GÉNÉRATION AUTOMATIQUE DE RÈGLES INDÉPENDANTES À PARTIR D'UNE BASE D'EXEMPLES

Jacques Lorigny

La quantité d'information de Shannon : le théorème fondamental de l'information

Bref historique

La théorie classique de l'information est due à l'ingénieur et mathématicien américain Claude Shannon, dont la publication mémorable de 1948 portait sur le théorie mathématique de la transmission par lignes [22]. Son théorème fondamental ouvrait la voie à de nombreux développements théoriques et pratiques, tant dans le domaine des techniques de communication que dans les processus de décision, de détermination, de reconnaissance des formes. Toutefois, rappelons qu'avant Shannon, et dès 1928, l'ingénieur en transmissions américain R. V. Hartley [7] avait proposé de mesurer le degré d'incertitude d'une épreuve à k issues par la quantité $\log k$; cette formulation n'était pas très éloignée de l'entropie de Shannon. Elle est même asymptotiquement identique dans le cas d'une répétition infinie de la même épreuve (cf. Yaglom et Yaglom, [25]).

Le théorème central de Shannon fut ensuite formulé en termes mathématiques rigoureux, et étendu à des processus de transmission généraux :

- en 1953, B. Mac Millan [19] précisa les notions de source d'information, de canal de transmission, et généralisa le théorème fondamental aux sources non markoviennes ;

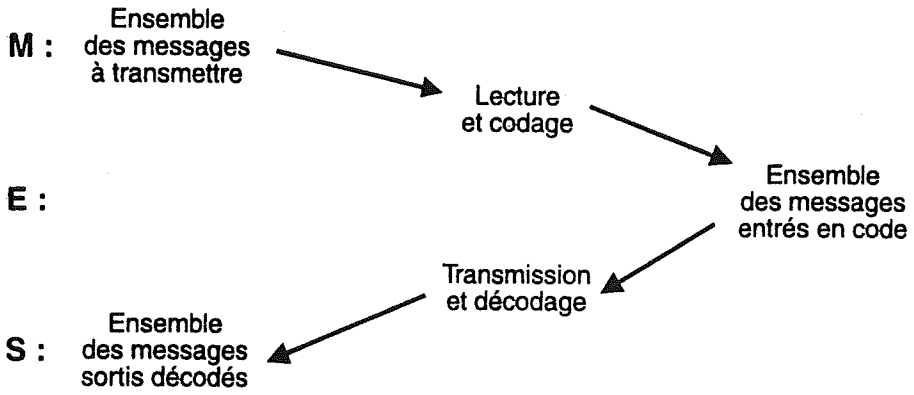
- en 1954, A. Feinstein [5] donna la première démonstration mathématique complète du théorème fondamental dans le cas des canaux sans mémoire, c'est-à-dire dont le

message de sortie S(t) ne dépend que du message d'entrée instantané E(t), et non des messages antérieurs E(t-1), E(t-2), etc ;

- en 1956, A.I. Khinchine [9] compléta la théorie mathématique pour les sources stationnaires, et les canaux stationnaires, sans anticipation et à mémoire finie[1] achevant ainsi le monument commencé dix ans plus tôt par Shannon.

Exposé sommaire du théorème

Un processus de transmission d'information peut se résumer selon le schéma suivant :



La source d'information

Isolons d'abord le problème de la source, en supposant le canal sans brouillage. Le décodage se fait sans erreur, et la seule question qui se pose est celle de l'économie de coût de transmission que l'on peut réaliser en choisissant un bon codage.

(1) *Source stationnaire* : dont le processus aléatoire de génération des messages d'entrée est invariant dans le temps ;
Canal stationnaire : dont le processus aléatoire de sortie des messages conditionnés par les messages entrés est invariant dans le temps ;
Canal sans anticipation : dont le message de sortie S(t) peut dépendre des messages d'entrée E(t), E(t-1), etc., mais pas des messages d'entrée postérieurs E(t+1), E(t+2), etc.
Canal de mémoire finie : dont le message de sortie S(t) peut dépendre des messages d'entrée E(t), E(t-1)... , E(t-m+1), mais pas des messages d'entrée précédents : E(t-m), E(t-m-1), etc.

Quel est le nombre minimum de signes codés nécessaire pour transmettre l'ensemble M des messages de la source?

Considérons un exemple simple : celui du codage en signes binaires (bits) d'un ensemble de messages qui sont des chiffres décimaux.

On observe d'abord qu'il faut 4 bits pour coder un chiffre décimal, puisque l'on a : $2^3 < 10 < 2^4$. Ensuite, on constate que, si l'on peut grouper l'ensemble M à transmettre par groupes de 2 chiffres décimaux, on a intérêt à coder directement les groupes : on a en effet : $2^6 < 100 < 2^7$. Il faut donc 7 bits par groupe, donc 3,5 bits par chiffre décimal au lieu de 4 précédemment. On augmente encore la longueur des tranches codées directement, en passant à des groupes de 3 chiffres décimaux. On a $2^9 < 1000 < 2^{10}$, d'où un coût, encore plus faible, de 10/3 = 3,33 bits par chiffre décimal. Etc.

Ainsi, on réalise une économie de transmission en codant l'ensemble des messages par blocs. Toutefois, il existe une limite à ce gain. On a : $2^{n-1} < 10^m < 2^n$, d'où l'on déduit : $n/m > \log 10/\log 2 = \log_2 10 = 3{,}3219\ldots$

D'où un **premier résultat** : pour transmettre un ensemble de messages pris dans un alphabet de base k (ici, k=10), à l'aide d'un code de base B (ici, B=2), il faut utiliser un nombre minimum *a priori* de $\log_B k$ signes par message, et l'on peut se rapprocher de ce minimum à condition de **grouper les messages** en séquences les plus longues possibles.

On s'aperçoit ensuite que le seuil peut encore être abaissé, en tenant compte de la **disparité des fréquences** d'apparition des messages dans la source réelle. Ainsi, dans un texte français moyen, la lettre E arrive avec une probabilité de 0,175 alors que la lettre W arrive avec la probabilité 0,0002. Le principe de l'économie va consister à adopter un **codage de longueur variable** : plus court pour les messages fréquents, plus long pour les messages rares. Pour mesurer le gain possible avec cette optimisation, on introduit une fonction d'entropie, définie par :

$$H_B = \sum_{i=i}^{k} p_i \log_B 1/p_i$$

dans laquelle k est la base de la source M, et $\{p_i\}$ la distribution de probabilité des messages de M. Cette fonction sera étudiée plus loin, et nous verrons que l'on a :

$$0 \leq H_B \leq \log_B k$$

H_B est minimum et égal à 0 si toutes les probabilités p_i sont nulles, sauf une, égale à 1.

H_B est maximum et égal à $\log_B k$ si toutes les probabilités sont égales à 1/k (équiprobabilité).

Par exemple, pour une lettre d'un texte français moyen, on trouve $H_{10} = 1{,}199$ alors que le seuil précédent, dû au simple groupage brut, est $\log_{10} 26 = 1{,}415$. La différence mesure le gain supplémentaire réalisé en tenant compte de la disparité des fréquences des lettres en français.

D'où un **second résultat** : on peut abaisser le seuil de codage jusqu'à $H_B(\{p_i\})$ signaux en moyenne par message, ce qui est inférieur à la première valeur obtenue plus haut, $\log_B k$.

Enfin, dernière étape, on peut encore diminuer la valeur du seuil de codage de la source, si les messages successifs sont **stochastiquement dépendants**. En effet, l'effet combiné du groupage en blocs et de la prise en compte des fréquences, révèle que la dépendance abaisse encore plus l'entropie des messages groupés que dans le cas où ils sont indépendants en probabilité.

Considérons deux distributions de probabilité :

$(\alpha) = [\, p\,(A_i)\,]$, $\quad i=1$ à I

$(\beta) = [\, p\,(B_j)]$, $\quad j=1$ à J, et la distribution conjointe :

$(\alpha\beta) = [\, p\,(A_i \text{ et } B_j)]$, $\quad i=1$ à I, et $j=1$ à J

On note :

$$H(\alpha) = \sum_{i=1}^{I} p\,(A_i) \,.\, \log 1 / p\,(A_i)$$

$$H(\beta) = \sum_{j=1}^{J} p\,(B_j) \,.\, \log 1 / p\,(B_j)$$

$$H(\alpha\beta) = \sum_{i=1}^{I} \sum_{j=1}^{J} p\,(A_i \text{ et } B_j) \,.\, \log 1 / p\,(A_i \text{ et } B_j)$$

On définit aussi une nouvelle quantité, l'entropie conditionnelle de β sachant α :

$$H(\beta/\alpha) = \sum_{i=1}^{I} p(A_i) \cdot \sum_{j=1}^{J} p(B_j/A_i) \cdot \log 1/p(B_j/A_i)$$

On montre à partir de la concavité de la fonction logarithme (cf. par exemple Yaglom et Yaglom [22], p.163), que l'on a :

$$0 \leq H(\beta/\alpha) \leq H(\beta)$$

- L'entropie conditionnelle $H(\beta/\alpha)$ est **minimum** et égale à 0 si et seulement si les épreuves α et β sont en dépendance stricte. En d'autres termes, la réalisation de l'expérience α ne laisse plus aucune indétermination sur l'issue de β.

- L'entropie conditionnelle est **maximum** et égale à $H(\beta)$ si et seulement si les épreuves α et β sont stochastiquement indépendantes. La réalisation de α laisse entière l'indétermination de β.

Par ailleurs, on vérifie aisément que l'on a :

$$H(\alpha\beta) = H(\alpha) + H(\beta/\alpha),$$

$$\text{d'où } H(\alpha) \leq H(\alpha\beta) \leq H(\alpha) + H(\beta),$$

l'égalité de gauche a lieu dans le cas de la dépendance stricte, celle de droite dans le cas de l'indépendance stochastique.

Notre troisième amélioration du seuil de codage provient de l'inégalité de droite : lorsque les messages sont mutuellement dépendants, $H(\alpha\beta)$ est strictement inférieur à $H(\alpha) + H(\beta)$, d'où l'économie possible. Voici quelques évaluations obtenues par Shannon dans la langue anglaise :

pour un couple de lettres, $1/2\ H_{10}(\alpha\beta) = 1{,}075$

pour un triplet de lettres, $1/3\ H_{10}(\alpha\beta\gamma) = 0{,}993$

On peut comparer à : $H_{10}(\alpha) = 1{,}242$

et à : $\log_{10} 26 = 1{,}415$ (codage fixe).

Shannon a estimé la valeur relative à un groupage en tranches de 8 lettres :

$1/8\ H_{10}(\alpha_1\alpha_2 \ldots \alpha_8)\ \#\ 0{,}6$

On vérifie que le seuil de codage, à savoir :

$1/\nu\ (\alpha_1\,\alpha_2\ \ldots\ \alpha_\nu)$, diminue lorsque ν croît.

Intéressons-nous à sa limite quand ν croît indéfiniment

$$H_{lim} = \lim_{\nu\infty}\ 1/\nu\ H(\alpha_1\alpha_2 \ldots \alpha_\nu)$$

On démontre (cf. par exemple J.C. Simon [23]) que, pour une source stationnaire, cette quantité est aussi égale à l'entropie conditionnelle d'un message connaissant la séquence infiniment longue des messages précédents, c'est-à-dire avec nos notations, $H_{lim} = lim\ H(\alpha_\nu/\alpha_1\alpha_2 \ldots \alpha_{\nu-1})/$. On montre qu'on ne peut pas descendre au-dessous de ce seuil de codage.

À titre indicatif, on estime que la quantité H_{lim} est de l'ordre de 0,5 pour les langues européennes, toujours exprimée en chiffres décimaux par lettre. On appelle excédent d'une langue la quantité $1 - H_{lim} / \log k$, écart entre l'information maximum qu'une lettre pourrait contenir et celle qu'elle apporte réellement (en moyenne).

On peut maintenant énoncer le théorème fondamental (pour la partie codage, c'est-à-dire en supposant la transmission sans brouillage) :

Une source de messages peut être codée à l'aide d'un nombre moyen de signes par message aussi voisin que l'on veut de la quantité Hlim / log B - mais en aucun cas inférieur - , en codant par blocs suffisamment longs, et en modulant la longueur des codes selon la fréquence.

