.

TURBO CODES ET DECODAGE ITERATIF

Didier LE RUYET

Laboratoire Electronique et Communications
CNAM Paris

leruyet@cnam.fr

PROBABILITE A POSTERIORI

la probabilité a posteriori $APP(x_i = a)$ s'exprime comme suit :

$$\begin{aligned} APP(x_i = a) &= Pr\{x_i = a|\mathbf{y}\} \\ &= \sum_{\mathbf{x}} Pr\{\mathbf{x}, x_i = a|\mathbf{y}\} \\ &= \sum_{\mathbf{x}:x_i=a} Pr\{\mathbf{x}|\mathbf{y}\} \\ &= \sum_{\mathbf{x}:x_i=a} \frac{p(\mathbf{y}|\mathbf{x})Pr\{\mathbf{x}\}}{p(\mathbf{y})} \\ &\propto \sum_{\mathbf{x}:x_i=a} p(\mathbf{y}|\mathbf{x})Pr\{\mathbf{x}\} \end{aligned} \tag{1}$$

$$APP(x_i = a) \propto \sum_{\mathbf{x}:x_i=a} p(\mathbf{y}|\mathbf{x}) Pr\{\mathbf{x}\}$$

En utilisant l'hypothèse d'indépendance :

$$APP(x_i = a) \propto \sum_{\mathbf{x}:x_i=a} \prod_j p(y_j|x_j) Pr\{x_j\} \tag{2}$$

Finalement, on obtient la relation suivante:

$$\boxed{APP(x_i = a) \propto Pr\{x_i = a\} p(y_i|x_i) EXTR(x_i = a)} \tag{3}$$

avec la probabilité extrinsèque $EXTR(x_i = a)$:

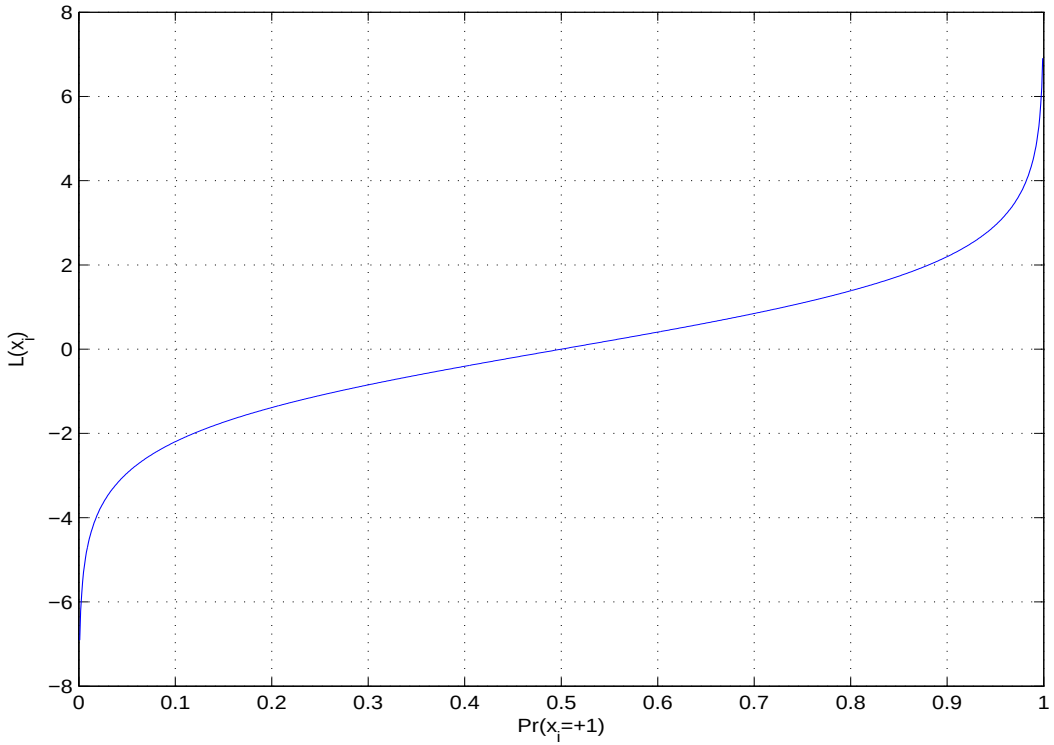
$$EXTR(x_i = a) = \sum_{\mathbf{x}:x_i=a} \prod_{j \neq i} p(y_j|x_j) Pr\{x_j\} \tag{4}$$

• Ainsi la probabilité a posteriori $APP(x_i = a)$ est le produit de trois termes: la probabilité a priori $Pr\{x_i = a\}$, la probabilité conditionnelle $p(y_i|x_i)$ et la probabilité extrinsèque $EXTR(x_i = a)$.

LOGARITHME DE RAPPORT DE VRAISEMBLANCE

- Nous nous restreindrons au cas de la modulation bipodale, $a \pm 1$.
- Il est plus pratique d'utiliser des logarithmes de rapport de vraisemblance LLR (logarithm likelihood ratio en anglais) :

$$L(x_i) = \ln \frac{Pr(x_i = +1)}{Pr(x_i = -1)} = \ln \frac{Pr(x_i = +1)}{1 - Pr(x_i = +1)} \tag{5}$$



$$\ln \frac{APP(x_i=+1)}{APP(x_i=-1)} = \ln \frac{Pr(x_i=+1)}{Pr(x_i=-1)} + \ln \frac{p(y_i|x_i=+1)}{p(y_i|x_i=-1)} + \ln \frac{EXTR(x_i=+1)}{EXTR(x_i=-1)} \quad (6)$$

Cette relation s'écrit simplement:

$$L_{APP}(x_i) = L_{APRI}(x_i) + L_{INT}(x_i) + L_{EXTR}(x_i) \quad (7)$$

- $L_{APRI}(x_i)$ logarithme du rapport de vraisemblance ou information a priori
- $L_{INT}(x_i)$ logarithme du rapport de vraisemblance ou information intrinsèque (issu du canal de transmission)
- $L_{EXTR}(x_i)$ logarithme du rapport de vraisemblance ou information extrinsèque
- $L_{APP}(x_i)$ logarithme du rapport de vraisemblance ou information a posteriori.

INFORMATION INTRINSEQUE POUR CANAL BBAG

• Soit un canal additif à bruit blanc gaussien. La probabilité conditionnelle $p(y_i|x_i)$ s'écrit :

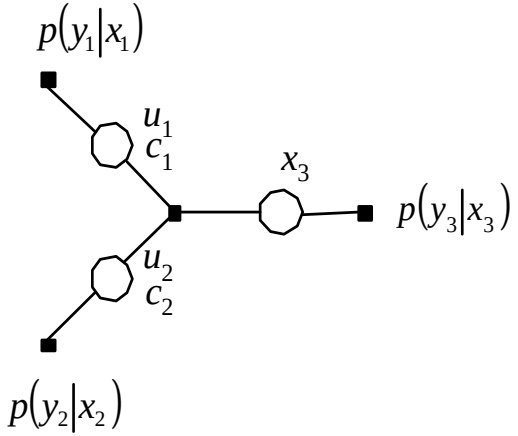
$$p(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i - x_i)^2}{2\sigma^2}\right) \tag{8}$$

• Calculons $L_{INT}(x_i)$ pour un canal BBAG avec $x_t = \pm 1$:

$$\begin{aligned} L_{INT}(x_i) &= \ln \frac{p(y_i|x_i = +1)}{p(y_i|x_i = -1)} \\ &= \ln \frac{\frac{1}{\sqrt{2\sigma^2}} \exp\left(\frac{-(y_i-1)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\sigma^2}} \exp\left(\frac{-(y_i+1)^2}{2\sigma^2}\right)} \\ &= \frac{2y_i}{\sigma^2} \end{aligned} \tag{9}$$

• Ainsi pour le canal BBAG, le signal reçu y_i est proportionnel à l'information intrinsèque $L_{INT}(x_i)$.

CODE DE PARITE (3,2)

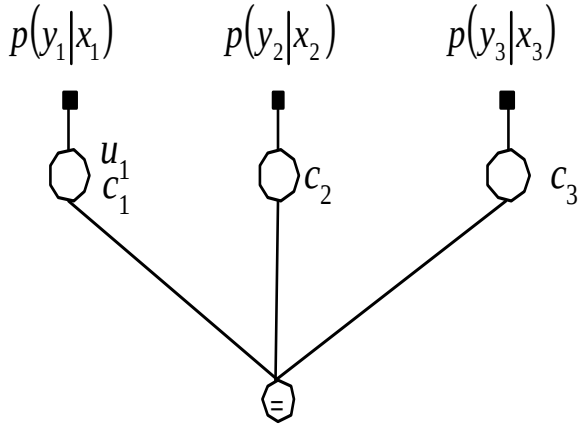


$$APP(x_1 = a) = Pr\{x_1 = a|y_1, y_2, y_3\}$$
$$\propto Pr\{y_1|x_1 = a\}Pr\{x_1 = a\}EXTR(x_1 = a)$$

avec

$$EXTR(x_1 = +1) = \sum_{\mathbf{x}:x_i=1} \prod_{j\neq 1} p(y_j|x_j)Pr\{x_j\}$$
$$= p(y_2|x_2 = -1)Pr\{x_2 = -1\}p(y_3|x_3 = +1)Pr\{x_3 = +1\}$$
$$+ p(y_2|x_2 = +1)Pr\{x_2 = +1\}p(y_3|x_3 = -1)Pr\{x_3 = -1\}$$

