v_1(A, b^0)$

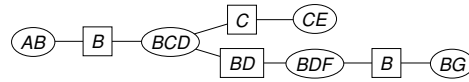
1 demander  $p'$  t.q.  $(a^0, b, c^0, d^0, e^0, f^0) \sim \frac{p'}{1-p'} (a, b^0, c^0, d^0, e^0, f^0)$

$$u_1(a^0, b) + u_2(c^0, e^0) + u_3(b, c^0, d^0) + u_4(b, d^0, f^0) + u_5(b, g^0) = p'$$

2 demander  $p$  t.q.  $(a, b, c^0, d^0, e^0, f^0) \sim \frac{p}{1-p} (a^0, b^0, c^0, d^0, e^0, f^0)$

$$u_1(a, b) + u_2(c^0, e^0) + u_3(b, c^0, d^0) + u_4(b, d^0, f^0) + u_5(b, g^0) = p$$

## Élicitation : construction sur toutes les cliques



$$u_1(A, B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

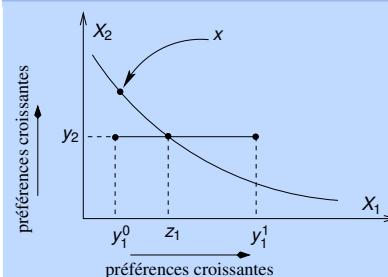
Élicitation des sous-utilités des cliques externes vers les cliques internes

- 1 Élicitation de la sous-utilité  $u_1(A, B)$
- 2 Élicitation de la sous-utilité  $u_2(C, E)$
- 3 Élicitation de la sous-utilité  $u_5(B, G)$
- 4 Élicitation de la sous-utilité  $u_4(B, D, F)$
- 5 Élicitation de la sous-utilité  $u_3(B, C, D)$

## Élicitation d'une utilité additive dans le certain (1/3)

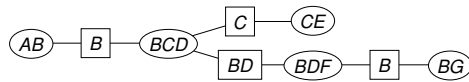
- unicité à une transformation affine : aucune raison rationnelle
- Contraintes sur l'espace des préférences :

Solvabilité restreinte



$$(y_1^0, y_2) \succsim_x x \succsim_x (y_1^1, y_2) \\ \implies \exists z_1 \text{ t.q. } (z_1, y_2) \sim_x x$$

## Rappel : unicité des utilités GAI



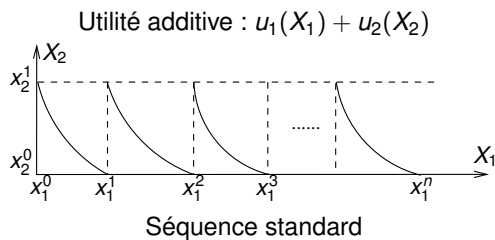
$$u_1(A, B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

### Proposition

- $(a^0, b^0, c^0, d^0, e^0, f^0, g^0)$  un n-uplet choisi arbitrairement
- solvabilité + essentialité + connexité des séparateurs  $\implies$  l'utilité GAI-décomposable  $u(\cdot)$  telle que :
  - $u_1(a^0, b) = 0 \forall b$
  - $u_2(c, e^0) = 0 \forall c$
  - $u_5(b, g^0) = 0 \forall b$
  - $u_4(b, d, f^0) = 0 \forall b, d$
  - $u_3(b^0, c^0, d^0) = 0$

est unique à une transformation linéaire strictement positive près.

## Élicitation d'une utilité additive dans le certain (2/3)

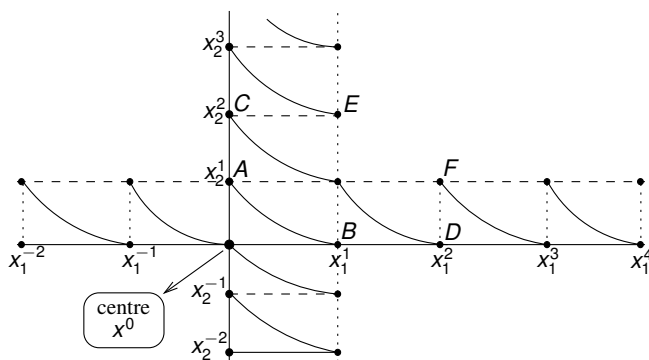


- $u_1(x_1^0) = 0 \quad u_2(x_2^0) = 0 \quad u_2(x_2^1) = 1$
- $u_1(x_1^1) = 1$
- $u_1(x_1^2) - u_1(x_1^1) = u_1(x_1^1) - u_1(x_1^0)$

Séquences standards  $\implies$  élicitation sur une grille

## Élicitation d'une utilité additive dans le certain (3/3)

Séquences standards  $\implies$  élicitation sur une grille



## Conclusion sur l'élicitation

### Procédure d'élicitation générale

- questionnaire dérivé de l'arbre GAI
- fonctionne aussi avec les graphes cycliques (après preprocessing : triangulation)