Le canal de transmission

La source d'information est maintenant supposée codée. L'ensemble E des messages codés entre dans le canal de transmission. Les messages sont décodés à la sortie du canal et l'on obtient l'ensemble S des messages reçus. En général, et pour des raisons diverses (déformations physiques, addition de perturbations extérieures) il se produit un brouillage à la transmission, c'est-à-dire que la correspondance entre E et S n'est plus une application : un message entré sous forme codée A_i se trouve reçu et décodé selon plusieurs messages possibles B_j. On note $p(B_j/A_i)$ les probabilités conditionnel-

les des messages sortis (décodés) connaissant les messages entrés (codés). Ces quantités forment une **matrice carrée qui caractérise entièrement le canal de transmission.**

On définit maintenant **l'information mutuelle** entre l'entrée E et la sortie S :

$$I(E,S) = H(S) - H(S/E) \qquad (2)$$

où H(S) est l'entropie des messages de sortie,

H(S/E) est l'entropie conditionnelle des messages sortis connaissant les messages entrés.

Compte tenu des inégalités vues plus haut, on a :

$$O \leq I(E, S) \leq (S) \qquad (3)$$

- l'information est minimum et égale à 0 quand les distributions de E et de S sont strictement indépendantes en probabilité (brouillage total),

- l'information est maximum et égale à H(S) quand la dépendance est stricte entre E et S. À tout message d'entrée ne correspond qu'un message de sortie.

En remarquant d'autre part que l'on a :

$H(S/E) = H(E,S) - H(E)$, on en déduit que :

$I(E,S) = H(E) + H(S) - H(E,S)$

l'information mutuelle est une fonction symétrique des entrées et des sorties.

d'où $I(E,S) = I(S,E) = H(E) - H(E/S)$,
et, par conséquent,
$I(E,S) \leq H(E)$ (4)

L'égalité a lieu quand H(E/S) est nulle, c'est-à-dire lorsqu'à tout message de sortie correspond un seul message d'entrée. Le canal de transmission est alors sans aucun brouillage.

On peut maintenant définir la **capacité d'un canal de transmission** par la formule :

$$C = \underset{p(E)}{max}\, I(E,S) \quad \text{(exprimée en bits par message)}$$

dans laquelle p(E) désigne une distribution de probabilités sur E, et le maximum selon p(E) est pris sur toutes les distributions possibles, de sorte que la capacité C ne dépend pas de la source E mais uniquement du canal lui-même, c'est-à-dire de la matrice des probabilités des messages de sortie conditionnés par les messages d'entrée.

Le théorème fondamental (avec brouillage) peut maintenant s'énoncer ainsi :

Si on dispose d'une source E d'entropie H(E) et d'un canal de transmission de capacité C et qu'on a l'inégalité $H(E) \leq C$ *, alors on peut trouver un code approprié tel que la probabilité d'erreur à la réception de chaque message transmis soit inférieure à* ε *, si petit soit-il.*

Par contre, si on a :

$H(E) > C$ *, la probabilité d'avoir une erreur à la réception sera toujours supérieure à une certaine valeur finie, quel que soit le code retenu.*

Notons que, dans le cas le plus général, H(E) représente ce qui a été noté plus haut H_{lim}. De même, la transmission et le décodage se font par blocs longs de messages. Enfin, R.M. Fano [4] a montré que la distribution p(E) qui maximise la quantité I(E,S), et par conséquent détermine la capacité du canal, est celle qui répartit l'information mutuelle uniformément entre les messages d'entrée. Ce principe sert de guide à l'obtention des codes optimaux.

Remarque : cet exposé s'est voulu intuitif mais sans recherche de rigueur mathématique. On trouvera les exposés rigoureux dans Guiasu et Theodorescu [6]. Il existe aussi des démonstrations simplifiées mais très instructives dans Yaglom et Yaglom [25]) ou dans J.C. Simon [23].

Entropie et quantité d'information sur les structures d'information arborescentes (codes et questionnaires arborescents)

Comment construire un code se rapprochant le plus possible de l'optimum théorique révélé par la théorie mathématique ? Shannon, et Fano [4], ont proposé une méthode consistant à équirépartir le plus possible les issues de chaque niveau de codage, en partant du sommet-racine. C'est donc un algorithme descendant et qui convient très bien la plupart du temps. Cependant, un algorithme meilleur a été donné par Huffman : il construit le code de façon ascendante en regroupant les issues de moindre fréquence. Contrairement à celui de Shannon-Fano, l'algorithme de Huffman [8] est rigoureusement optimal, c'est-à-dire qu'il aboutit à un nombre moyen de symboles par message le plus faible possible, mais pas forcément égal à la borne inférieure théorique.

À partir de la représentation en arborescence du code, on définit des quantités qui joueront un rôle dans les structures plus complexes dans lesquelles nous allons entrer plus loin.

Formule générale de coût moyen sur une arborescence

Considérons une arborescence probabilisée A = (Y, Δ, P), où

Y est l'ensemble des sommets : on note y_0 le sommet-racine

Y_t l'ensemble des sommets terminaux : on a : $Y_t \subset Y$

Δ l'ensemble des arcs : on note $\Delta^+(y)$ l'ensemble des sommets successeurs immédiats de y

P une distribution de probabilité conservative en chaque sommet ; on a :

$$\begin{cases} p(y_0) = 1 \\ \forall\, y \in Y - Y_t \quad : \quad \sum_{y_i \in \Delta^+(y)} p(y_i) = p(y) \end{cases}$$

On introduit sur les sommets terminaux y_t de Y_t, une fonction générale de coût w - censée représenter une longueur, une énergie, un temps, ... - , qui est la somme, le long du chemin (unique) allant de y_0 à y_t, d'une fonction *g* des sommets traversés *y* et d'une fonction *k* des arcs empruntés δ. On s'intéresse au calcul du coût moyen $\bar{w}$ sur l'ensemble Y_t, et l'on note μ_t le chemin $[y_0..y_t]$.

$$\forall\, y_t \in Y_t : w(y_t) = \sum_{y \in \mu_t} g(y) + \sum_{\delta \in \mu_t} k(\delta)$$

$$\bar{w} = \sum_{Y_t} p(y_t) \cdot w(y_t)$$

Posons $\forall\, y \in Y$ et $\forall\, y_t \in Y_t$ $\quad a(y, y_t) \begin{cases} = 1 & \text{si } y \in \mu_t \\ = 0 & \text{sinon} \end{cases}$

$\forall\, \delta \in \Delta$ et $\forall\, y_t \in Y_t$ $\quad b(\delta, y_t) \begin{cases} = 1 & \text{si } \delta \in \mu_t \\ = 0 & \text{sinon} \end{cases}$

Il vient :

$$\overline{w} = \sum_{Y_t} p(y_t) \cdot \sum_{Y} g(y) \cdot a(y, y_t) + \sum_{Y_t} p(y_t) \cdot \sum_{\Delta} k(\delta) \cdot b(\delta, y_t)$$

$$\overline{w} = \sum_{Y} g(y) \cdot \sum_{Y_t} p(y_t) \cdot a(y, y_t) + \sum_{\Delta} k(\delta) \cdot \sum_{Y_t} p(y_t) \cdot b(\delta, y_t)$$

D'où : $\overline{w} = \sum_{Y} g(y) \cdot p(y) + \sum_{\Delta} k(\delta) \cdot p(\delta)$ (5)

Longueur de cheminement sur une arborescence

On appelle **longueur de cheminement** sur une arborescence le nombre moyen d'arcs des chemins (uniques) allant du sommet-racine aux sommets terminaux y_t. Notons que c'est aussi le nombre moyen de signes par message dans la théorie du codage. Pour calculer : $L = \sum_{Y_t} p(y_t) \cdot l(y_t)$,

On applique la formule (5) avec : $g(y) \begin{cases} = 1 & si\ y \in Y - Y_t \\ = 0 & si\ y \in Y_t \end{cases}$

et $k(\delta) = 0 \quad \forall \quad \delta$

Il vient aussitôt : $L = \sum_{Y - Y_t} p(y)$

Quantités d'information sur une arborescence

Rappelons d'abord les définitions données par Cl-F. Picard, fondateur de la Théorie des questionnaires en 1962. Ces mesures ont été étudiées par lui-même dans le cas des questionnaires arborescents ou latticiels (cf. [21]), puis ensuite généralisées (cf. [11, 12]) à des graphes plus généraux, les "graphes ouverts" que nous allons considérer dans toute la suite. Dans ce paragraphe, nous reprenons sa terminologie, mais en nous limitant aux questionnaires arborescents.

L'information transmise

Étant donné un questionnaire arborescent A = (Y,Δ, P), Picard appelle "information transmise" par A la quantité :

$$I(A) = \sum_{Y_t} p(y_t) \cdot \log 1 / p(y_t)$$

L'information traitée par un sommet

Pour tout sommet non terminal $y \in Y - Y_t$, il pose :

$$J(y) = \sum_{y_i \in \Delta^+(y)} \frac{p(y\,i)}{p(y)} \cdot \log \frac{p(y)}{p(y\,i)}$$

L'information traitée par l'arborescence

L'information traitée par l'ensemble du questionnaire arborescent est alors définie par :

$$J(A) = \sum_{Y-Y_t} J(y) \cdot p(y)$$

On vérifie aisément que l'on a, du fait de la conservativité de la probabilité en chaque noeud, J(A) = I(A). Mais ce n'est vrai que pour les questionnaires arborescents. Dans le cas des questionnaires latticiels, on a seulement : $I(A) \leq J(A)$.

Retrouvons le lien entre la longueur de cheminement et l'information transmise dans le cas du codage le plus simple :

En remarquant que l'on a : $\forall y \in Y\text{-}Y_t : 0 \leq J(y) \leq \log B$
(B base du code)

on en déduit :

$$0 \leq \sum_{Y - Y_t} J(y) \cdot p(y) \leq \log B \cdot \sum_{Y - Y_t} p(y)$$

Soit : $L \geq I(A) / \log B$

et l'on retrouve la première partie du théorème du codage optimal, dans le cas d'un codage groupé par blocs mais ne tenant pas compte de la dépendance stochastique entre les messages d'un même bloc.

On démontre (cf. Picard [21], tome 2, p.89) que le seuil est effectivement atteint si et seulement si tout sommet terminal de niveau n - avec $n = 0$ pour le sommet-racine - a pour probabilité B^{-n}.

Entropie et quantité d'information sur les graphes ouverts

Définition des graphes ouverts

Nous allons définir des êtres mathématiques très proches des graphes classiques, mais un peu différents, de la même façon qu'en topologie borélienne, un intervalle ouvert est différent d'un intervalle fermé. Les arcs d'entrée dans le graphe ouvert seront démunis de sommet-amont et les arcs de sortie seront sans sommet-aval. En voici la raison : alors que les graphes classiques, conçus pour représenter des échanges matériels (par exemple, les réseaux de transport) sont composés d'abord de sommets, puis, ensuite, d'un certain nombre d'arcs reliant ces sommets, les graphes nouveaux que nécessite la cognitique (par exemple, les réseaux neuronaux) partent d'abord d'arcs et de chemins, et ne créent des sommets que pour supporter et séparer les chemins. Les sommets jouent en quelque sorte un rôle second.

Nota : lorsque le terme de "graphe" sera employé seul, il s'agira toujours de graphe au sens classique.

Considérons un graphe fini valué $G = (X, \Gamma, V)$ à valuation strictement positive $v > 0$.

Soient $v^-(x)$ *et* $v^+(x)$ les flux entrant et sortant d'un sommet x. On note :

$X_e = \{ x \in X : v^-(x) < v^+(x) \}$ l'ensemble des "sommets d'entrée"

$X_s = \{ x \in X : v^-(x) > v^+(x) \}$ l'ensemble des "sommets de sortie"

$X_a = \{ x \in X : v^-(x) = v^+(x) \neq 0 \}$ l'ensemble des "sommets d'articulation"

et l'on suppose $X_e \neq \varnothing$, d'où l'on déduit $X_s \neq \varnothing$.