CODE DE REPETITION (3,1)



$$APP(x_1 = a) = Pr\{x_1 = a|y_1, y_2, y_3\}$$
$$\propto Pr\{y_1|x_1 = a\}Pr\{x_1 = a\}EXTR(x_1 = a)$$

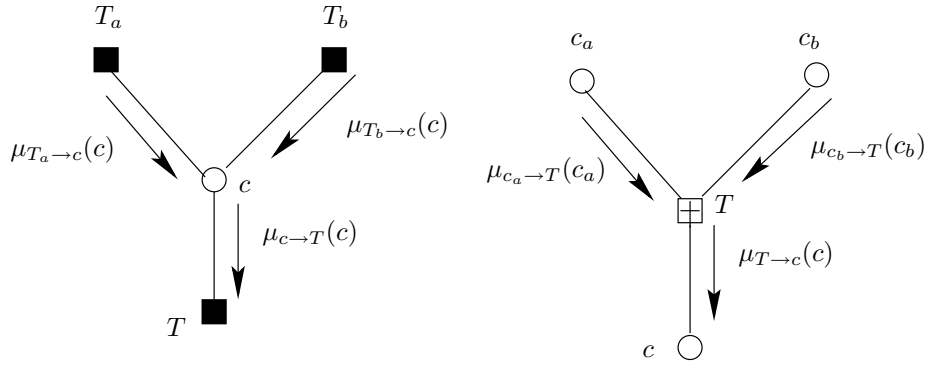
avec

$$EXTR(x_1 = +1) = \sum_{\mathbf{x}:x_i=1} \prod_{j\neq 1} p(y_j|x_j)Pr\{x_j\}$$
$$= p(y_2|x_2 = +1)Pr\{x_2 = +1\}p(y_3|x_3 = +1)Pr\{x_3 = +1\}$$

ALGORITHME SOMME PRODUIT

• Permet un calcul efficace de toutes les probabilités a posteriori $APP(x_i = a)$.

• Les messages qui transitent sur les branches sont des paires de probabilité (p_{i0}, p_{i1}) avec $p_{i0} + p_{i1} = 1$.



• Règle 1 :

$$\mu_{c \to T}(c) = \left(\frac{p_{a0}p_{b0}}{p_{a1}p_{b1} + p_{a0}p_{b0}}, \frac{p_{a1}p_{b1}}{p_{a1}p_{b1} + p_{a0}p_{b0}}\right) \tag{10}$$

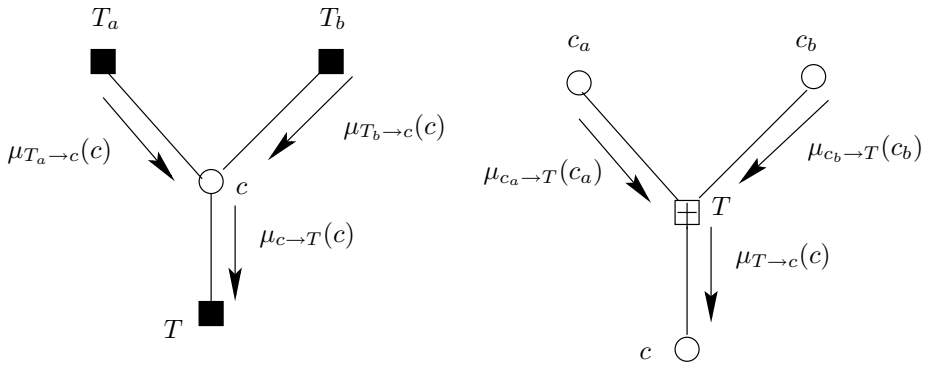
• Règle 2 :

$$\mu_{T \to c}(c) = (p_{a0}p_{b0} + p_{a1}p_{b1}, p_{a1}p_{b0} + p_{a0}p_{b1}) \tag{11}$$

ALGORITHME SOMME PRODUIT

• Permet un calcul efficace de toutes les probabilités a posteriori $APP(x_i = a)$.

• Les messages qui transitent sur les branches sont des logarithmes de rapport de vraisemblance.



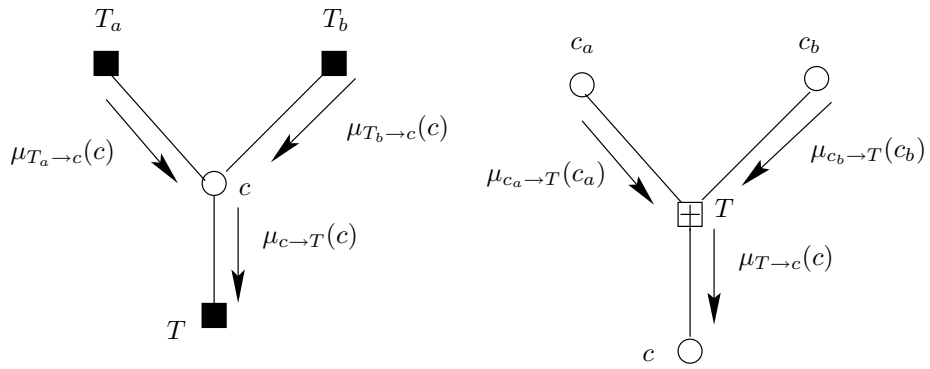
• Règle 1 :

$$\mu_{c \to T}(c) = \mu_{T_a \to c}(c) + \mu_{T_b \to c}(c) \tag{12}$$

• Règle 2 :

$$\mu_{T \to c}(c) = -2 \tanh^{-1} \left(\tanh \left(\frac{\mu_{c_a \to T}(c_a)}{2} \right) \tanh \left(\frac{\mu_{c_b \to T}(c_b)}{2} \right) \right) \tag{13}$$

ALGORITHME MINIMUM SOMME



- Règle 1 :

$$\mu_{c\to T}(c) = \mu_{T_a\to c}(c) + \mu_{T_b\to c}(c) \tag{14}$$

- Règle 2

$$\begin{aligned}\mu_{T\to c}(c) &= -\min\left(|\mu_{c_a\to T}(c_a)|, |\mu_{c_b\to T}(c_b|\right)\,\text{sgn}(\mu_{c_a\to T}(c_a))\text{sgn}(\mu_{c_b\to T}(c_b)) \\ &= -\mu_{c_a\to T}(c_a) \boxplus \mu_{c_b\to T}(c_b)\end{aligned} \tag{15}$$

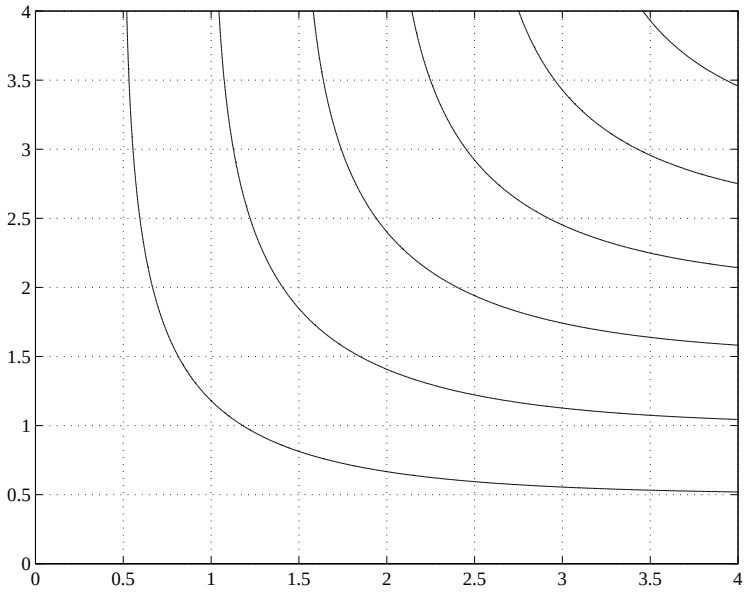
où

$$\text{sgn(a)} = \begin{cases} +1 & \text{si} \quad a \geq 0 \\ -1 & \text{si} \quad a < 0 \end{cases} \qquad \text{et} \quad a \boxplus b = \min(|a|, |b|)\text{sgn}(a)\text{sgn}(b) \tag{16}$$

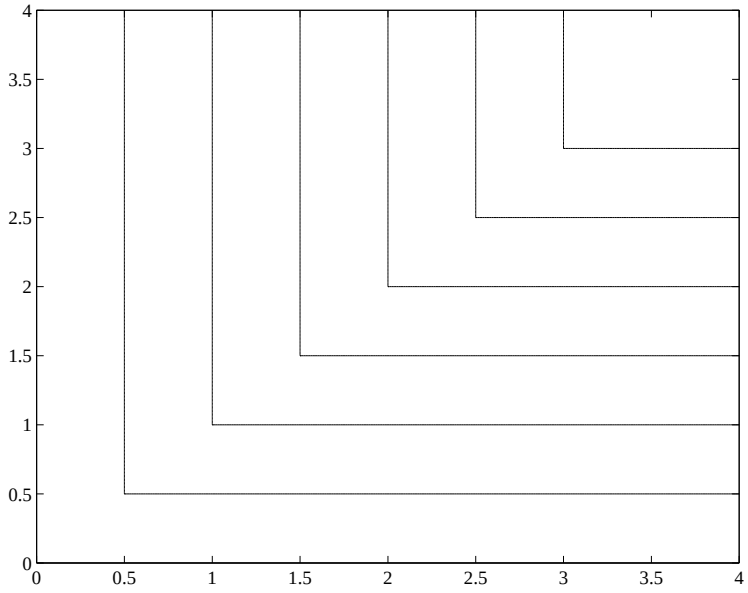
COMPARAISON DES COURBES DE CONTOUR POUR LA REGLE 2

- For $\mu_{T\to c}(c)$=0.5, 1, 1.5, 2, 2.5 et 3 .