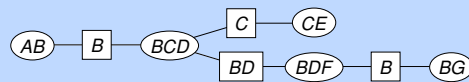
### Avantages principaux

- la procédure s'applique à **toute** décomposition GAI
- questions sont aussi simples que possible
- nombre réduit de questions

## Prise de décision avec les réseaux GAI

## optimisation sur l'ensemble des résultats

### Élément maximal d'une utilité GAI-décomposable



### Expression de l'optimum

$\max_{A,B,C,D,E,F,G}$

$$u_1(A, B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

$\max_A \max_B \max_C \max_D \max_E \max_F \max_G$

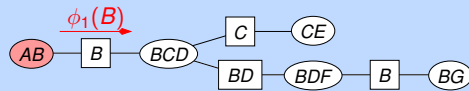
$$u_1(A, B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

## Phase de collecte (1/5)

### Élimination de A

$$\max_B \max_C \max_D \max_E \max_F \max_G \max_A u_1(A, B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

$$= \max_B \max_C \max_D \max_E \max_F \max_G \phi_1(B) + u_2(C, E) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

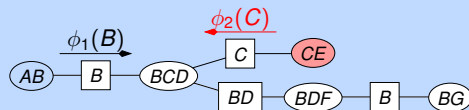


## Phase de collecte (2/5)

### Élimination de E

$$\max_B \max_C \max_D \max_F \max_G \max_E u_2(C, E) + \phi_1(B) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

$$= \max_B \max_C \max_D \max_F \max_G \phi_1(B) + \phi_2(C) + u_3(B, C, D) + u_4(B, D, F) + u_5(B, G)$$

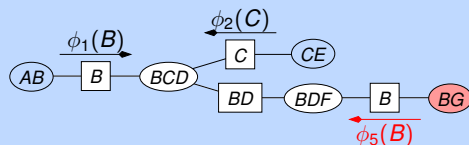


## Phase de collecte (3/5)

### Élimination de G

$$\max_B \max_C \max_D \max_F \max_G u_5(B, G) + \phi_1(B) + \phi_2(C) + u_3(B, C, D) + u_4(B, D, F)$$

$$= \max_B \max_C \max_D \max_F \max_G \phi_1(B) + \phi_2(C) + u_3(B, C, D) + u_4(B, D, F) + \phi_5(B)$$



## Phase de collecte (4/5)

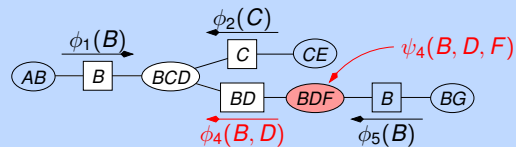
### Élimination de F

$$\max_B \max_C \max_D \max_F \psi_4(B, D, F)$$

$$+ \phi_1(B) + \phi_2(C) + u_3(B, C, D)$$

$$\text{où } \psi_4(B, D, F) = u_4(B, D, F) + \phi_5(B)$$

$$= \max_B \max_C \max_D \phi_1(B) + \phi_2(C) + u_3(B, C, D) + \phi_4(B, D)$$



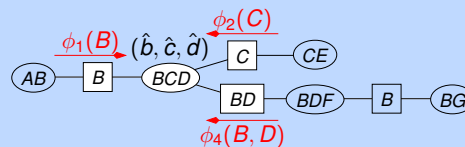
## Phase de collecte (5/5)

### Élimination de B, C, D

$$\max_{B, C, D} \psi_3(B, C, D)$$

$$\text{où } \psi_3(B, C, D) = \phi_1(B) + u_3(B, C, D) + \phi_4(B, D) + \phi_2(C)$$

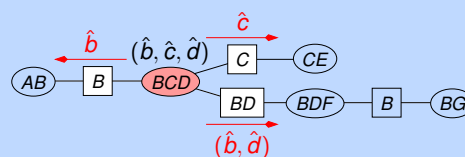
⇒ élément préféré :  $(\hat{b}, \hat{c}, \hat{d})$



## Phase de distribution (1/4)

### Distribution par la clique BCD

La clique  $BCD$  envoie des messages à ses séparateurs adjacents contenant la valeur de leurs attributs à l'optimum