G est complété par l'adjonction de :

- deux sommets notés ω_e et ω_s ,

- les arcs $\{ (\omega_e , x_e) : x_e \in X_e \}$,
  munis des valuations $v^*(\omega_e, x_e) = v^+(x_e) - v^-(x_e)$

- les arcs $\{ (x_s, \omega_s) : x_s \in X_s \}$,
  munis des valuations $v^*(x_s, \omega_s) = v^-(x_s) - v^+(x_s)$

soit G^* le graphe ainsi complété. La valuation v, complétée par v^* sur les nouveaux arcs, est conservative en tous les sommets, sauf ω_e qu'on appelle **sommet d'environnement-amont** et ω_s, **sommet d'environnement-aval**.

On désignera par C l'ensemble des chemins d'entrée-sortie de G:

$$C = \{ c = [x_e x_1 x_2 \dots x_s] : x_e \in X_e , x_s \in X_s \}$$

et par C^* l'ensemble des chemins d'entrée-sortie de G^* :

$$C^* = \{ c^* = [\omega_e x_e x_1 x_2 \dots x_s \omega_s] : x_e \in X_e , x_s \in X_s \}$$

C et C^* sont en correspondance biunivoque.

Hypothèse de connexité d'entrée-sortie : on suppose que tout sommet x de X se trouve sur au moins un chemin c de C. En d'autres termes, il n'existe pas de sommet isolé ou de circuit isolé dans G.

Autres notations :

v^*	Valuation externe totale : $v^* = v^*(\omega_e) = v^*(\omega_s)$)
$\Gamma^+(x)$	Ensemble des sommets z successeurs immédiats de x dans G, c'est-à-dire tels qu'il existe dans Γ un arc $\gamma = (x, z)$
$\Gamma^-(x)$	Ensemble des sommets z antécédents immédiats de x dans G, c'est-à-dire tels qu'il existe dans Γ un arc $\gamma = (z, x)$

Ce qu'on appelle le "graphe ouvert", c'est en toute rigueur l'"entité" - qui **n'est pas** un graphe classique -, composée du graphe G **et** des arcs de G^* **sans** les sommets d'environnement ω_e et ω_s (*cf. Fig.1*).

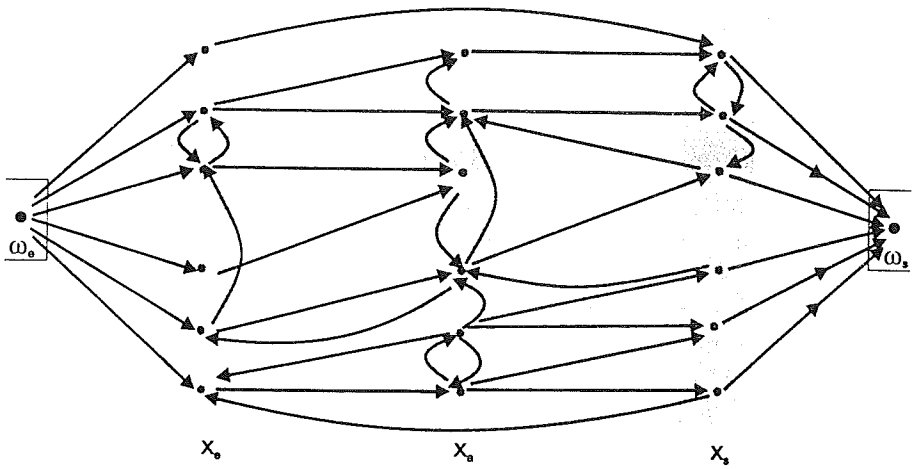


Fig. 1 Graphe ouvert général

Arborescence dépliée compatible

Nous appelons "arborescence dépliée" une arborescence qui déplie - déploie, développe - l'ensemble des chemins C^* du graphe en réseau G^*, un peu comme on démêle un écheveau qui s'est mis en boule. Cela signifie physiquement que l'on désire associer à un graphe de flux quantitatifs (typiquement, un réseau de flux routiers, un réseau d'échange matériel), un ensemble de cheminements individualisés (les itinéraires personnels des usagers, ou des lots échangés).

Considérons le graphe arborescent $A = (Y, \Delta, P)$ défini à partir du graphe G^*, de la façon suivante :

L'ensemble des sommets Y

- Au sommet ω_e de G* on associe le sommet-racine y_0 de A,

- À tout chemin partant de ω_e dans G^*, on associe un sommet y de Y (si G contient une boucle ou un circuit, Y est infini).

L'ensemble des arcs Δ

Notons $\mu = [\omega_e \ldots x]$, l'un de ces chemins de G^* conduisant à un sommet x de G, et y^j le sommet correspondant de A. Et soit:

$\{ [\omega_e \ldots x z_i] : z_i \in \Gamma^{*+}(x) \}$ l'ensemble des k chemins qui prolongent μ d'un arc dans G^*.

Alors, dans l'arborescence A, on relie le sommet y^j aux sommets $\{y^j_i : i=1 \text{ à } k\}$ qui correspondent à ces k chemins prolongés, par des arcs (y^j, y^j_i).

Remarque : l'ensemble C* correspond de façon biunivoque à l'ensemble, que nous noterons Y_t, des sommets terminaux de A.

La valuation de probabilité P

Nous allons définir sur A une famille de valuations qui seront dites **compatibles avec la valuation V**, soit P, positive et vérifiant les conditions :

- $p(y_0) = 1$, P est une distribution de probabilité

- **conservativité du flux** de P, soit :

$$\forall\, y \in Y - Y_t \quad \sum_{y_i \in \Delta^+(y)} p(y, y_i) = p(y)$$

- Compatibilité de P avec V, soit :

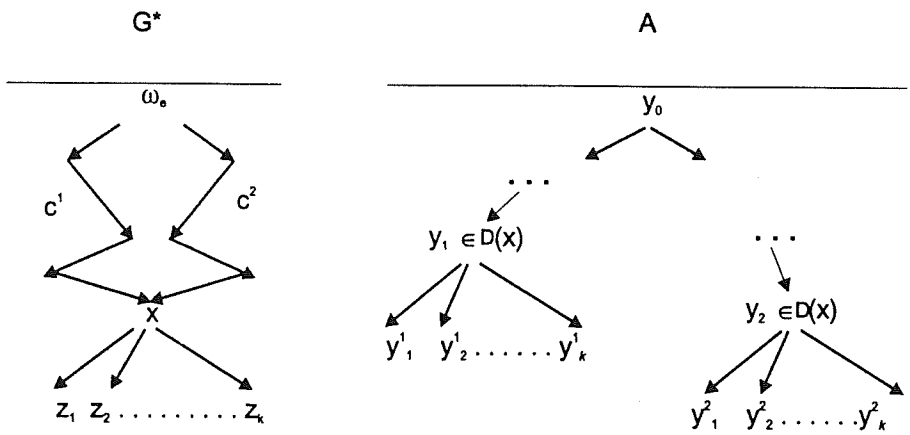
Pour tout x de X^*, on note $D(x)$ l'ensemble des sommets y^j qui lui correspondent dans l'arborescence dépliée A, ainsi que l'ensemble des chemins de G^* aboutissant à x : $[\omega_e \ldots x]$, dont ils sont les images bijectives.

On pose la **condition de compatibilité** :

$$\forall x \in X^* \quad \sum_{y^j \in D(x)} p(y^j) = v(x)/v^*, \qquad (6)$$

$$\forall x \in X^* \quad \forall z_i \in \Gamma^{*+}(x) \sum_{y^j \in D(x)} p(y^j, y^j{}_i) = v(x, z_i)/v^* \qquad (7)$$

où $y^j{}_i$ désigne le sommet de A correspondant au chemin [ω_e ...xz_i] de G*, qui prolonge d'un arc le chemin [ω_e ...x] ayant pour image dans A le sommet y^j.



Remarques :

- la seconde condition de compatibilité est suffisante. La première s'en déduit par suite de la conservativité du flux sur A ;

- la valuation d'un sommet terminal y_t de A, soit $p(y_t)$, est aussi la probabilité du chemin correspondant de C^*, en vertu de la bijection entre les ensembles Y_t de A et C^* sur G^*.

Cas particulier : probabilité dite "**proportionnelle aux flux**"

Soit $x \in X^*$ de G et un chemin μ = [$\omega_e\, x_1\, x_2$ .. $x_r x$] allant de ω_e jusqu'à x. Considérons la valuation sur A : p(y) telle que :

$$\begin{cases} p(y_0) = 1 \\ p(y) = \dfrac{v^*(\omega_e, x_1)}{v^*} \cdot \dfrac{v(x_1, x_2)}{v(x_1)} \cdot \ldots \cdot \dfrac{v(x_r, x)}{v(x_r)} \end{cases}$$

On montre que c'est bien une probabilité compatible (cf.[11]). Cette distribution est appelée à jouer un rôle particulier dans les mesures d'entropie et d'information qui vont être introduites.

Remarque : Quand on se donne un graphe G, il lui correspond en général sur l'arborescence dépliée A une infinité de probabilités P compatibles avec la valuation V de G. Inversement, si l'on part d'un graphe non valué G = (X, Γ) et d'une arborescence dépliée munie d'une probabilité quelconque P, on en déduit une valuation V sur G, par sommation des probabilités sur les arcs de A correspondant à un même arc de G* . C'est ce que nous appellerons un **repliement de l'arborescence A sur le graphe ouvert G**. Le repliement opère une agrégation, et donc occulte les différences entre les cheminements individuels. Mais il fournit une représentation finie et plus économique.

Formule générale de coût moyen sur les graphes ouverts.

Soit le graphe ouvert G = (X, Γ , V ; ω_e, ω_s) et son arborescence dépliée compatible A = (Y, Δ, P). On étudie un coût moyen de cheminement $\overline{w}$ analogue à celui des arborescences étudié plus haut - cf. formule (5) -, en tant que somme d'un coût de sommet $g(x)$ et d'un coût d'arc $k(xz)$.

Soit C^* l'ensemble des chemins c^* d'entrée-sortie de G :

$c^* = [\omega_e, x_e, x_2, \ldots, x_s, \omega_s]$

et l'on pose :

$$w(c^*) = \sum_{x \in c^*} g(x) + \sum_{(x,z) \in c^*} k(x,z)$$

On cherche $\overline{w} = \sum_{C^*} w(c^*) \cdot p(y_t)$, où y_t est l'image de c^* dans A.