(a) Algorithme Somme Produit



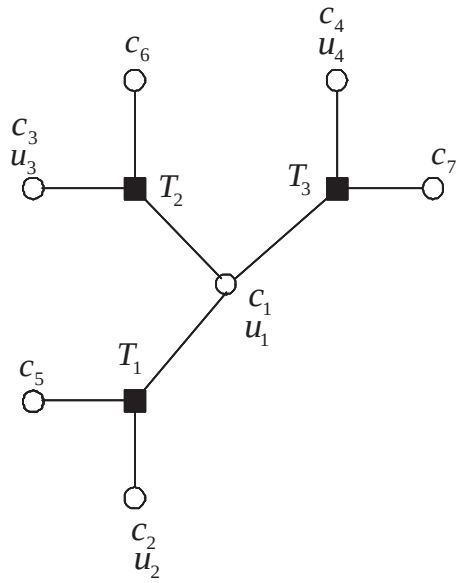
(b) Algorithme Minimum Somme



EXEMPLE : CODE (7,4)

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

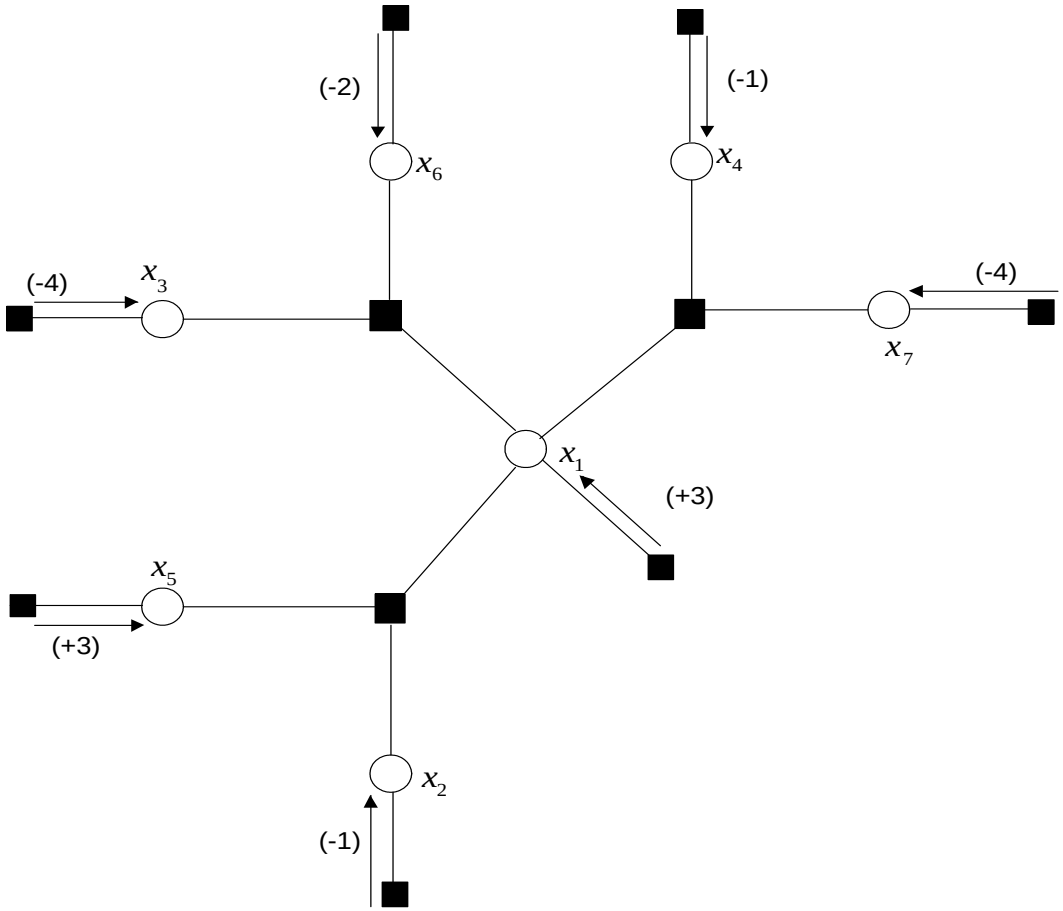
$$\begin{aligned} u_1 + u_2 + c_5 &= 0 \qquad T_1 \\ u_1 + u_3 + c_6 &= 0 \qquad T_2 \\ u_1 + u_4 + c_7 &= 0 \qquad T_3 \end{aligned}$$



EXEMPLE

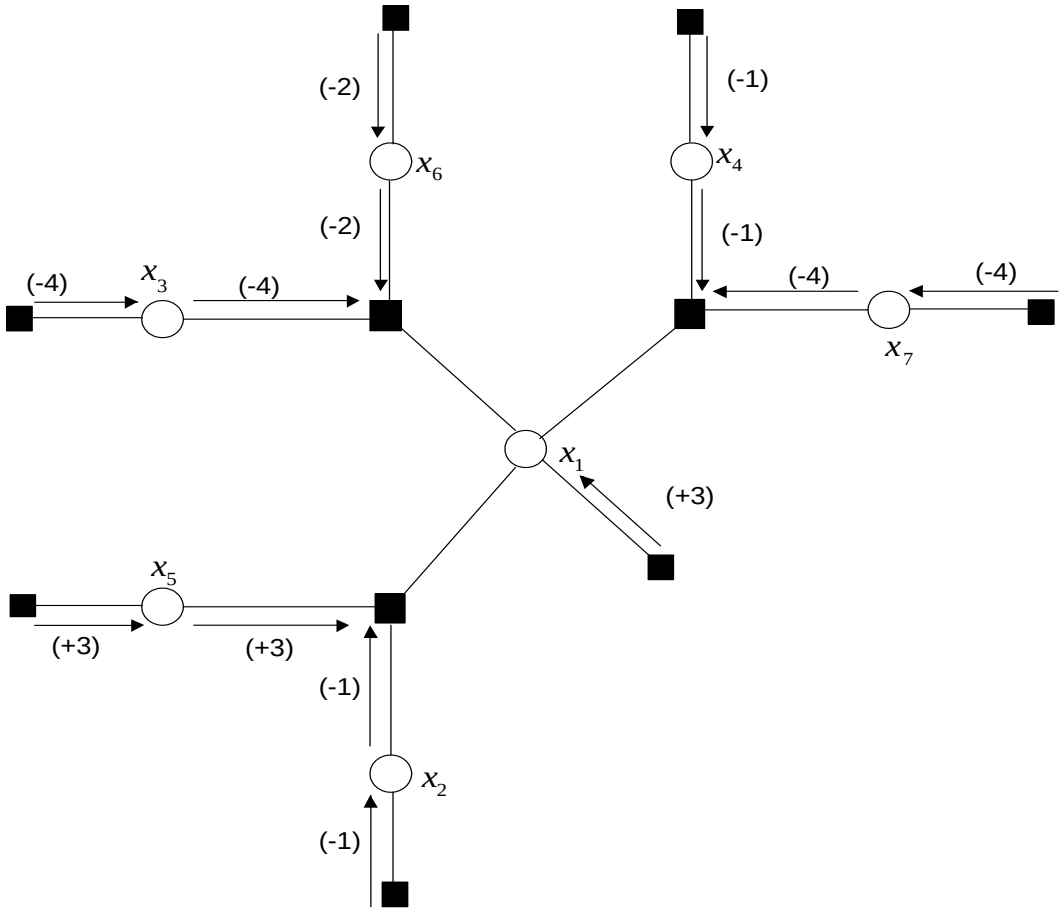
- Mot de code binaire $\mathbf{c} = [1001110]$
- Mot émis : $\mathbf{x} = [+1 - 1 - 1 + 1 + 1 + 1 - 1]$
- Vecteur reçu : $\mathbf{y} = [+1.2 - 0.4 - 1.5 - 0.4 + 1.2 - 0.75 - 1.5]$
- Variance du bruit gaussien : $\sigma^2 = 0.75$
- Information intrinsèque : $\{L_{INT}(x_i)\} = \frac{2\mathbf{y}}{\sigma^2} = [+3 - 1 - 4 - 1 + 3 - 2 - 4]$