## Phase de distribution (2/4)

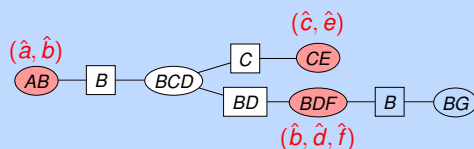
### Propagation par les séparateurs

Clique  $AB \implies \hat{a} = \text{Argmax}_A u_1(A, \hat{b})$

Clique  $CE \implies \hat{e} = \text{Argmax}_E u_2(\hat{c}, E)$

Clique  $BDF \implies \hat{f} = \text{Argmax}_F \psi_5(\hat{b}, \hat{d}, F)$

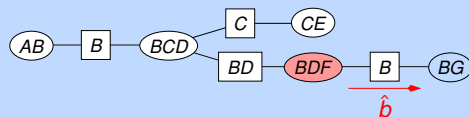
⚠ Les Argmax sont calculés pendant la phase de collecte ( $\phi$ )



## Phase de distribution (3/4)

### Distribution par la clique BDF

La clique  $BDF$  envoie un message au séparateur  $B$  contenant la valeur à l'optimum de l'attribut  $B$

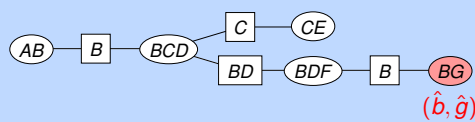


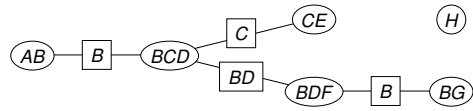
## Phase de distribution (4/4)

### Distribution par la clique BDF

Clique  $BG \implies \hat{g} = \text{Argmax}_G u_5(\hat{b}, G)$

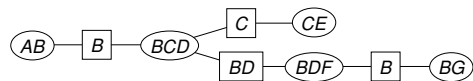
$\implies$  un élément optimal (préfér )  $(\hat{a}, \hat{b}, \hat{c}, \hat{d}, \hat{e}, \hat{f}, \hat{g})$





- optimisation avec une forêt GAI
- optimisation sous contraintes de certaines valeurs d'attributs
- optimisation sous contraintes exprimées sous forme de valeurs possibles de cliques
- optimisation sous contraintes générales  
 $\Rightarrow$  exprimer les contraintes comme de nouvelles cliques

## Rangement (1/4)



n-uplet optimal :  $(\hat{a}, \hat{b}, \hat{c}, \hat{d}, \hat{e}, \hat{f}, \hat{g})$

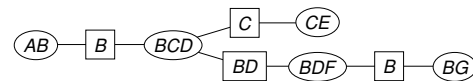
Quel est le deuxième élément préféré ?

un élément qui diffère du premier par au moins un attribut  
 quel attribut ?

on ne sait pas mais on sait qu'il appartient à une clique

$\Rightarrow$  au moins une des cliques devrait avoir une valeur différente

## Rangement (2/4)



n-uplet optimal :  $(\hat{a}, \hat{b}, \hat{c}, \hat{d}, \hat{e}, \hat{f}, \hat{g})$

le prochain n-uplet est dans l'un de ces ensembles :

Set 1 :  $(B, C, D) \neq (\hat{b}, \hat{c}, \hat{d})$

Set 2 :  $(B, C, D) = (\hat{b}, \hat{c}, \hat{d})$  et  $(B, D, F) \neq (\hat{b}, \hat{d}, \hat{f})$

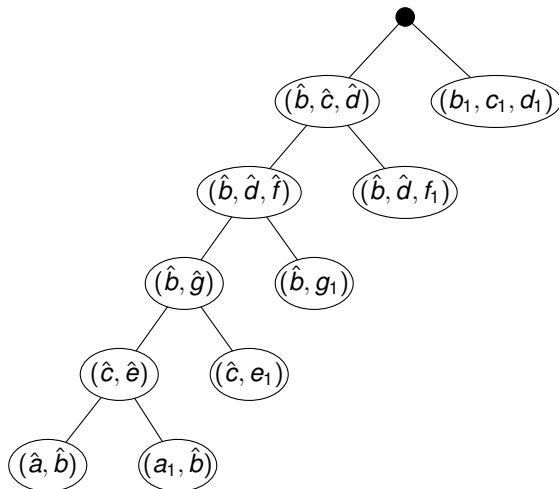
Set 3 :  $(B, C, D, F) = (\hat{b}, \hat{c}, \hat{d}, \hat{f})$  et  $(B, G) \neq (\hat{b}, \hat{g})$

Set 4 :  $(B, C, D, F, G) = (\hat{b}, \hat{c}, \hat{d}, \hat{f}, \hat{g})$  et  $(C, E) \neq (\hat{c}, \hat{e})$

Set 5 :  $(B, C, D, E, F, G) = (\hat{b}, \hat{c}, \hat{d}, \hat{e}, \hat{f}, \hat{g})$  et  $(A, B) \neq (\hat{a}, \hat{b})$

*Algorithmme* : utiliser la procédure d'optimisation sur chaque ensemble et choisir le n-uplet préféré