Soient les fonctions d'incidence analogues à celles du cas arborescent, mais définies, cette fois-ci, relativement à chaque sommet x de X^*. On pose :

$$\forall y^j \in D(x), \forall y_t \in Y_t : a'_x(y^j, y_t) \begin{cases} = 1 & si \quad y^j \in [y_0 \ldots y_t] \\ = 0 & \text{sinon} \end{cases}$$

$\forall y^j \in D(x), \forall y^j_i \in \Delta^{*+}(y^j), \forall y_t \in Y_t$:

$$b'_x((y^j, y^j_i), y_t) \begin{cases} = 1 & \text{si } (y^j, y^j_i) \in [y_0 \ldots y_t] \\ = 0 & \text{sinon} \end{cases}$$

Le coût moyen s'écrit donc comme une somme de deux termes $\overline{w}_1$ et $\overline{w}_2$:

$$\overline{w}_1 = \sum_{Y_t} p(y_t) \cdot \sum_{X^*} \sum_{y^j \in D(x)} a'_x(y^j, y_t) \cdot g(x)$$

$$\overline{w}_2 = \sum_{Y_t} p(y_t) \cdot \sum_{X^*} \sum_{z_i \in \Gamma^{*+}(x)} \sum_{y^j \in D(x)} b'_x((y^j, y^j_i), y_t) \cdot k(x, z_i)$$

Commençons par calculer $\overline{w}_1$. En intervertissant les sommations, on obtient :

$$\overline{w}_1 = \sum_{X^*} g(x) \cdot \sum_{y^j \in D(x)} \sum_{Y_t} p(y_t) \cdot a'_x(y^j, y_t)$$

d'où : $\overline{w}_1 = \sum_{X^*} g(x) \cdot \sum_{y^j \in D(x)} p(y^j) = \sum_{X^*} g(x) \cdot v(x)/v^*$

(d'après (6))

Calculons le terme relatif aux coûts des arcs, $\overline{w}_2$:

$$\overline{w}_2 = \sum_{X^*} \sum_{z_i \in \Gamma^{*+}(x)} k(x, z_i) \cdot \sum_{y^j \in D(x)} \sum_{Y_t} p(y_t) \cdot b'_x((y^j, y^j_i), y_t)$$

$$\overline{w}_2 = \sum_{X^*} \sum_{z_i \in \Gamma^{*+}(x)} k(x, z_i) \cdot \sum_{y^j \in D(x)} p(y^j, y^j_i)$$

$$\overline{w}_2 = \sum_{X^*} \sum_{z_i \in \Gamma^{*+}(x)} k(x, z_i) \cdot v(x, z_i)/w^* \qquad \text{d'après (7)}$$

c'est-à-dire $\overline{w}_2 = \sum_{\Gamma^*} k(\gamma) \cdot v(\gamma)/v^*$

On obtient, en récapitulant :

$$\overline{w} = 1/v^* \cdot [\sum_{X^*} v(x) \cdot g(x) + \sum_{\Gamma^*} v(\gamma) \cdot k(\gamma)] \qquad (8)$$

Remarque : Le calcul suppose en toute rigueur que toutes les sommes manipulées sont finies, ce qui n'est pas le cas lorsque le graphe ouvert présente des boucles ou des circuits, et, donc, que l'arborescence dépliée est infinie. La démonstration est donnée en [10], dans l'hypothèse de probabilité proportionnelle aux flux et pour le coût "longueur de cheminement". Sa généralisation à des coûts quelconques est aisée :

- pour des coûts entiers, on se ramène au problème de longueur de cheminement en créant un graphe fictif à partir de G , par adjonction de sommets fictifs intermédiaires afin de remplacer chaque coût entier par un chapelet de coûts égaux à 1. Il suffit de vérifier que les hypothèses définissant un graphe ouvert, notamment l'hypothèse de connexité d'entrée-sortie sont bien vérifiées dans le nouveau graphe ;

- pour des coûts fractionnaires, ou des approximations fractionnaires de coûts quelconques, on dilate l'échelle des valuations jusqu'à se ramener à des coûts partout entiers.

Longueur de cheminement sur les graphes ouverts

On applique la formule obtenue ci-dessus avec :

$$g(x) \begin{cases} = 1 & \text{si } x \in X \text{ ou } x = \omega_e \\ = 0 & \text{si } x = \omega_s \end{cases}$$

et $k(\delta) = 0 \quad \forall \delta$

On obtient : $L = 1 + \sum_{X} v(x)/v^*$

Mesures d'information sur les graphes ouverts

On généralise les mesures de Picard aux graphes ouverts, en vue de réseaux cognitiques que nous définirons plus loin par repliement de questionnaires d'identification (quid) :

- **l'entropie systémique** de G est une extension de l'information traitée de Picard,
- **l'information aléatoire** de G est une extension de l'information transmise de Picard,
- **l'information acquise** de G est la différence entre l'entropie systémique et l'information aléatoire de G.

L'entropie systémique

Etant donné un graphe G=(X, Γ,V), complété en G^* comme plus haut, on appelle **entropie systémique du graphe ouvert G** la quantité :

$$HSYS\,(G) = v^* \,.\, H\,(\omega_e) + \sum_X v\,(x) \,.\, H_S\,(x)$$

avec :

$$H\,(\omega_e) = \sum_{X_e} \frac{v^*\,(\omega_e, x_e)}{v^*} \,.\, \log \frac{v^*}{v^*\,(\omega_e, x_e)}$$

(entropie de Shannon des flux externes entrant dans G)

$$\forall\, x \in X \quad H_s\,(x) = \sum_{z \in \Gamma^{*+}(x)} \frac{v^*\,(x, z)}{v\,(x)} \,.\, \log \frac{v\,(x)}{v^*\,(x, z)}$$

(entropie de Shannon des flux sortant de x)

L'information aléatoire

On suppose l'arborescence développée A du graphe G munie d'une probabilité compatible P, et l'on appelle **information aléatoire du graphe ouvert G muni de la probabilité compatible P** la quantité :

$$INFAL\,(G, P) = v^* \,.\, \sum_{y_t \in Y_t} p\,(y_t) \,.\, \log 1/p\,(y_t)$$

ou : $\text{INFAL}(G, P) = v^* \cdot \sum_{C^*} p(c^*) \cdot \log 1/p(c^*)$

en rappelant la bijection qui existe entre les ensembles C^* sur G^* et Y_f de A.

L'information acquise

On établit un certain nombre de résultats (cf.[11, 12]), en procédant schématiquement ainsi : On associe **à tout sommet x** un canal de transmission au sens de Shannon, c'est-à-dire une matrice M(x) pour laquelle nous reprenons les notations mathématiques introductives (cf. p.172) :

- **en entrée** : les sommets y^j de A qui sont images des chemins [$\omega_e \ldots x$], dont la distribution est notée en résumé (α),

- **en sortie** : les sommets z_i successeurs immédiats de x dans G, dont la distribution est notée (β).

On calcule H (α) et H (α/β), puis on applique l'inégalité (1) à partir de laquelle a été définie l'information mutuelle:

$$I = (\alpha, \beta) = H(\beta) - H(\beta/\alpha)$$

On en déduit les contributions de x aux deux mesures définies jusqu'ici : l'entropie systémique et l'information aléatoire, soit $v(x).H_S(x)$ et I(x,P) :

- $v(x) \cdot H_s(x) = v(x) \cdot H(\beta)$, est appelée **entropie systémique du sommet x**, et notée HSYS(x).

- $I(x, P) = v(x) \cdot H(\beta/\alpha)$, est appelée **information aléatoire du sommet x**, et notée INFAL(x,P).

$$I(x, P) = v(x) \cdot \sum_{y^j \in D(x)} \frac{p(y^j)}{p(D(x))} \cdot \sum_i \frac{p(y^j, y^j_i)}{p(y^j)} \cdot \log \frac{p(y^j)}{p(y^j, y^j_i)}$$

où D(x) est l'ensemble des sommets y^j , images de x dans A.

La différence entre les deux mesures est positive et on l'appelle **information acquise du sommet x** : INFAC(x,P), avec, par conséquent l'équation :

$\forall x \in X \quad HSYS(x) = INFAL(x,P) + INFAC(x,P)$

Par sommation sur X, on retrouve :

$$HSYS(G) = HENT(G) + \sum_x HSYS(x), \text{ en notant } HENT(G) = \nu^*. H(\omega_e),$$

$$INFAL(G,P) = HENT(G) + \sum_x INFAL(x,P), \text{ d'après I.2.c.3 et (6)}$$

Enfin, on pose : $INFAC(G,P) = \sum_x INFAC(x,P)$,

où la quantité INFACQ(G,P) est appelée l'**information acquise du graphe ouvert G muni de la probabilité compatible P.**

On a : $HSYS(G) = INFAL(G,P) + INFAC(G,P)$

Remarque :

- l'entropie systémique ne dépend pas de la probabilité individuelle P, mais uniquement des flux collectifs V. C'est une fonction additive de sommets, calculable de façon finie, même si l'ensemble des trajectoires individuelles est infini (boucles, circuits) ;

- l'information aléatoire est maximum, et égale à l'entropie systémique, quand les trajectoires individuelles sont entièrement **indéterministes**. C'est le cas qu'on a appelé plus haut "probabilité proportionnelle aux flux". A chaque noeud du graphe, le cheminement individuel choisit un arc de sortie indépendamment du trajet qu'il a suivi jusque là, à l'image d'un réseau automobile où les usagers tireraient au sort, à chaque carrefour, la poursuite de leur route, les tirages étant indépendants à chaque carrefour et les probabilités étant constantes dans le temps ;

- l'information aléatoire est minimum, et égale à l'entropie d'entrée externe HENT, quand les trajectoires individuelles sont entièrement **déterministes**. A chaque noeud du graphe, elles empruntent une issue bien déterminée, unique, à l'image d'un réseau automobile où des usagers variés accompliraient chaque jour le même trajet. Alors, l'information acquise est maximum et égale à HSYS - HENT.

Propriétés de l'entropie systémique

Facteur structurel du coût de régulation d'un réseau d'échanges matériels

Lorsque l'on régule un réseau d'échanges matériels, c'est-à-dire lorsqu'on propage à travers le réseau une perturbation externe (par exemple, de la demande externe en économie) d'un montant unitaire - élémentaire, atomique - , on doit faire, à chaque noeud, le choix d'un arc adjacent pour rééquilibrer le flux, et, pour cela, consommer un certain nombre de tests binaires ou *bits*, dont le temps d'exécution - et la consommation d'énergie - dépend du système employé (matériel, logiciel). On montre néanmoins (cf. [11], [12], [13]) que le coût global de régulation du réseau d'échange G peut s'écrire de façon théorique :

$$W(G) = \beta \Theta \ HSYS\ (G),$$

avec :

Θ nombre de perturbations à réguler par unité de flux externe (assimilable à une "température" du milieu de G, par exemple fluctuabilité de la demande externe)

β temps unitaire (ou coût unitaire) du test binaire

HSYS(G) entropie systémique du graphe G

Remarque : Le même résultat s'applique au problème de la régulation d'un modèle économique d'entrée-sortie régionalisé, multipériodes et multitechnologies, celle-ci étant opérée à l'aide de cheminements individuels aléatoires selon la méthode de Monte-Carlo. Il s'en déduit une possibilité de rendre endogène le coût de la régulation du modèle (cf. [15],[16]).

En toute rigueur, la méthode ne fonctionne que si la matière échangée est suffisamment **atomisée** (par lots discontinus). On relie l'atomicité requise à la précision de la régulation obtenue.

La mesure d'entropie systémique permet, par exemple, de comparer le coût de régulation d'un réseau matériel unique à celui de deux sous-réseaux séparés et en échange mutuel (cf. [16],[17]) - choix de décentralisations -, ou encore l'inertie d'un carrefour routier avant et après son aménagement (cf. [15]). Enfin, le même calcul s'applique à

un réacteur biochimique combinant différentes synthèses imbriquées (cf. [12]), et l'on en déduit un équivalent thermodynamique du test binaire, ou *bit*, du même ordre de grandeur que celui établi par Brillouin [1].

Indicateur de complexité structurelle d'un réseau d'échange matériel

Ce qui vient d'être dit de l'entropie systémique incline à penser qu'elle reflète une certaine complexité d'un réseau, au sens d'une lourdeur, d'une inertie globale face aux perturbations à réguler.

Comparons deux mesures d'entropie **pour une simple distribution de fréquences** $\{n_i / N : i = 1 \text{ à } k\}$:

- la mesure classique de Shannon $H = \sum_i n_i / N \log N / n_i$;

- la mesure systémique HSYS = N. H

Pour k=8 , N= 8 et $n_i = 1 \quad \forall$ i, on obtient H = 3

et HSYS = 24

Ajoutons une issue, de fréquence 8, à la distribution initiale. On obtient maintenant :

H = 2,5 et HSYS = 40

C'est dire que la mesure classique n'a pas reflété la complexification opérée, puisqu'elle a subi une diminution, tandis que la mesure systémique, au contraire, l'a bien enregistrée puisqu'elle a augmenté.

On montre que le fait est général et que la mesure HSYS augmente toujours, sur une distribution, lorsque :

- on ajoute une issue ;

- on augmente la fréquence d'une issue ;

- on remplace une issue par deux issues en conservant la fréquence totale.

L'entropie systémique donne bien une indication fidèle de la complexité d'une structure.

Pour un réseau d'échange matériel - ayant la structure mathématique de graphe ouvert G -, maintenant, on montre que l'entropie systémique augmente lorsque :

- sans modifier l'ensemble des sommets X, on ajoute de nouveaux arcs entre eux, ou qu'on intensifie les flux des arcs anciens ;

- on ajoute des sommets et des arcs en conservant le réseau ancien (cf. [12] p.45).