EXEMPLE

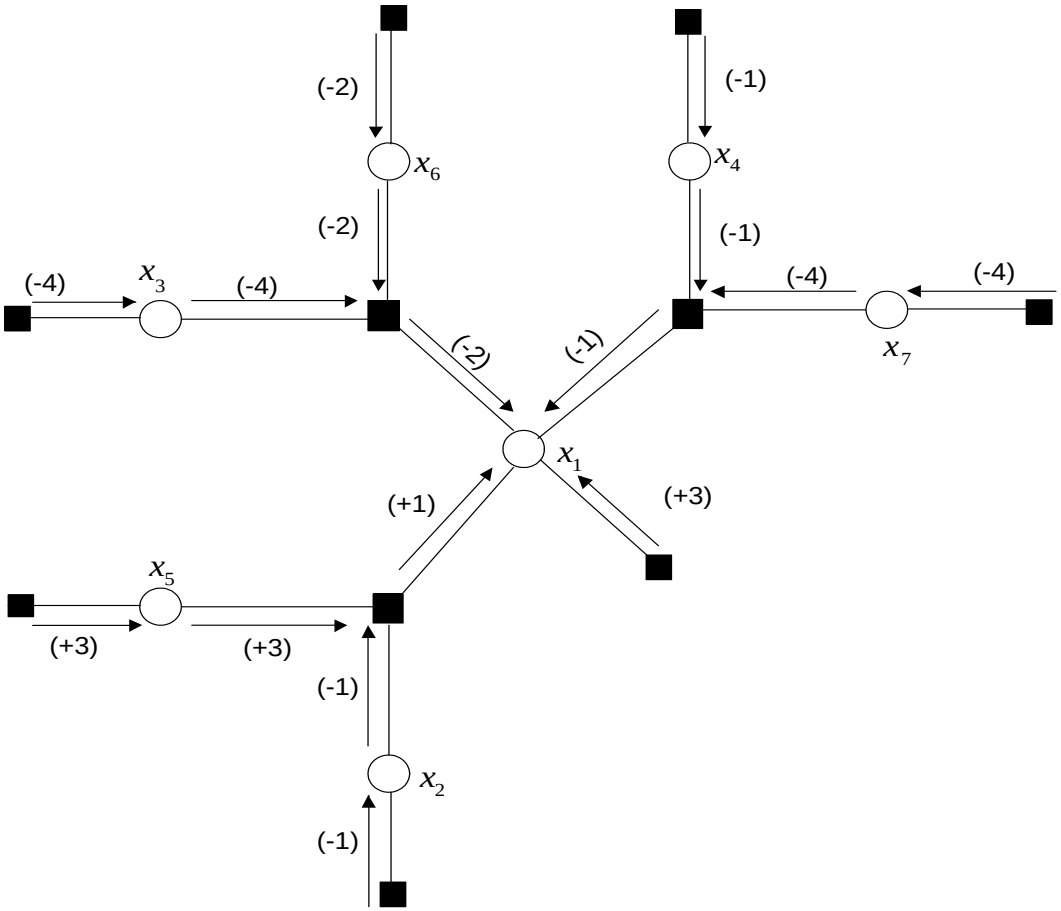


- $\{L_{INT}(x_i)\} = [+3 - 1 - 4 - 1 + 3 - 2 - 4]$

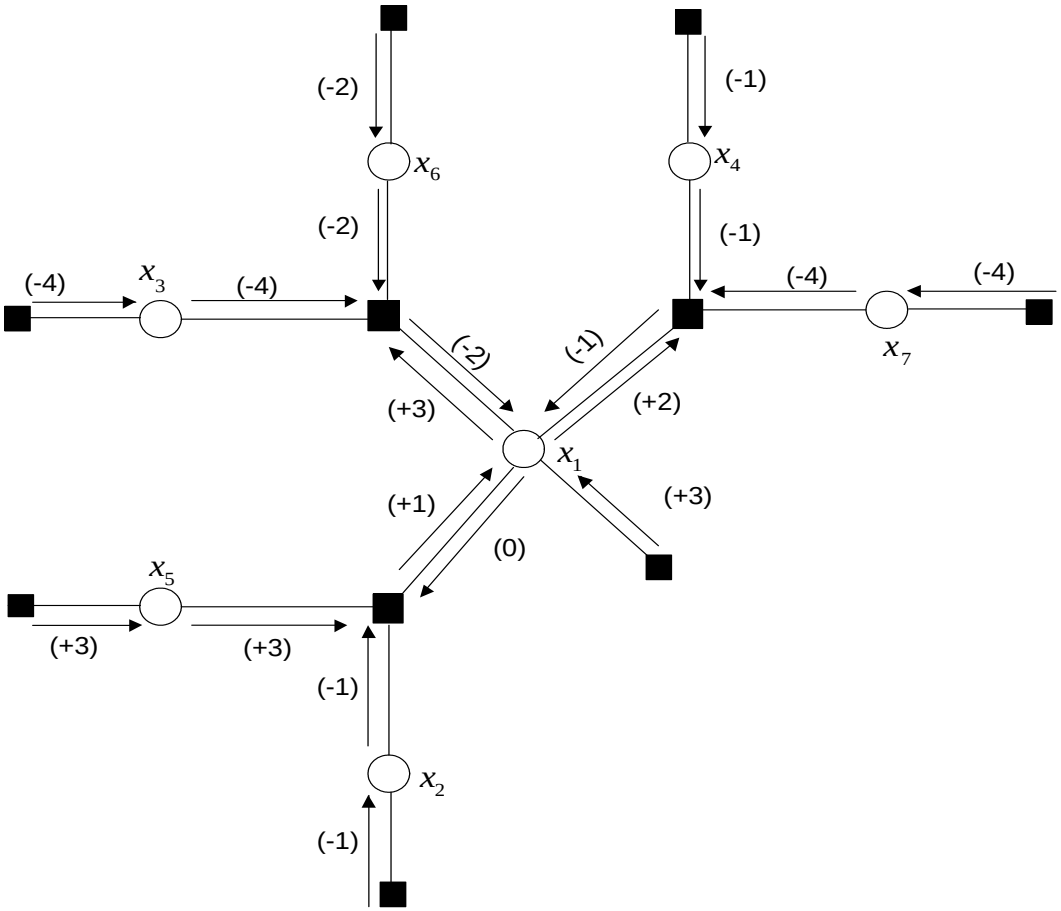
EXEMPLE



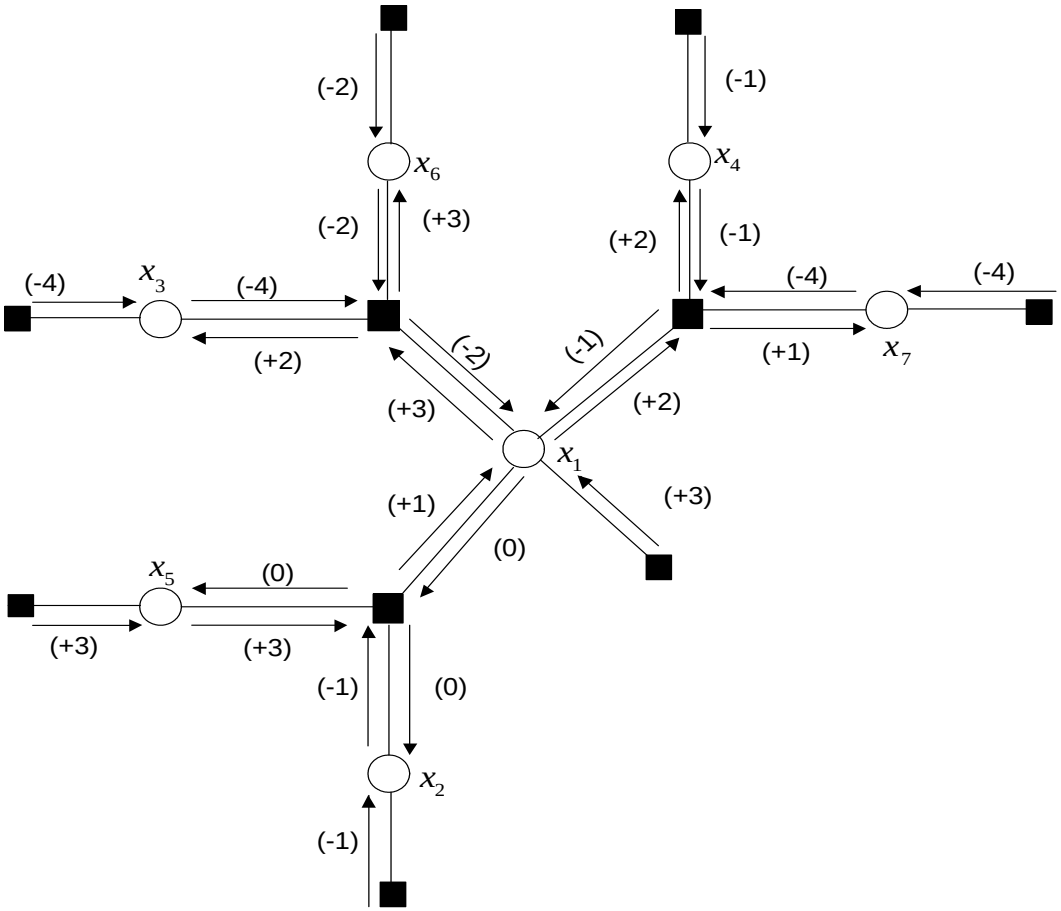
EXEMPLE



EXEMPLE



EXEMPLE



- $L_{APP}(x_i) = L_{APRI}(x_i) + L_{INT}(x_i) + L_{EXTR}(x_i)$.
- $L_{APRI}(x_i) = 0$ en absence d'information *a priori*.
- $\{L_{EXTR}(x_i)\} = [-2 \quad 0+2+2 \quad 0+3+1]$
- $\{L_{APP}(x_i)\} = [+1-1-2+1+3+1-3]$

ALGORITHME ALLER RETOUR

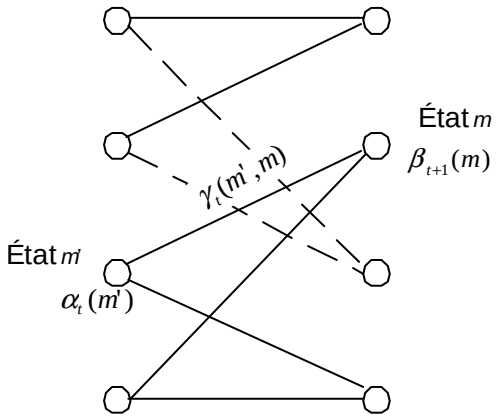
• L'algorithme Aller-Retour ou MAP ou BCJR estime les probabilités a posteriori des états et des transitions d'une source de Markov observée à travers un canal discret, bruité et sans mémoire.