## Rangement (3/4)



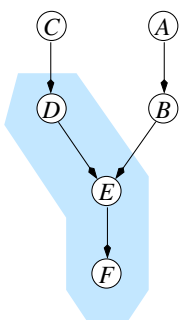
## Rangement (4/4)

Temps (en ms)	sans contraintes			avec contraintes		
	meilleur	Top 50	Top 100	meilleur	Top 50	Top 100
$K_1$	7	18	29	40	55	67
$K_2$	1	3	6	121	132	142
$K_3$	1	3	4	2067	2070	2073
$K_4$	40	45	50	4399	4410	4419
$K_5$	663	664	665	12662	12663	12665

## Conclusion sur les utilités GAI

- GAI = représentation compacte des préférences
- permet de diriger des procédures d'élicitation efficaces
  - questionnaires « simples » cognitivement
  - nombre réduit de questions
- permet de répondre efficacement à des requêtes d'optimisation
- application de Shafer-Shenoy aux utilités

UCP-net = Utility Conditional Preferential network  
= décomposition GAI + CP-net



1/ Décomposition GAI :

$$u(a, b, c, d, e, f) = u_1(a) + u_2(a, b) + u_3(c) + u_4(c, d) + u_5(b, d, e) + u_6(e, f)$$

2/ CP-net : Les attributs en "haut" du graphe

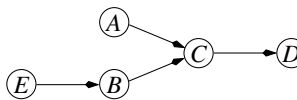
sont "indépendants" de ceux du bas :

$$(a, b, c, d, e, f) \succsim_X (a', b', c', d, e, f)$$

$$\iff (a, b, c, d', e', f') \succsim_X (a', b', c', d', e', f')$$

$\implies$  le graphe doit être un DAG (pas de circuit)

Élicitation de fonctions d'utilité par UCP-net (1/2)



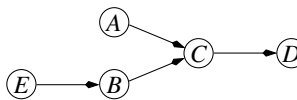
$$f_1(A) + f_2(E) + f_3(B|E) + f_4(C|A, B) + f_5(D|C)$$

$$f_1(A) : a \sim \begin{matrix} p & a'' \\ 1-p & a' \end{matrix} \quad f_1(a) = p$$

$$f_2(B|E) : (b, e) \sim \begin{matrix} p' & (b'', e) \\ 1-p' & (b', e) \end{matrix} \quad f_2(b|e) = p'$$

$$f_3(C|A, B) : (c, a, b) \sim \begin{matrix} p'' & (c'', a, b) \\ 1-p'' & (c', a, b) \end{matrix} \quad f_3(c|a, b) = p''$$

Élicitation de fonctions d'utilité par UCP-net (2/2)



on veut :  $f_1(A) + f_2(E) + f_3(B|E) + f_4(C|A, B) + f_5(D|C)$

les  $f_i$  ont été élicités séparément  $\implies$  il faut les agréger

$$f_3(c|a, b) \in [0, 1] \implies \pi_3(a, b)f_3(c|a, b) + \sigma_3(a, b)$$

$\implies$  pour chaque  $f_i$ , calculer  $\pi_i$  et  $\sigma_i$

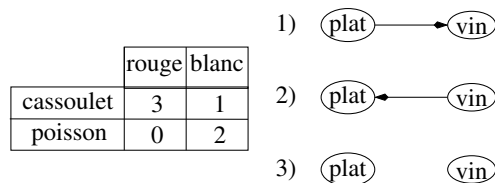
$\implies$  questions + résolution d'un programme linéaire

## Inconvénient culinaire des UCP-nets

- Repas = Plat de résistance + vin + dessert  
plat  $\in$  {cassoulet, poisson}  
vin  $\in$  {vin rouge, vin blanc}  
dessert  $\in$  {rien, profiterole au chocolat}

- $u(\text{repas}) = u_1(\text{plat}, \text{vin}) + u_2(\text{plat}, \text{dessert})$

(cassoulet,rouge)  $\succ_X$  (poisson,blanc)  $\succ_X$  (cassoulet,blanc)  $\succ_X$  (poisson,rouge)



$\Rightarrow$  non représentable par un UCP-net

## Biblio sur les modèles graphiques de « préférences »

### Références

- F Bacchus et A Grove (1995). "Graphical models for preference and utility", Proceedings d'UAI-95
- D Brazianus, C Boutilier (2005) "local utility elicitation in GAI models", Proceedings d'UAI-05
- C. Gonzales, P. Perny (2004) "GAI Networks for Utility Elicitation". Proceedings de KR-04.
- C. Gonzales, P. Perny (2005) "GAI Networks for Decision Making under Certainty", Proceedings d'IJCAI-05 – workshop on advances in preference handling.
- P Fishburn (1970) "Utility Theory for Decision Making", Wiley, NewYork
- C. Gonzales, P. Perny (2006) "GAI networks : from axiomatization to applications", EMPG-06.