En revanche, dans une opération de désagrégation, c'est-à-dire de remplacement d'un simple noeud x par un sous-graphe ouvert plus détaillé $G(x)$, l'entropie systémique **peut augmenter ou diminuer**, selon que le sous-graphe "détaillant" est de type "mélangeant" ou de type "séparant". L'entropie systémique donne précisément un critère pratique pour distinguer les deux situations: bonne désagrégation / mauvaise désagrégation - dans le domaine des branches économiques, ou dans un problème de carrefour routier -.

Formules de calcul de l'entropie systémique

Formule 1 (rappel) $HSYS = v^* . H(\omega_e) + \sum_X v(x) . H_s(x)$

Formule 2 (rappel) $HSYS = v^* . \sum_{C^*} p(c) . \log 1/p(c)$

où P est la **probabilité proportionnelle aux flux**

Formule 3 : $HSYS = v^* . H(\omega_s) + \sum_X v(x) . H_e x$

où $H(\omega_S)$ est l'entropie de Shannon des flux entrant dans ω_S,

et $H_e(x)$ est l'entropie de Shannon des flux entrant dans x.

Formule 4 : $HSYS = \sum_{X^*} v(x) . \log v(x) - \sum_{\Gamma^*} v(\gamma) . \log v(\gamma)$

On remarque que les sommets et les arcs jouent des rôles **antagonistes** par rapport à la complexité et l'inertie du réseau. Du fait de la concavité de la fonction $u \log u$, l'entropie systémique est plus faible quand, eu égard aux contraintes, **les arcs sont le plus concentré possible et les sommets le plus déconcentré possible.** Ainsi, pour un réseau

automobile, où l'on cherchera à économiser le *stress* des conducteurs aux intersections - fatigue nerveuse, perte de temps pour cause de ralentissement hésitant, fréquence de collision - on choisira un réseau en forme d'autoroutes avec des accès allégés, plutôt qu'un transfert enchevêtré conflictuel ressemblant à la place de l'Étoile à Paris.

En utilisant à l'envers la formule de coût moyen sur les graphes obtenue ci-dessus, on voit que si l'on voulait un jour facturer aux usagers le coût de la régulation d'un réseau automobile, soit $\beta\ \theta$ HSYS(G), il faudrait que chacun paye à chaque noeud traversé un coût proportionnel à log $v(x)$, et que la société lui reverse à chaque tronçon de voie emprunté une prime en log v (γ). Comme il s'agit de quantités liées à des flux collectifs, un ordinateur pourrait calculer le bilan pour chaque trajet personnel et le facturer automatiquement sur le ticket de l'usager.

Réseaux cognitiques. Repliement des Quids.

Représentation mathématique du connecteur élémentaire

Un système-expert est composé de connecteurs ou règles ou tables de décision, dont la structure élémentaire est la suivante :

L : SI condition c ALORS action a_1
SI condition $\overline{c}$ ALORS action a_2

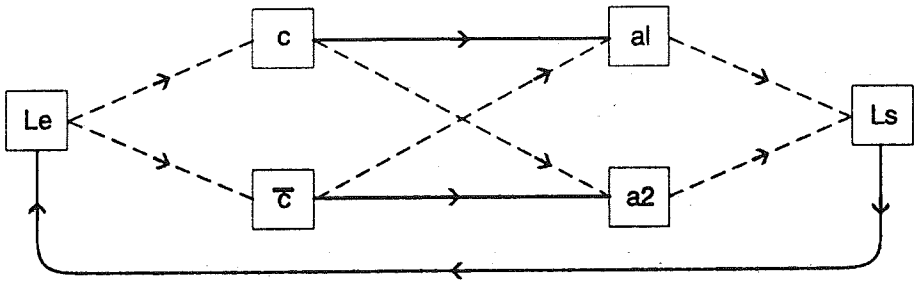
L' : SI condition c' ALORS exécuter L
SI condition $\overline{c}'$ ALORS etc.

Les conditions c et $\overline{c}$ doivent être contradictoires, afin que la machine sache que faire dans tous les cas de figure. En revanche, les actions a_1 et a_2 ne sont pas forcément contraires, mais simplement différentes en général, et même identiques sans inconvénient.

Dans le cas d'une **règle floue**, on a :

L : SI condition c ALORS action a_1 (0,80)
action a_2 (0,20)

La structure élémentaire de la cognitique se représente mathématiquement par le graphe ouvert de la figure 2.



Réseau cognitique du connecteur élémentaire

Les arcs en traits gras continus sont ceux du graphe au sens classique. Les traits gras discontinus sont ceux de "l'extérieur immédiat" du système. Ce sont aussi des arcs d'un graphe mais complété à partir du précédent par l'adjonction de deux sommets supplémentaires :

L_e , qui représente l'environnement-amont de la règle. C'est l'extrémité-amont commune à l'ensemble des arcs d'entrée dans le graphe.

L_s , qui représente son environnement-aval, extrémité-aval commune à l'ensemble des arcs sortants du graphe.

En termes cogniticiens, on peut dire aussi que L_e et L_s sont les deux "faces" de la règle L : L_e comme objet d'appel par d'autres connecteurs, L_s comme lieu, adresse (laissée "en blanc") d'un appel ouvert.

Enfin, le trait fin continu indique le possible retour au départ, la possible réinstanciation de la règle. Mais il **ne fait pas partie de la représentation de la règle** elle-même, et ne figure ici que pour mémoire. Ce qui vient du dehors n'appartient pas au connecteur lui-même mais à son environnement.

Enfin, les traits fins discontinus représentent le cas de la règle probabiliste, ou plus généralement floue.

Réseaux cognitiques des systèmes-experts

Nous entendrons par réseau cognitique d'un système-expert le modèle de graphe ouvert qui généralise celui du connecteur élémentaire au cas du système-expert (cf. Figure 3).

L'ensemble des sommets est réparti en trois couches :

- celle des étiquettes de règles (L:), occupant la position du sous-ensemble mathématique des sommets d'entrée X_e,

- celle des conditions de règles($c/\overline{c}$), occupant la position du sous-ensemble des sommets d'articulation X_a ,

- celle des actions finales (a_1,a_2), occupant la position du sous-ensemble des sommets de sortie X_s .

Les ensembles d'arcs sont moins complets que dans le modèle mathématique général : aucun arc ne revient de X_s vers X_a ou X_e, aucun arc ne va directement de X_e à X_s, aucun arc n'est interne à X_e, à X_a, ou à X_s. Toutefois, il existe des arcs de retour de X_a vers X_e, et donc des circuits *a priori* possibles.

Le réseau cognitique d'un système-expert n'est qu'un **résumé statistique**, une photographie globale du système, à l'image d'un trafic automobile vu d'avion comme un flot continu. Il ignore les chemins individuels d'entrée-sortie, car il laisse indéterminé - mais sous contraintes globales- le questionnaire arborescent déplié qui, seul, représenterait le savoir exhaustif du système et son fonctionnement complet dans tous les cas répertoriés.

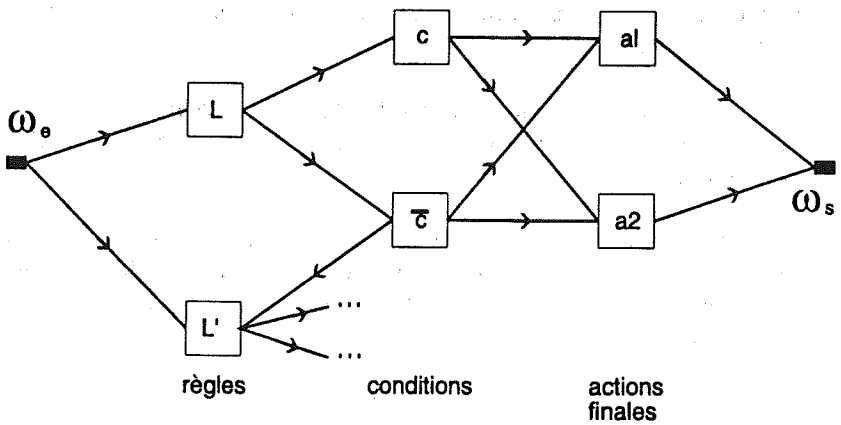


Fig.3 Réseaux cognitiques des systèmes-experts

Réseaux cognitiques et réseaux de transports

Les réseaux cognitiques ressemblent aux réseaux de transport bien connus en recherche Opérationnelle. Mais ils s'en distinguent par leur nature même, qui consiste à ne rien transporter de matériel. Les flux sont uniquement des cumuls de fréquences d'occurrence de règles. Les appels extérieurs du système-expert sont considérées comme des perturbations à réguler.

Réseaux cognitiques et réseaux neuronaux

Les réseaux cognitiques ressemblent par un point essentiel aux réseaux neuronaux, à savoir le fait que les sommets y jouent un rôle effacé, secondaire par rapport aux arcs et plus encore aux chemins.

Dans les réseaux neuronaux, chaque sommet est un "neurone formel", et simule, comme son nom l'indique, un neurone biologique. Le concept est dû à McCulloch et Pitts (1943) et résume sous une forme schématique les entrées et les sorties de la boîte noire qu'est un neurone physiologique.

Le neurone formel reçoit en entrée des signaux binaires (0/1) pondérés par des valuations d'intensité. Il en fait la sommation. Le résultat est confié à une fonction de seuillage, qui envoie, ou n'envoie pas, un signal binaire en sortie, selon que la somme des entrées dépasse, ou non, le seuil de perception du neurone.

Les arcs du réseau, reliant les neurones deux à deux, représentent les synapses formelles. En 1949, Hebb a proposé une règle d'apprentissage devenue célèbre, qui met l'accent sur le rôle de ces synapses, représentées par les arcs du réseau neuronal. Si deux neurones i et j , connectés entre eux par la synapse (i,j), sont simultanément activés, la valuation d'intensité de l'arc (i,j) doit être renforcée. Ainsi, au fur et à mesure que l'être apprend par la répétition, le paysage des connaissances acquiert son relief, et donc sa diversité de plus en plus fine, et cela dans l'espace des synapses et des enchaînements de synapses, ou encore, en termes mathématiques, dans l'ensemble des arcs et des chemins.

Rappelons que les neurones de notre mémoire ne sont pas des cases : une pour l'idée de pain, une autre pour celle d'amitié, mais de simples "piquets" supportant et séparant l'écheveau des chemins de la connaissance. Le "pain" est un composé de chemins. L'"amitié" est un composé de chemins. Les deux faisceaux ont des connexions évidentes quelque part.

La littérature sur les réseaux neuronaux est abondante et l'on se contentera de recommander au lecteur deux bons ouvrages d'initiation scientifique : Perez [20], Davalo et Naïm [3].

Les réseaux cognitiques, auxquels nous revenons maintenant, partagent donc avec les réseaux neuronaux, deux principes fondamentaux communs : celui du rôle second des sommets par rapport aux arcs et aux chemins, et celui du renforcement par l'usage. Mais ils s'en distinguent par plusieurs traits, notamment l'absence de fonction de seuillage quantitatif, et la signification fonctionnelle déterminée, logique des trois couches de sommets.

Généralités sur le repliement des quids

Étant donné un problème de reconnaissance de Forme, défini par:

- l'ensemble de questions $Q = (q_1,q_2,...,q_m)$;
  de réponses $(a_{11},a_{12},...a_{1r}; a_{21},a_{22},...,a_{2r}; ...; a_{m1},a_{m2},...a_{mr})$;

- la Forme T de modalités $(t_1,t_2,...,t_n)$,

- et l'ensemble d'apprentissage FA,

on sait construire des questionnaires d'identification - en abrégé des quids - permettant la reconnaissance automatique de la modalité de la Forme t_j qu'il convient d'associer à une occurrence à traiter, au vu de ses réponses au quid.

Apprentissage en mode déterministe, en mode aléatoire

Si on applique le **critère infomax** (cf. Annexe QUID) on construit un quid (unique) qui est quasi-optimal du point de vue du temps de traitement et de la compression canonique des données utiles du FA. On peut aussi atténuer la rigueur excessive du critère en construisant des quids à partir d'un critère "info-sous-max" - où l'on prend comme question la seconde par rang des quantités d'information décroissantes -, ou même "info-sous-sous-max" - où l'on prend la question de rang 3 - . On se munit ainsi d'une batterie de quids voisins de l'optimum et dont la réunion est plus efficace que le simple quid infomax. Toutes ces variantes constituent des **apprentissages déterministes** parce qu'à chaque construction, la règle de choix de la question est déterminée.