• Un codeur convolutif binaire de rendement k/n peut être vu comme une source markovienne avec $S = 2^M$ états internes, où M est le nombre de cellules mémoires du codeur.

• On considère un code convolutif systématique de rendement 1/2. Soit $\mathbf{x}_t = (x_t^S, x_t^P)$ le mot de code ou sortie du codeur à l'instant t et $\mathbf{y}_t = (y_t^S, y_t^P)$ les échantillons reçus à la sortie du canal sans mémoire.

• L'objectif du décodeur est de calculer la probabilité a posteriori (APP) sur les bits d'information x_t^S.

$$APP(x_t^S = a) = Pr\{x_t^S = a|\mathbf{y}_0^{T-1}\}$$
$$\propto \sum_{m',m/x_t^S=a} Pr\{s_t = m'; s_{t+1} = m; \mathbf{y}_0^{T-1}\}$$



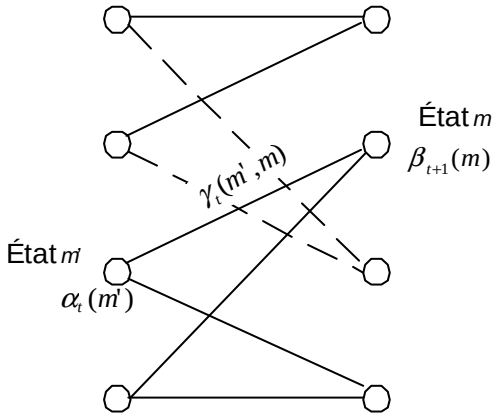
$$APP(x_t^S = a) \propto \sum_{m',m/x_t^S=a} \alpha_t(m')\gamma_t(m',m)\beta_{t+1}(m) \tag{18}$$

avec $$\gamma_t(m',m) = Pr\{s_{t+1} = m; \mathbf{y}_t | s_t = m'\} \tag{19}$$

$$\alpha_t(m') = Pr\{s_t = m'; \mathbf{y}_0^{t-1}\} \tag{20}$$

$$\beta_t(m) = Pr\{\mathbf{y}_t^{T-1} | s_t = m\} \tag{21}$$

• Les quantités $\alpha_t(m)$, $\gamma_t(m', m)$ et $\beta_{t+1}(m)$ représentent l'influence de la séquence reçue respectivement avant l'instant t, à l'instant t et après l'instant t sur la probabilité conjointe $\sigma_t(m', m)$.



Les calculs des probabilités $\alpha_t(m)$ et $\beta_t(m)$ sont effectués par récurrence :

$$\alpha_t(m) = \sum_{m'=0}^{S-1} \alpha_{t-1}(m')\gamma_{t-1}(m', m) \tag{22}$$

$$\beta_t(m) = \sum_{m'=0}^{S-1} \beta_{t+1}(m')\gamma_t(m', m) \tag{23}$$

Le calcul de la métrique de branche $\gamma_t(m', m)$ s'exprime comme suit :

$$\begin{aligned}
\gamma_t(m', m) &= Pr\{s_{t+1} = m; \mathbf{y}_t | s_t = m'\} \\
&= p(m|m') \times Pr\{\mathbf{y}_t | s_t = m'; s_{t+1} = m\} \\
&= p(m|m') \sum_{\mathbf{x}_t \in \mathbb{X}} Pr\{\mathbf{y}_t, \mathbf{x}_t | s_t = m'; s_{t+1} = m\} \\
&= p(m|m') \sum_{\mathbf{x}_t \in \mathbb{X}} p(\mathbf{y}_t \mid \mathbf{x}_t) Pr\{\mathbf{x}_t = \mathbf{x} | s_t = m'; s_{t+1} = m\}
\end{aligned} \tag{24}$$

avec $p(m|m') = Pr\{s_{t+1} = m | s_t = m'\}$ probabilité de transition
$= APRI(x_t^S = x^S(m', m))$

où $x^S(m', m)$ est la valeur du bit d'information associé à la branche (m', m) .

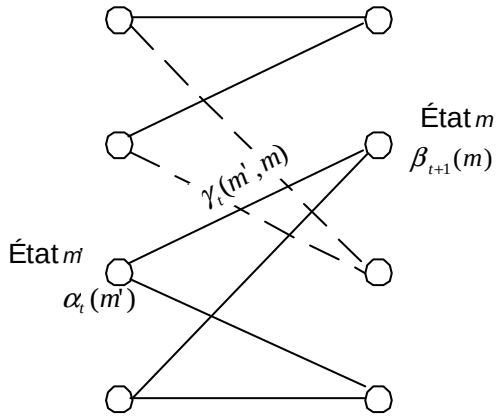
• Si aucune probabilité a priori sur les bits d'information alors $APRI(x_t^S) = 1/2$.

$$Pr\{\mathbf{x}_t = \mathbf{x} | s_t = m', s_{t+1} = m\} = \begin{cases} 1 & \text{si} \quad \mathbf{x}_t(m', m) = \mathbf{x}_t \\ 0 & \text{sinon} \end{cases}$$

- cas du code convolutif systématique de rendement 1/2

$$APP(x_t^S = a) \propto \sum_{m',m/x_t^S=a} \alpha_t(m')\gamma_t(m',m)\beta_{t+1}(m)$$

$$\gamma_t(m',m) = APRI(x_t^S = x_t^S(m',m)) \times p(y_t^S \mid x_t^S(m',m)) \times p(y_t^P \mid x_t^P(m',m))$$



Nous pouvons résumer l'algorithme Aller-Retour en 4 étapes :

- **Initialisation**
  En supposant que l'état initial et final soient l'état zéro

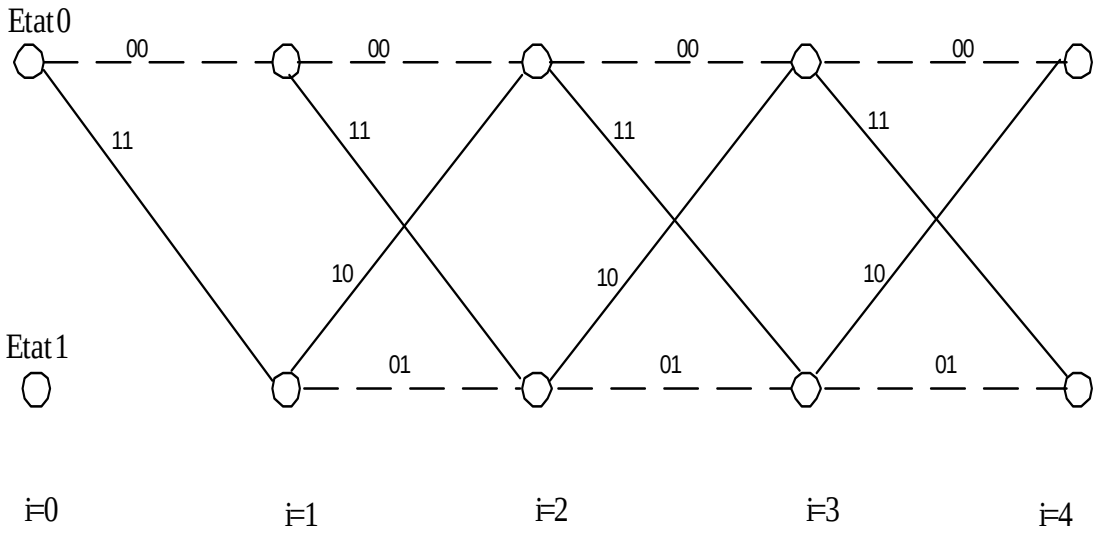
$$\alpha_0(0) = 1 \qquad \text{et} \qquad \alpha_0(m) = 0 \quad \forall m \neq 0$$
$$\beta_T(0) = 1 \qquad \text{et} \qquad \beta_T(m) = 0 \quad \forall m \neq 0$$

- **Calcul des γs et Procédure aller**
  Les γs sont calculés pour toutes les branches et simultanément les probabilités $\alpha_t(m)$ sont calculées
- **Procédure retour**
  Lorsque la séquence $\mathbf{y}_0^{T-1}$ est reçue, les probabilités $\beta_t(m)$ sont calculées
- **Calcul des probabilités a posteriori**

EXEMPLE DECODEUR MAP

Considérons un codeur convolutif récursif systématique de rendement 1/2

$G(D) = \begin{pmatrix} 1 & \frac{1}{1+D} \end{pmatrix}$. Le treillis associé est le suivant :