Une autre variante méthodologique consiste à construire un nombre donné N de quids, comme autant de quids construits en tirant au sort la question suivante, selon des probabilités favorisant les questions les plus discriminantes mais n'excluant pas les autres. Il s'agit alors d'un **apprentissage aléatoire**.

Exploitation en mode déterministe, en mode aléatoire

Au cours de la phase du chiffrement proprement dit, un quid simple ne peut être exploré que d'une façon unique, mais un ensemble de quids, ou "quid multiple", peut l'être de plusieurs manières, et cela **indépendamment de la façon, déterministe ou aléatoire, dont ils ont été construits** au cours de la phase d'apprentissage :

- de façon ordonnée, dans un ordre déterminé. Cet ordre, nous l'avons vu, n'a pas d'incidence sur l'efficacité du résultat puisque l'on réunit les chiffrements obtenus, mais il en a un sur la fiabilité ;

- de façon aléatoire, en tirant au sort un échantillon de quids dans l'ensemble dont on dispose, ou encore, de façon plus dynamique, en tirant au sort chaque question posée, selon des probabilités restant à définir.

Repliement d'un quid simple

On considère un réseau cognitique particulier, dans lequel la couche d'entrée est l'ensemble des questions Q, la couche intermédiaire, l'ensemble des réponses R(Q) et la couche de sortie les modalités de la Forme T (cf. Fig. 4).

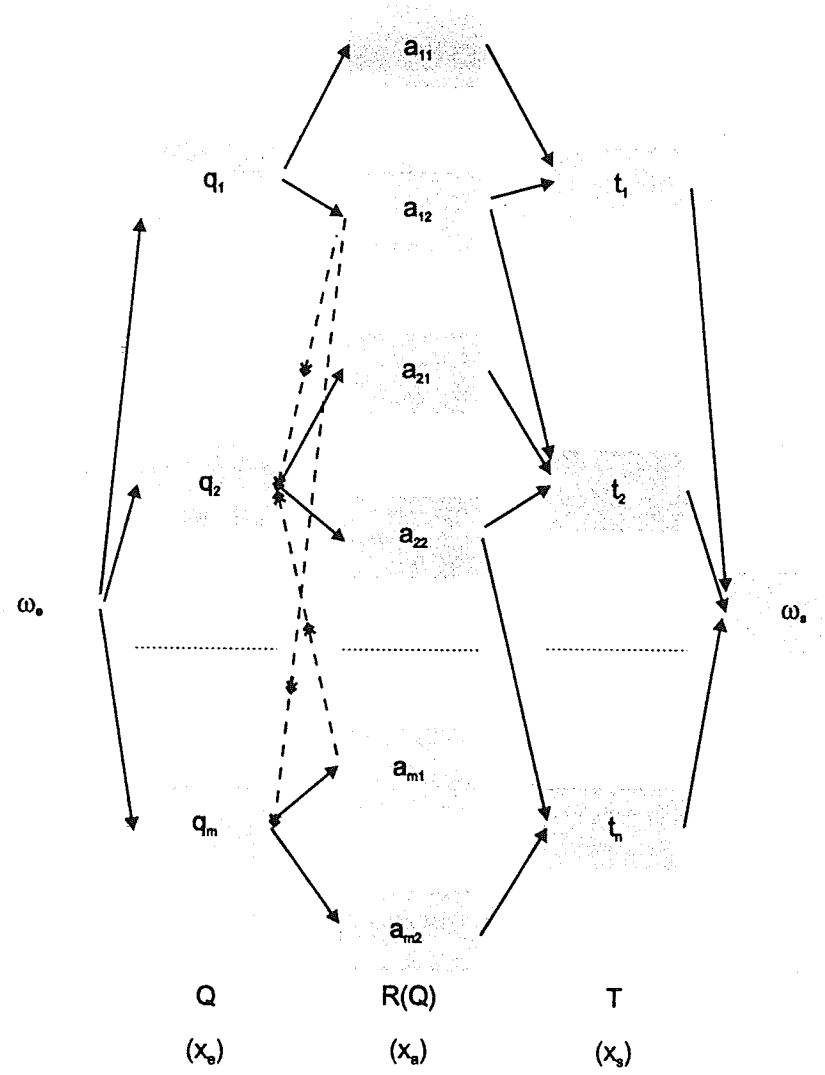


Fig. 4 - Réseau cognitique d'un quid replié

Les chemins d'exploration du quid sont superposés, cumulés en fréquence sur le réseau, sachant bien que, de toute façon, l'on perdra de l'information puisque, pour un arc sortant d'un noeud, rien n'indique plus, à la simple vue du réseau, à quel arc entrant il devait succéder. De la même façon que pour les réseaux cognitiques plus généraux des systèmes-experts, le repliement en réseau n'est qu'un résumé statistique qui occulte l'individualité des chemins réels.

La quantité d'information perdue est précisément ce que nous avons appelé "l'information acquise", INFAC(G,P). En effet, quand on replie un questionnaire arborescent déplié A=(Y, Δ ,P) en un graphe résumé G= (X, Γ ,V), l'entropie - en tant que mesure générale du degré d'incertitude, d'imprévisibilité, d'ignorance - passe de INFAL(G,P) à HSYS(G), et donc augmente d'une quantité égale à INFAC(G,P).

Si l'on voulait conserver tout le savoir dans le graphe replié, il faudrait doter chaque sommet x de la table de décision, - ou, ce qui est équivalent, du canal de communication au sens de Shannon, ou encore de la matrice stochastique de passage - dont l'information acquise est INFAC(x,P) - cf.II.4.c. C'est parce qu'elle concernait des tables de décision "logées" aux sommets du réseau, mais invisibles de l'extérieur, que l'expression "d'information acquise" a été choisie. Dans les réseaux neuronaux classiques, le traitement - occulte - qui s'effectue à l'intérieur des neurones-boîtes noires est résumé par une fonction de sommation et une fonction de seuillage. Ici, le rôle cognitif interne à un noeud x est joué par une table de décision M(x) qui reste invisible à nos yeux, et dont la quantité d'information - au sens de l'information mutuelle de Shannon - est mesurée par ce que nous avons appelé INFAC(x,P).

Remarque : Ces différents termes d'"entropie systémique", d'"information aléatoire", et d'"information acquise" ne sont que des conventions d'écriture. Ce qui compte, ce sont les formules mathématiques sous-jacentes et non pas leur "sens" humain, toujours un peu trompeur. Le terme d'"information aléatoire", par exemple, a été retenu parce qu'il correspondait à la part de cheminement tirée au hasard dans la régulation par la méthode de Monte-Carlo du modèle économique d'entrée-sortie mais, là encore, il ne doit rien signifier de plus que la formule mathématique. D'ailleurs, il ne faudrait pas hésiter, un jour ou l'autre, à rebaptiser ces trois quantités, lorsque nous aurons acquis une expérience suffisante de leur comportement dans les applications réelles.

Le plus difficile est de s'habituer à utiliser alternativement les termes d'entropie et d'information pour caractériser une même entité physique, comme ce fameux verre à moitié plein, et à moitié vide. En fait, **on ne peut mesurer ce qu'on ignore qu'en sachant tout de même ce qu'on mesure.**

Génération automatique de règles indépendantes

Exemple introductif

En dépit de ce qui vient d'être dit, où l'on se plaçait du point de vue strictement théorique et mathématique, les premières mesures effectuées semblent indiquer que l'information acquise est souvent faible par rapport à l'information aléatoire, et qu'on ne perd que peu d'information en repliant le quid. Il faut certes rester circonspect, car c'est peut-être moins vrai pour des quids de plus grande taille que ceux étudiés jusqu'ici, mais l'observation mérite d'être retenue et exploitée.

Or, l'intérêt de la représentation en réseau cognitique est d'abord de fournir une description visible, finie, et, par suite, de permettre la création d'un fichier de règles indépendantes: une pour chaque question réellement posée dans le quid :

Table i : Si $Q_i = a_{ik}$ alors $T = t_j$

ou, selon les cas :
Table i : Si $Q_i = a_{ik}$ alors exécuter Table *i'*

On trouvera à la figure 5 le quid infomax construit sur le fichier FA01 de la Recherche Multiquid (cf. Annexe QUID), et à la figure 6, le réseau replié exprimé sous forme de règles indépendantes. A noter que certaines règles ont été coupées en deux pour tenir dans la largeur d'une feuille de papier, mais, en unités logiques, la table 01 contient 3 règles, la table 07, 5 règles, les tables 02 et 08 chacune 2 règles, soit 12 règles au total pour les 118 lignes du fichier quid.

Les mesures effectuées sur le réseau donnent INFAL = 1174 et INFAC = 79. Le repliement ne crée qu'un seul sommet d'indécision.

SI Rép.(Big.01) = 'CH' on ne peut pas conclure au simple vu du réseau cognitif ; les deux codes CS=21 et CS=22 sont encore possibles.

L'examen du quid arborescent, au contraire, nous aide à conclure :

Si (de plus) Rép.(Big.07) = ' ' alors CS = 21
Si (de plus) Rép.(Big.07) = 'EN alors CS = 22
Si (de plus) Rép.(Big.07) = 'TA' alors CS = 22

La fréquence d'occurrence de ce cas occulté par le repliement est de 3, sur une fréquence totale de 195.

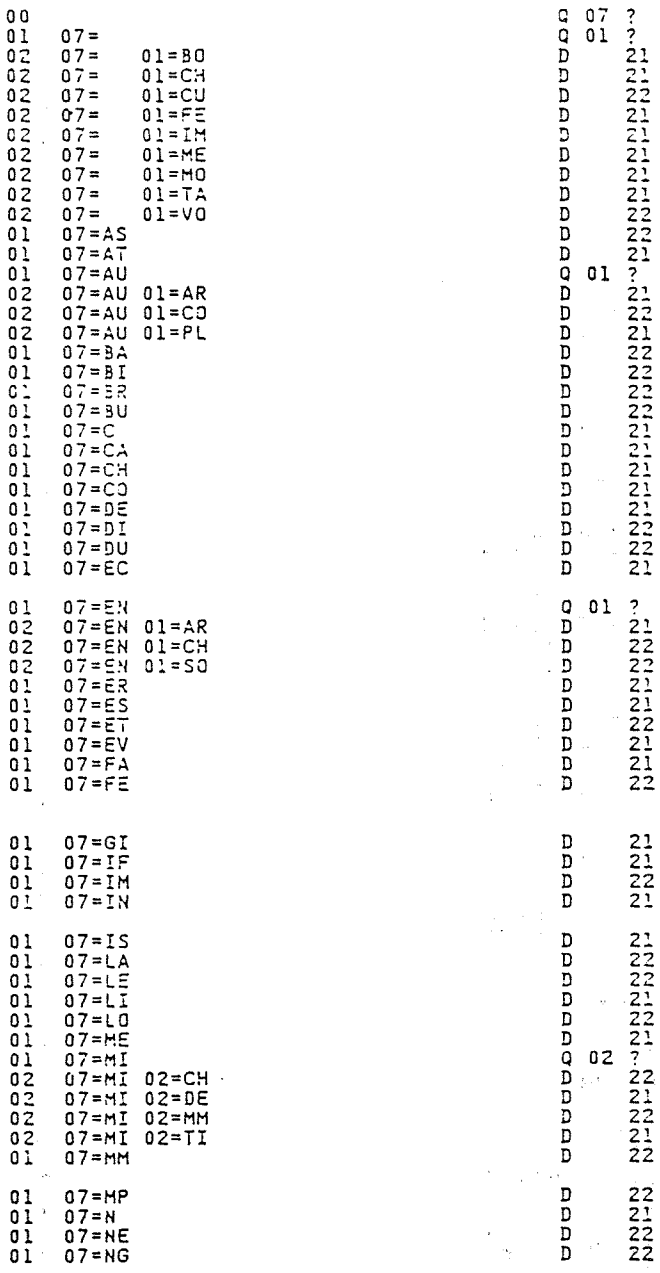


Figure 5 : Exemple de Quid arborescent.

```
01  07=NN                                  D    22
01  07=NS                                  D    22
01  07=NT                                  D    21
01  07=NU                                  D    21
01  07=OD                                  D    22
01  07=OG                                  D    22
01  07=OM                                  D    21
01  07=OP                                  D    22
01  07=PA                                  D    21
01  07=PI                                  D    21
01  07=R                                   D    22

01  07=RC                                  D    22
01  07=RE                                  Q 01 ?
02  07=RE 01=AM                            D    21
02  07=RE 01=CO                            D    22
01  07=RL                                  D    22
01  07=RR                                  D    21

01  07=RV                                  D    22
01  07=SS                                  Q 02 ?
02  07=SS 02=IF                            D    21
02  07=SS 02=MM                            D    22
01  07=ST                                  Q 01 ?
02  07=ST 01=AR                            D    21
02  07=ST 01=CO                            D    22
02  07=ST 01=HO                            D    22
02  07=ST 01=TR                            D    21
01  07=TA                                  Q 01 ?
02  07=TA 01=CH                            D    21
02  07=TA 01=CO                            D    22
02  07=TA 01=MA                            D    22
02  07=TA 01=SE                            D    21
01  07=TE                                  D    21
01  07=TH                                  D    21
01  07=TI                                  Q 08 ?
02  07=TI 08=CL                            D    22
02  07=TI 08=SA                            D    21
02  07=TI 08=SS                            D    21
01  07=TO                                  D    21
01  07=TR                                  D    21
01  07=TT                                  D    21
01  07=UC                                  D    21
01  07=UL                                  D    21
01  07=UN                                  D    22
01  07=UR                                  D    22
01  07=UT                                  Q 01 ?
02  07=UT 01=CO                            D    22
02  07=UT 01=TR                            D    21
01  07=UV                                  Q 01 ?
02  07=UV 01=CO                            D    22
02  07=UV 01=EN                            D    21
01  07=XI                                  D    21
01  07=ZA                                  D    22
```

Figure 5 : Exemple de Quid arborescent (suite et fin).

```
TABLE 01:
 SI  REP(BIG.01) = (AM,AR,BO,EN,FE,IM,ME,MO,PL,SE,TA)     ALORS     DECISION 21
 SI  REP(BIG.01) = (TR)                                  ALORS     DECISION 21
 SI  REP(BIG.01) = (CO,CU,HO,MA,SO,VO)                   ALORS     DECISION 22
 SI  REP(BIG.01) = (CH)                                  ALORS     SUITE    01

TABLE 02:
 SI  REP(BIG.02) = (DE,IF,TI)                            ALORS     DECISION 21
 SI  REP(BIG.02) = (CH,MM)                               ALORS     DECISION 22

TABLE 07:
 SI  REP(BIG.07) = (  ,AU,EN,RE,ST,TA,UT,UV)             ALORS     TABLE    01
 SI  REP(BIG.07) = (MI,SS)                               ALORS     TABLE    02
 SI  REP(BIG.07) = (TI)                                  ALORS     TABLE    08
 SI  REP(BIG.07) = (AT,C ,CA,CH,CO,DE,EC,ER,ES,EV,FA)    ALORS     DECISION 21
 SI  REP(BIG.07) = (GI,IF,IN,IS,LI,ME,N ,NT,NU,OM,PA)    ALORS     DECISION 21
 SI  REP(BIG.07) = (PI,RR,TE,TH,TO,TR,TT,UC,UL,XI)       ALORS     DECISION 21
 SI  REP(BIG.07) = (AS,BA,BI,BR,BU,DI,DU,ET,FE,IM,LA)    ALORS     DECISION 22
 SI  REP(BIG.07) = (LE,LO,MM,MP,NE,NG,NN,NS,OD,OG,OP)    ALORS     DECISION 22
 SI  REP(BIG.07) = (R ,RC,RL,RV,UN,UR,ZA)                ALORS     DECISION 22

TABLE 08:
 SI  REP(BIG.08) = (SA,SS)                               ALORS     DECISION 21
 SI  REP(BIG.08) = (CL)                                  ALORS     DECISION 22
```

Figure 6 : Exemple de Quid exprimé en règles indépendantes (ou replié).

On peut soit accepter un sommet d'indécision pour ce cas spécial, ou préférer ne pas perdre l'information décisive du quid en créant une sous-table SUITE01 de longueur 2 :

SUITE01 :

Si Rép.(Big.01) = (CH) & Rép.(Big.07) = ( ) Alors DÉCISION 21

Si Rép.(Big.01) = (CH) & Rép.(Big.07) = (EN,TA) Alors DÉCISION 22

La comparaison du quid arborescent (Fig.5) et du fichier de règles (Fig.6) montre que la proportion des règles de longueur 1 est bien plus forte que celle des décisions de niveau 1 du quid déplié. **La représentation par les règles est plus synthétique que celle du quid**, et donc plus avantageuse du point de vue du travail d'expertise, où le confort dans la communication homme-machine est prédominant. En revanche, le quid arborescent conserve sa primauté en matière de temps de traitement et de compression des données actives au cours de la phase d'exploitation.

Comparons maintenant le chiffrement par les quids et par les règles (déduites des quids). Lorsqu'on dit que les tables sont "indépendantes", on entend par là que chacune peut servir pour chiffrer certains cas possibles, et qu'on peut *a priori* les employer **dans un ordre quelconque**, contrairement au chiffrement par des quids où l'ordre des questions est fixé par l'apprentissage . D'où **l'algorithme de chiffrement par les tables** :

Pour chiffrer une occurrence donnée, on essaye successivement toutes les tables connues, **dans un ordre choisi** au départ, et l'on s'arrête dès que l'on obtient une solution.

Remarquons tout de suite que le résultat du chiffrement dépend de "l'ordre choisi au départ". Ainsi, si l'on chiffre le FA01 lui-même à l'aide des tables ci-dessus qui en ont été extraites, on obtient ceci :

- si l'ordre choisi commence par TAB07, l'efficacité du chiffrement est totale et la fiabilité est parfaite, puisque l'on ne fait que décalquer le quid.

- si l'ordre choisi commence par une autre table, telle que TAB01, on aboutit à un certain nombre d'erreurs, ici de l'ordre de 10 %. Par exemple, l'intitulé MAÇON CARRELEUR, de CS 21, se trouve chiffré en CS 22.

Les résultats de la comparaison du chiffrement par les quids et par les règles déduites des quids, sont donnés au tableau de la Figure 7. On constate que le chiffrement par les règles se rapproche plutôt du chiffrement quid-multiple que du chiffrement quid-simple, ce qui se comprend, puisqu'on essaye toutes les tables sans les conditionner mutuellement, en toute indépendance.

méthode	ordre	efficacité (en %)	fiabilité (en %)
quid simple	Big. 07 Big. 02	67,9 72,2	86,5 93,1
quid multiple	(1, 2, 3) (2, 3, 1)	89,1 89,1	86,9 88,4
tables	7, 1, 2, 8 1, 2, 8, 7 2, 1, 8, 7 8, 1, 2, 7 8, 2, 1, 7	85,7 84,3 id id id	85,7 87,0 89,3 87,5 89,7

Fig. 7 Comparaison du chiffrement par les quids et par les règles déduites des quids

La méthode générale, l'algorithme QUIDREG

L'algorithme QUIDREG a pour fonction de traduire un quid - simple ou multiple - en un fichier plat de règles indépendantes, et cela de façon canonique si l'on admet que la réduction préalable opérée, au stade de la construction du quid par le critère infomax - ou des critères voisins - est bien canonique, ou du moins quasi-canonique.

L'algorithme général est le suivant :

- repliement du quid en réseau cognitique ;
- extraction des règles de longueur 1 : il suffit de parcourir tous les sommets-réponses et de retenir ceux pour lesquels HSYS = 0 , ou, ce qui est équivalent, qui n'ont qu'un seul arc de sortie ;
- extraction, du FA, des sous-FA résiduels (par ex. Rép.(Big.01) = CH) ;
- pour chacun d'eux, exécution d'un APPREND pour obtenir un quid partiel relatif au sous-FA ;
- pour chacun de ces quids partiels, retour au point 1 : on obtient ainsi les règles de longueur k ;
- poursuite jusqu'à épuisement du FA : le nombre d'itérations ne peut pas dépasser la longueur du quid initial, et, de plus, on sait que les sous-FA résiduels diminueront bien plus vite (cf. exemple ci-dessus). On peut aussi s'arrêter avant l'épuisement du FA et accepter des indécisions résiduelles.

Le logiciel existe en version Recherche pour les deux premiers points. Il reste à rédiger l'enchaînement, la gestion des lots et le contrôle de fin.

Repliement des quids multiples

On commence par replier en réseau cognitique N quids obtenus par un apprentissage déterministe ou aléatoire. Pour cela, les N faisceaux de chemins sont disposés sur le réseau unique, et les fréquences sont cumulées localement.

Exploitation en mode déterministe

On peut ensuite, en appliquant l'algorithme QUIDREG extraire du réseau les règles de longueur 1, puis de longueur 2, etc., et les exploiter en système-expert classique.

Exploitation en mode aléatoire

On peut aussi imaginer une exploration en mode aléatoire, qui serait plus proche du fonctionnement naturel biologique - ce qui n'est pas forcément un critère de pertinence en IA, mais intéresse certainement la science physiologique - . A chaque sommet-réponse du réseau, le sommet-question suivant serait tiré au sort selon des probabilités proportionnelles aux flux sortants : rappelons que ces flux sont des fréquences d'occurrence qui reflètent non seulement la structure du FA de base - somme d'exemples vécus et expertisés -, mais aussi le fruit des optimisations ou sous-optimisations qu'ont réalisées les diverses "quidifications" opérées avant le repliement cumulé. De sorte que ces tirages au sort ne sont pas quelconques ou "chaotiques", mais savants, même dans le mode d'exploitation aléatoire, et cela à travers le jeu du système complexe des **probabilités acquises par le réseau**.

L'entropie systémique HSYS(G) mesure la complexité du réseau cognitique, et, en cela, la richesse de son potentiel de savoir accumulé, de sa capacité de différenciation du monde. Au cours d'une exploitation particulière de ce réseau, sa puissance se réalise en acte, à travers le choix d'une probabilité compatible P. Le savoir se trouve alors séparé en deux parties : une partie visible, mesurée par la quantité INFAL(G,P) et une partie cachée - cachée à notre réseau -, et correspondant aux déterminations internes aux noeuds, mesurée par INFAC(G,P).

Coût d'exploitation d'un réseau cognitique de quid replié (simple ou multiple)

Pour un cheminement individuel correspondant au traitement d'une occurrence de chiffrement proprement dit, le coût d'exploitation a la forme :

$$w(c) = \sum_{q_i \in c} g_1(q_i) + \sum_{a_{ik} \in c} g_2(a_{ik})$$

où l'on distingue le coût g_1 des noeuds-questions q_i (simple temps de consultation du contenu du bigramme dans l'occurrence en cours de traitement) et celui, g_2, des noeuds-réponses a_{ik} (en mode aléatoire : temps de tirage au sort de la question suivante).

En application de la formule de coût moyen sur les graphes ouverts (8), on obtient aussitôt :

$$\overline{w} = 1/v^* \sum_i v(q_i) \cdot g_1(q_i) + \sum_i \sum_k v(a_{ik}) \cdot g_2(a_{ik})$$

où les flux v résultent du cumul des N quids appris sur le FA de base. Notons que durant les N apprentissages élémentaires superposés, le FA a très bien pu évoluer et s'enrichir en cas nouveaux, s'intensifier et se moduler en fréquence d'occurrence, tandis que l'âge cognitif se rapprochait de 1.

QUID 1

QUID 1 est un système général de codification automatique fondé sur une méthode de reconnaissance de formes à base d'exemples. En réalité, le logiciel s'appliquerait aussi bien à tout autre problème de reconnaissance, tel que la lecture automatique de caractères par exemple. Ainsi, on constate dans les applications de l'INSEE que le système s'applique directement à de nouveaux champs sémantiques, tels que la Commune ou les Produits alimentaires, à l'exception de la phase de normalisation préalable des intitulés.