Soit le mot d'information et le mot de code associé :

i	0	1	2	3	4	5
u_i	1	0	0	1	1	1
x_i^S	+1	-1	-1	+1	+1	+1
x_i^P	+1	+1	+1	-1	+1	-1

Soient les informations intrinsèques L_{INT} reçues du canal BBAG:

i	0	1	2	3	4	5
$L_{INT}(x_i^S)$	+1.21	-1.00	-2.86	+4.84	+0.90	+5.87
$L_{INT}(x_i^P)$	+1.09	-1.07	+3.49	+0.72	+0.77	-1.13

1) calcul de γ

Pour un code convolutif systématique de rendement 1/2, le calcul de la métrique de branche est le suivant :

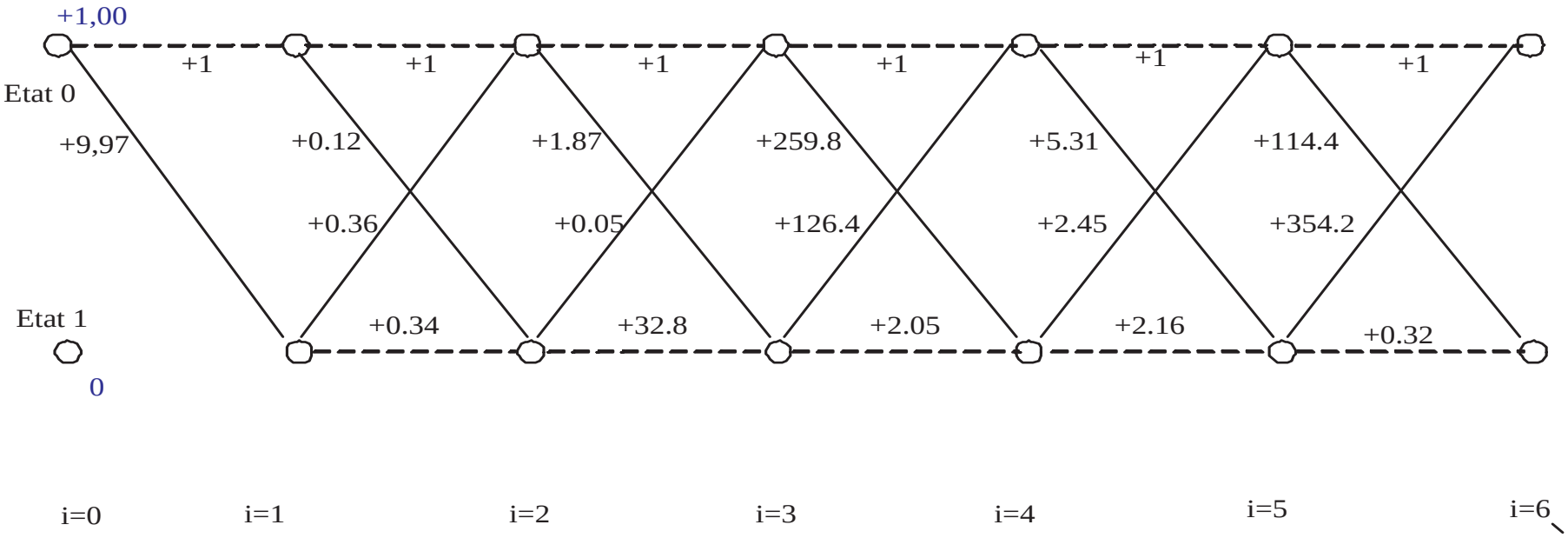
$$\gamma_t(m', m) = APRI(u_t = u(m', m)) \times p(y_t^S \mid x_t^S(m', m)) \times p(y_t^P \mid x_t^P(m', m)) \quad (25)$$

Calculons $p(y_t \mid x_t)$ pour un canal BBAG avec $x_t = \pm 1$:

$$\begin{aligned} p(y_t \mid x_t = a) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_t - x_t)^2}{2\sigma^2}\right) \\ &\propto \exp\left(\frac{2y_t x_t}{2\sigma^2}\right) \\ &\propto \exp\left(\frac{x_t}{2} L_{INT}(x_t)\right) \\ &\propto \exp\left(\frac{(x_t + 1)}{2} L_{INT}(x_t)\right) \end{aligned} \quad (26)$$

Pour notre exemple précédent, en absence d'information a priori, on obtient les métriques de branche γ_t^{pq} suivantes avec p et q qui sont respectivement les bits relatifs à x_t^S et x_t^P :

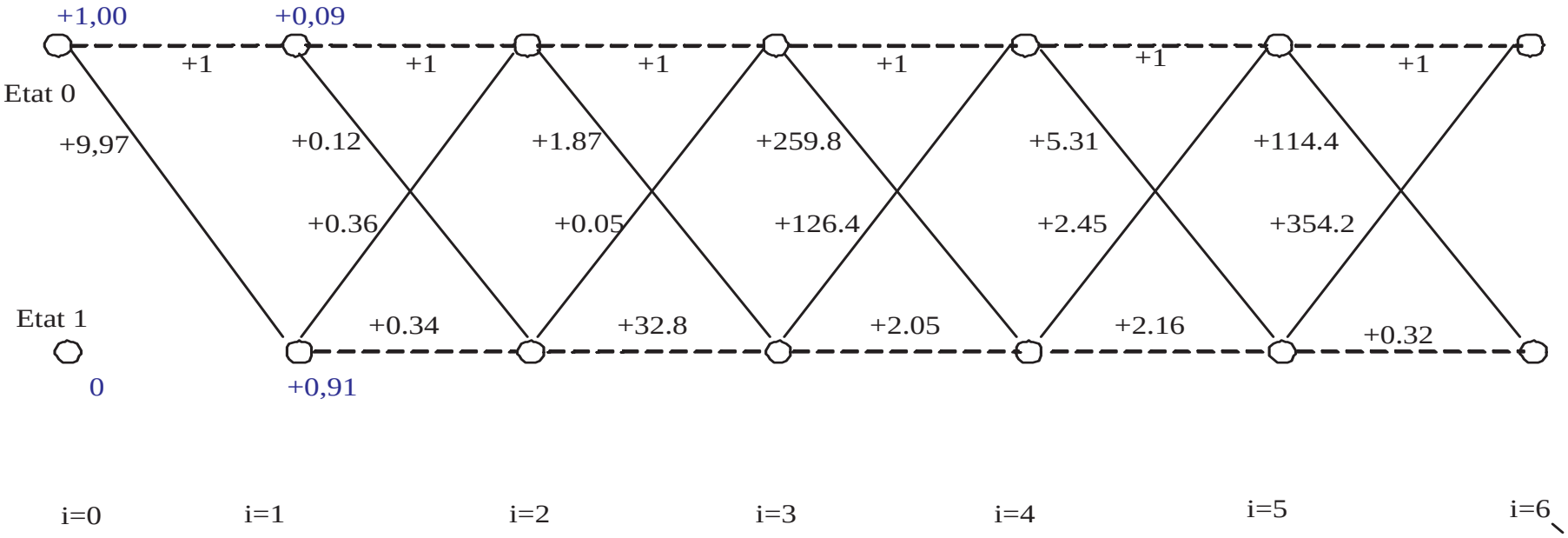
i	0	1	2	3	4	5
γ_i^{00}	1	1	1	1	1	1
γ_i^{10}	3.35	0.36	0.05	126.46	2.45	354.24
γ_i^{01}	2.97	0.34	32.78	2.05	2.16	0.32
γ_i^{11}	9.97	0.12	1.87	259.81	5.31	114.42