Dans cette phase de normalisation préalable, qui traite chaque intitulé individuellement mais automatiquement, n'importe quelle transformation peut être commandée par l'utilisateur : élimination des "mots vides" - c'est-à-dire appartenant à une liste donnée d'articles, de prépositions, etc.-, phonémisation, extension d'abréviations courantes - telles que ST en SAINT -, génération d'intitulés par synonymie ou *alias*, ou encore par permutation systématique de certains mots, etc. Ainsi, un simple intitulé peut être remplacé par une grappe d'intitulés similaires que le système apprend ensemble. Par le principe même de son **critère infomax** que nous allons voir tout de suite, le système est fait pour accepter des bases de très grande dimension pour un temps d'exploration quasi-instantané - typiquement de l'ordre de la milliseconde pour nos applications réelles. Ainsi, QUID 1 est conçu pour un apprentissage massif à base d'exemples réels. C'est sa particularité. Le dictionnaire des références de nomenclature, qui, dans les autres méthodes employées par certains pays étrangers, sert de "fichier d'apprentissage", ne constitue pour nous qu'une petite partie de la connaissance acquise, une sorte de *bootstrap* de notre base de masse.

Le critère infomax

Le compactage de la base d'intitulés est obtenu grâce à un découpage des intitulés en bigrammes - tranches de deux lettres - et à un choix optimal des bigrammes à consulter, reposant sur le critère de maximisation de l'information au sens de Shannon.

Notons $T = (t_1, t_2, \ldots t_j, \ldots t_n)$ le code à chiffrer, par exemple l'ensemble des modalités du code Profession,

$Q = (q_1, q_2, \ldots q_i, \ldots q_m)$ l'ensemble des bigrammes découpés dans l'intitulé après normalisation préalable, par exemple m=24 si l'on a choisi le nombre 4 comme paramètre "nombre de mots" et 12 caractères comme paramètre "longueur de mot",

X le questionnaire arborescent - ou arbre quid - à construire à partir d'un fichier d'apprentissage (FA) donné.

L'algorithme construit X en descendant du sommet-racine x_0 - placé par convention au niveau 0 - jusqu'aux sommets de niveau 1, puis 2, etc. Au sommet x_0, il associe le FA entier et cherche le meilleur bigramme à interroger en premier, c'est-à-dire le plus discriminant - ou le plus "informant" - possible quant au code cherché T.

Notons $N(x_0)$ la fréquence d'occurrence du FA entier, c'est-à-dire la somme des fréquences accompagnant les intitulés de la base, et $N(x_0,j)$ la fréquence du code t_j dans le FA entier. Nous supposons le FA statistiquement représentatif de la population à chiffrer.

On peut donc estimer la probabilité de rencontrer, au sommet x_0, le code t_j par : $Pr(t_j/x_0) = N(x_0,j) / N(x_0)$. L'incertitude sur le code cherché T se mesure par l'entropie conditionnelle :

$$H(T/X_0) = \sum_j Pr(t_j/x_0) \cdot \log 1/Pr(t_j/x_0)$$

Supposons qu'un bigramme quelconque q_i soit posé pour interrogation au sommet x_0. A chaque modalité $(a_i^1, a_i^2, a_i^k, \ldots)$ qu'il prend dans le FA, nous associons le sous-FA constitué des intitulés possédant cette modalité, et nous attribuons dans l'arborescence X un "sommet-fils" y succédant à x_0 au niveau 1. L'information apportée par ce bigramme q_i au sommet x_0 est mesurée par l'information de Shannon :

$$I(x_0, T, q_i) = H(T/x_0) - \sum_y Pr(y) \cdot H(T/y)$$

où $Pr(y) = N(x_0,a_i^k) / N(x_0)$, est la fréquence relative des intitulés répondant la modalité a_i^k à la question q_i dans le FA entier.

On effectue le calcul d'information pour tous les bigrammes de Q, et l'on choisit le bigramme q_i^* qui maximise l'information I, d'où le nom de *critère infomax* donné à cette règle qui joue un rôle central dans le système QUID.

Pour chaque "sommet-fils" y^* - c'est-à-dire obtenu en fonction du bigramme retenu qi * -, on recommence le même calcul que ci-dessus, mais les fréquences sont maintenant mesurées sur le sous-FA qui lui est associé et non plus sur le FA entier.

etc., jusqu'à épuisement du FA.

Des explications plus détaillées peuvent être trouvées dans [12], [18] ou dans Viglino [24].

QUID 2

QUID 2 a été développé pour traiter le cas particulier des "variables annexes", c'est-à-dire des variables secondaires par rapport à l'intitulé principal, mais qui permettent d'achever le chiffrement cherché, par exemple le code Activité Economique d'une entreprise utilisé pour chiffrer le code Profession de ses employés. Ces variables annexes sont maintenant traitées par des tables de décision qui ont été soit extraites du système Colibri soit rédigées par les statisticiens responsables d'enquêtes. Des codes intermédiaires ont été ajoutés à la nomenclature à codifier, et servent de pointeurs qui appellent les tables de décision à la sortie d'exploration de l'arbre quid. Actuellement, QUID 2 a été développé de façon spécifique à chaque application (Enquête Emploi, RPDOM) mais un projet est en préparation pour en faire un logiciel général qui serait conçu en architecture de système-expert, c'est-à-dire que la base de tables annexes serait gérée de façon externe par rapport aux logiciels du système.

QUID 3 (Recherche Multiquid)

Une nouvelle recherche est en cours depuis deux ans et vise un horizon à plus long terme (décennie 2000).

Depuis une dizaine d'années, nous avions remarqué que le critère infomax pourrait se révéler trop rigide dans certains contextes de données où le premier bigramme, celui du sommet-racine de l'arbre quid, risquerait de nous priver d'autres cheminements de questionnements intéressants, d'où l'idée de construire des quids voisins de l'optimum et de les comparer.

La recherche s'effectue sur un extrait du FA ayant servi au chiffrement de la CS (Catégorie socio-professionnelle) dans le RPDOM (Recensement de 1990 dans les départements et territoires d'Outremer), et qui sert de lot-test de chiffrement pour les mesures d'efficacité (pourcentage d'échos uniques) et de fiabilité (pourcentage de bons chiffrements parmi les échos uniques).

On en tire d'abord un échantillon au 1/10e, le fichier intitulé FA01, qui est employé comme FA pour construire différents quids voisins de l'optimum et comparer leurs performances (efficacité, fiabilité) sur le lot-test. On constate que, parfois, une question *sous-max* - c'est-à-dire arrivée seconde dans le classement par informations décroissantes au sommet-racine de l'arbre - peut donner de meilleurs résultats que la question infomax. L'idée qui vient alors à l'esprit est de procéder au chiffrement d'un même intitulé par toute une batterie de quids voisins de l'optimum - de préférence sur une architecture de machines parallèles, cf. Creecy [2] - et de comparer les échos renvoyés par chacun des quids. La difficulté réside dans le règlement des conflits, et plus largement dans l'interprétation du vecteur de codes obtenus. Dans la méthode un peu primaire envisagée au cours des premiers essais, on aboutit à des résultats qui se résument, par rapport au chiffrement simple infomax (QUID 1), en une amélioration de 10 % de l'efficacité au prix d'une perte de fiabilité d'environ 2 % . Il est d'ailleurs curieux de constater que ces chiffres rejoignent ceux qu'obtient le Bureau du Census des USA dans une recherche similaire, bien qu'utilisant une méthodologie différente.

Un sous-produit de la recherche a consisté à augmenter graduellement l'échantillon d'apprentissage, depuis un taux de sondage de 1/50e jusqu'à l'exhaustivité, et d'observer l'évolution des performances. Il s'agit d'une sorte de simulation de la croissance cognitive depuis l'âge 0 jusqu'à l'âge 1. On aperçoit une discontinuité au passage de l'âge 1/5e, sorte de *crise de l'adolescence*, caractérisée par la fixation définitive des premiers bigrammes interrogés, ceux qui structurent le questionnement aux niveaux majeurs des arbres quids. De sorte que la suspicion que nous éprouvions, au départ, envers la sensibilité excessive de ces premières questions semblerait finalement liée à un stade infantile de l'apprentissage.

En conclusion, il reste deux pistes - au moins - à poursuivre:

- Améliorer la technique de règlement final des vecteurs d'échos renvoyés par le chiffrement multiple, de façon à gagner en efficacité sans perdre en fiabilité quitte à créer quelques cas de rejets pour cause d'indécision.

- Développer la maîtrise des réseaux cognitiques pour l'aide aux ateliers d'expertise (génération automatique de tables résumant une base d'exemples réels).

BIBLIOGRAPHIE

[1] BRILLOUIN, L. : *La Science et la Théorie de l'Information*, Masson, 1959.

[2] CREECY, R. : *Massively Parallel Computing and Automated Industry and Occupation Coding*, US Bureau of the Census (1991).

[3] DAVALO, P. , NAÏM, E. : *Des réseaux de neurones*, Eyrolles, 1990.

[4] FANO, R.M. : *Transmission of information*, MIT Press et J. Wiley, New-York. 1961.

[5] FEINSTEIN, A. : *A new basic theorem of Information theory*, Trans. IRE, 4, 2-22 (1954).

[6] GUIASU, S., THEODORESCU, R. : *La théorie mathématique de l'information*, Dunod, 1968.

[7] HARTLEY, R.V.H. : *Transmission of information*, Bell Syst. Tech. J. , 7, 535-563 (1928).

[8] HUFFMAN, D.A. : *A method for the construction of minimum redundancy codes*, Proc. Inst. Radio. Engrs. 9, 1098-1101 (1952).

[9] KHINCHINE, A.I. : *Mathematical foundations of Information Theory*, Dover (1957).

[10] LORIGNY, J. : *Longueur de cheminements de graphes d'entrée-sortie*, CR Acad. Sci. Paris 279A, 351-354 (1974).

[11] LORIGNY, J. : *Théorie de l'information appliquée aux systèmes sociaux*, Annales de l'INSEE, N43, 47-97 (1981).

[12] LORIGNY, J. : *Mesures d'entropie et d'information pour les systèmes ouverts complexes*, Thèse d'Etat, Paris VI et XII, 1982.

[13] LORIGNY, J. : *Nouvelles mesures d'information dans les systèmes ouverts*, 6ème Cong. int. de cybernétique et de systémique, 489-495, Paris (1984).

[14] LORIGNY, J. : *New measures of entropy and information for social systems*, Systems-letter, International Communic. Assoc., Austin (Texas. USA), 1985.

[15] LORIGNY, J. : *Mesure d'information pour les réseaux*, Analyse de Système, N3, 10-16 Paris (1985).

[16] LORIGNY, J. : *Une approche théorique du coût de l'information dans la régulation du modèle d'entrée sortie*, Économie et Sociétés, (1986).

[17] LORIGNY, J. : *Application de mesures d'entropie systémique aux modèles économiques d'entrée-sortie*, 13ème Colloque Int. d'économétrie appliquée, Sophia-Antipolis, 1986.

[18] LORIGNY, J. : *QUID, une méthode générale de chiffrement automatique*, Techniques d'enquête, Vol. 14, N2, pp. 307-316, Statistique Canada, déc. 1988.

[19] MAC-MILLAN, B. : *The basic theorems of Information Theory*, Ann. Math. Stat. 24, 196-219 (1953).

[20] PEREZ, J.C. : *De nouvelles voies vers l'Intelligence Artificielle*, Masson, 1988.

[21] PICARD, C.F. : *Graphes et Questionnaires*, Gauthier-Villars, 1972.

[22] SHANNON, C.E. : A *mathematical theory of communication*, Bell Syst. Tech. J.27, 379-423, 623-656 (1948).

[23] SIMON, J.C. : *Traitement du signal*, Cours polycopié, Institut de Programmation, Paris (1970).

[24] VIGLINO, L. : *QUID, an automatic coding method.* Application to the Census in the French Overseas Departments, 47ème Congrès de l'IIS (Le Caire 1991).

[25] YAGLOM, A.M. , YAGLOM, I.M. : *Probabilité et Information*, Dunod, 1959.