2) calcul des α

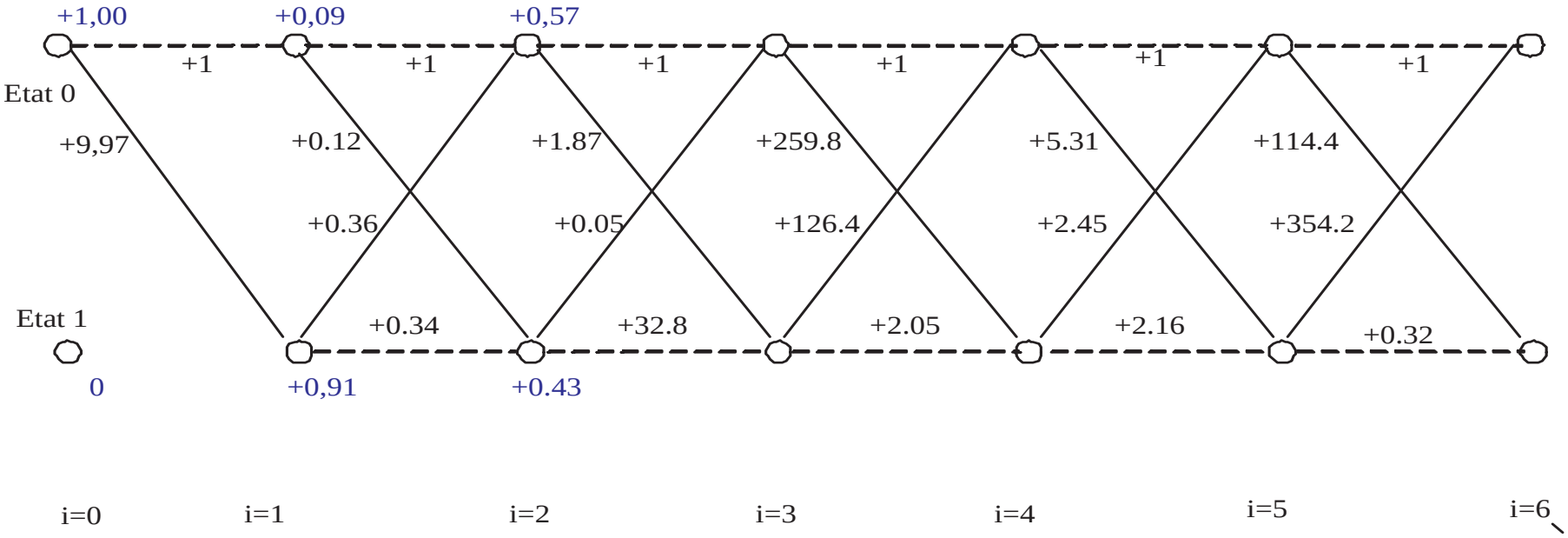
$i = 0$: initialisation

$\alpha_0(0) = 1$
$\alpha_0(1) = 0$



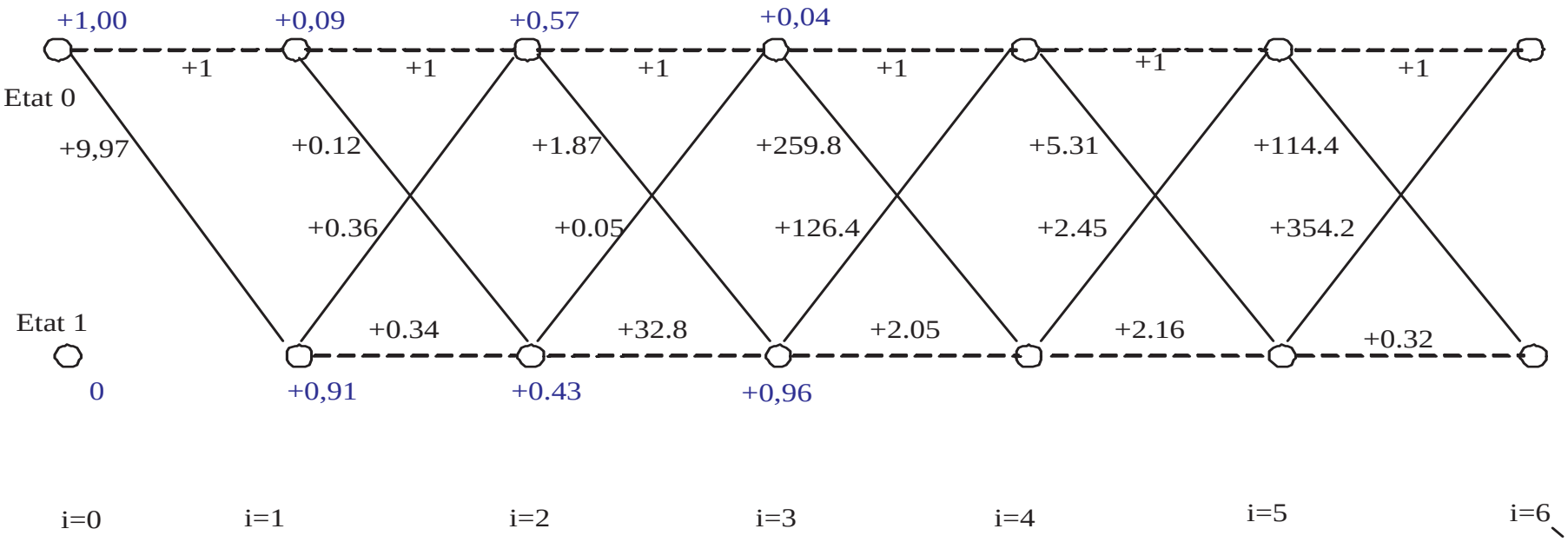
$i = 1$

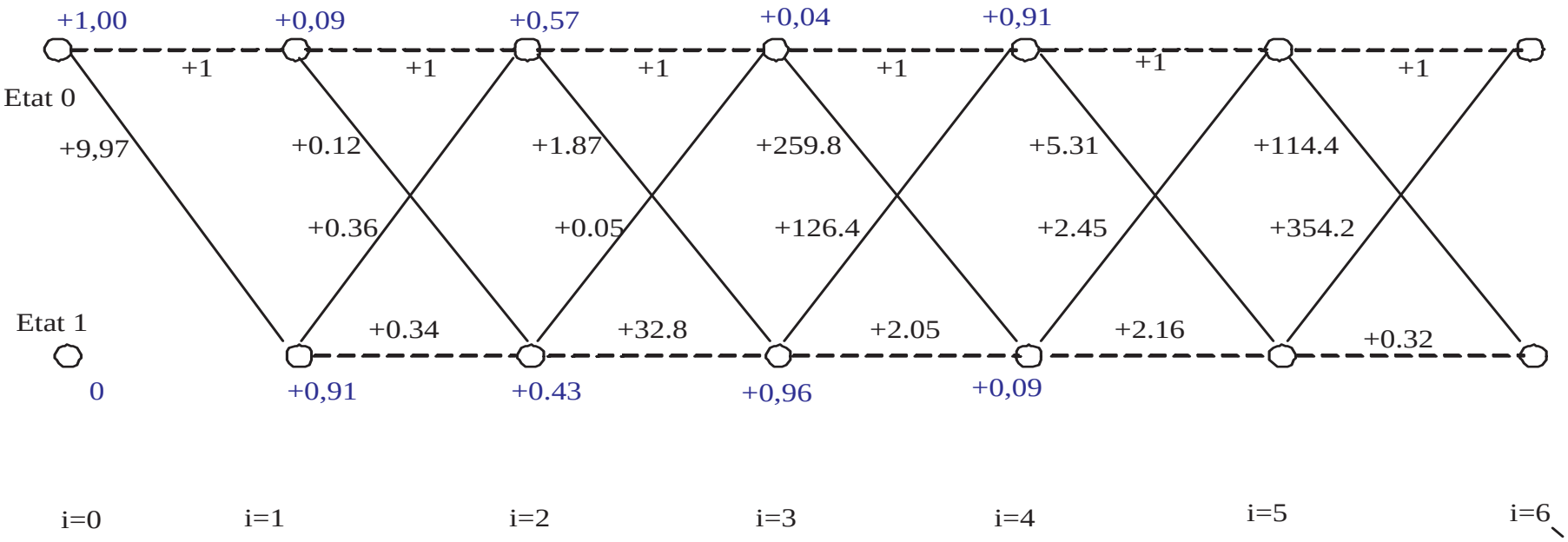
$\alpha_1(0) \propto \alpha_0(0)\gamma_0^{00} = 1$
$\alpha_1(1) \propto \alpha_0(0)\gamma_0^{11} = 9.97$
$\alpha_1(0) = 0.09$
$\alpha_1(1) = 0.91$

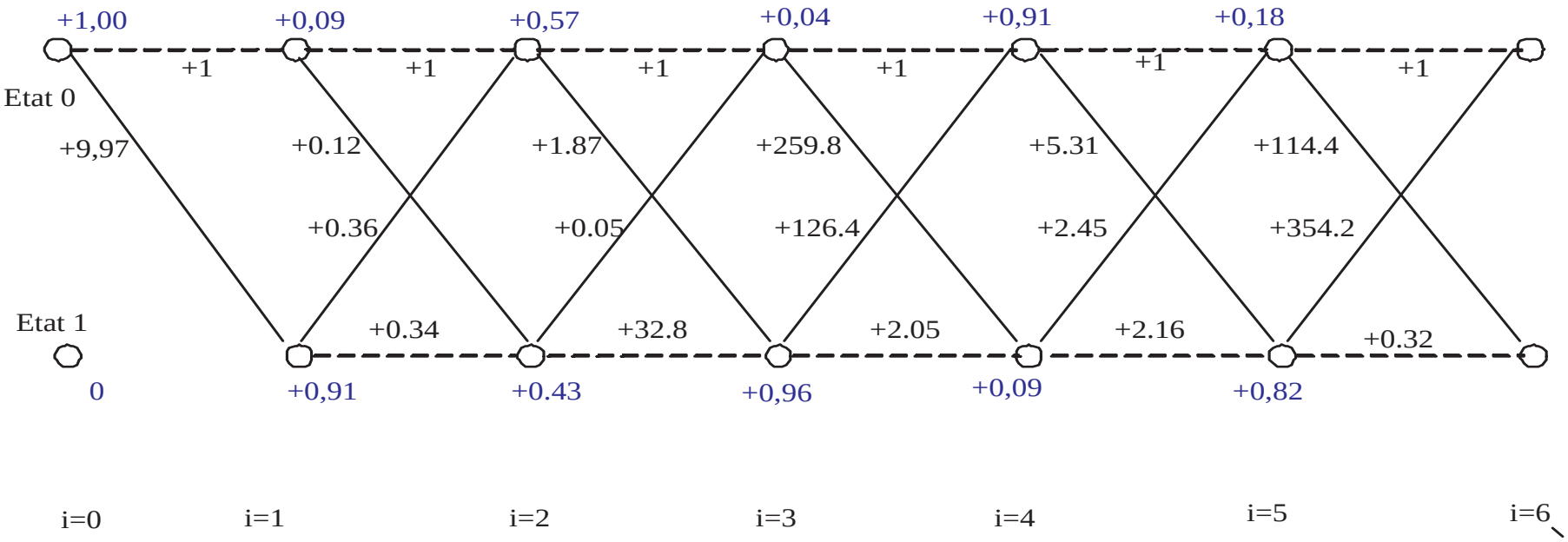


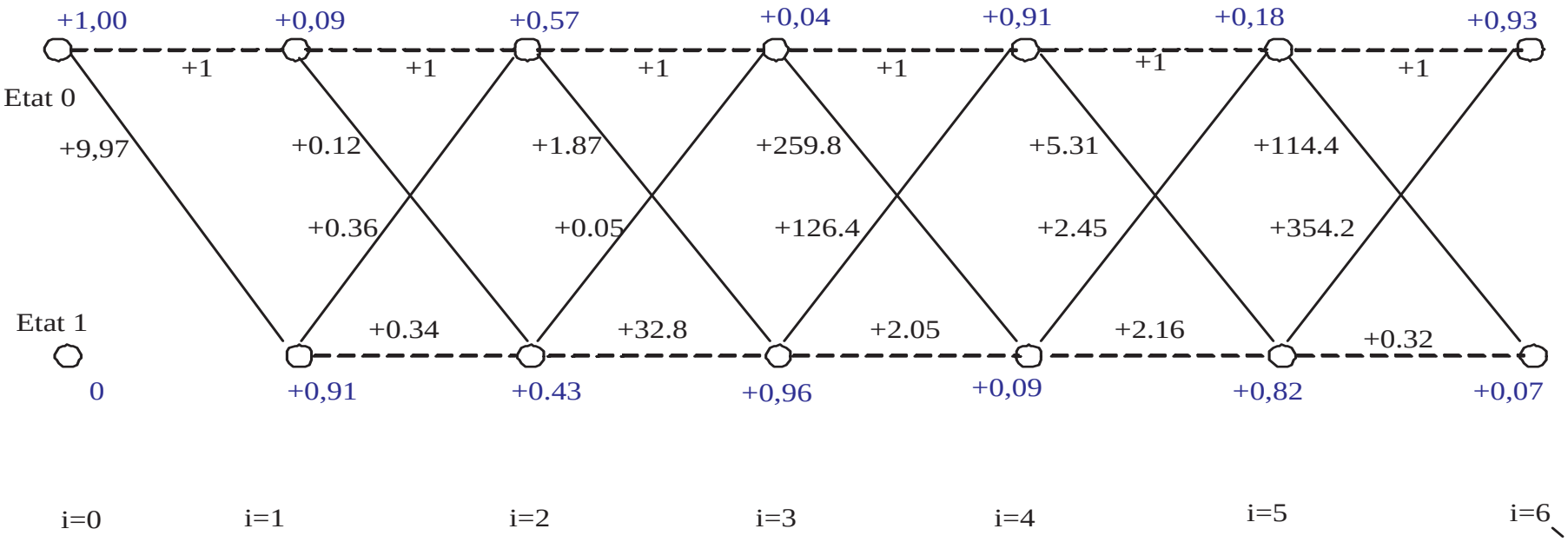
$i = 2$

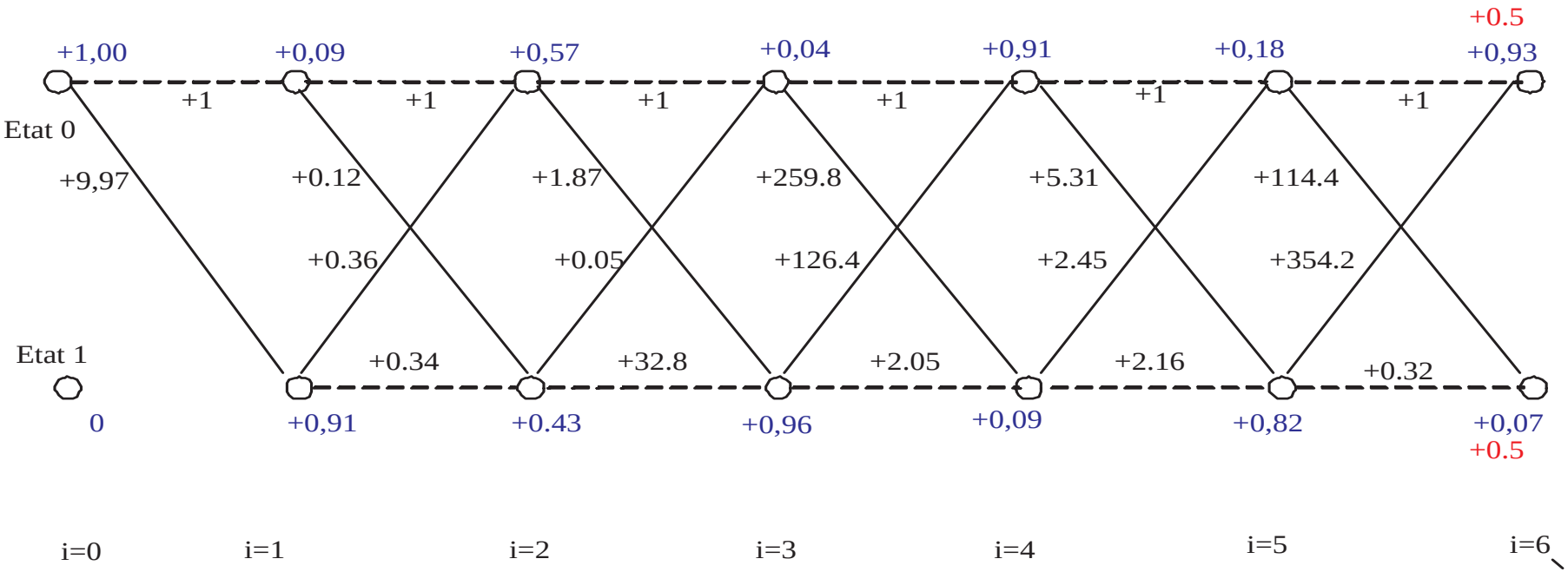
$$\alpha_2(0) \propto \alpha_1(0)\gamma_1^{00} + \alpha_1(1)\gamma_1^{10} = 0.417$$
$$\alpha_2(1) \propto \alpha_1(0)\gamma_1^{11} + \alpha_1(1)\gamma_1^{01} = 0.32$$
$$\alpha_2(0) = 0.57$$
$$\alpha_2(1) = 0.43$$


+1,00
+0,09
+0,57
+0,04
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+1,00
+0,09
+0,57
+0,04
+0,91
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
+0,09
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
+0,09
+0,82
i=0
i=1
i=2
i=3
i=4
i=5
i=6

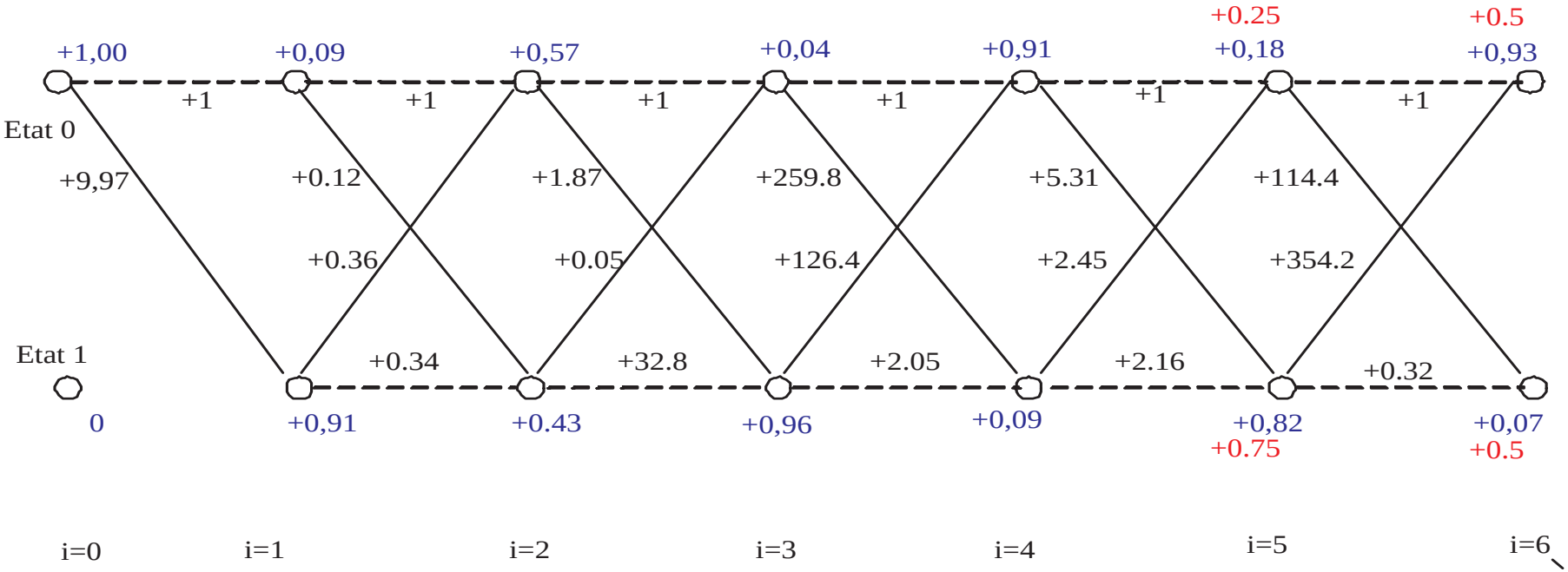

+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
+0,09
+0,82
+0,07
i=0
i=1
i=2
i=3
i=4
i=5
i=6



3) calcul des β

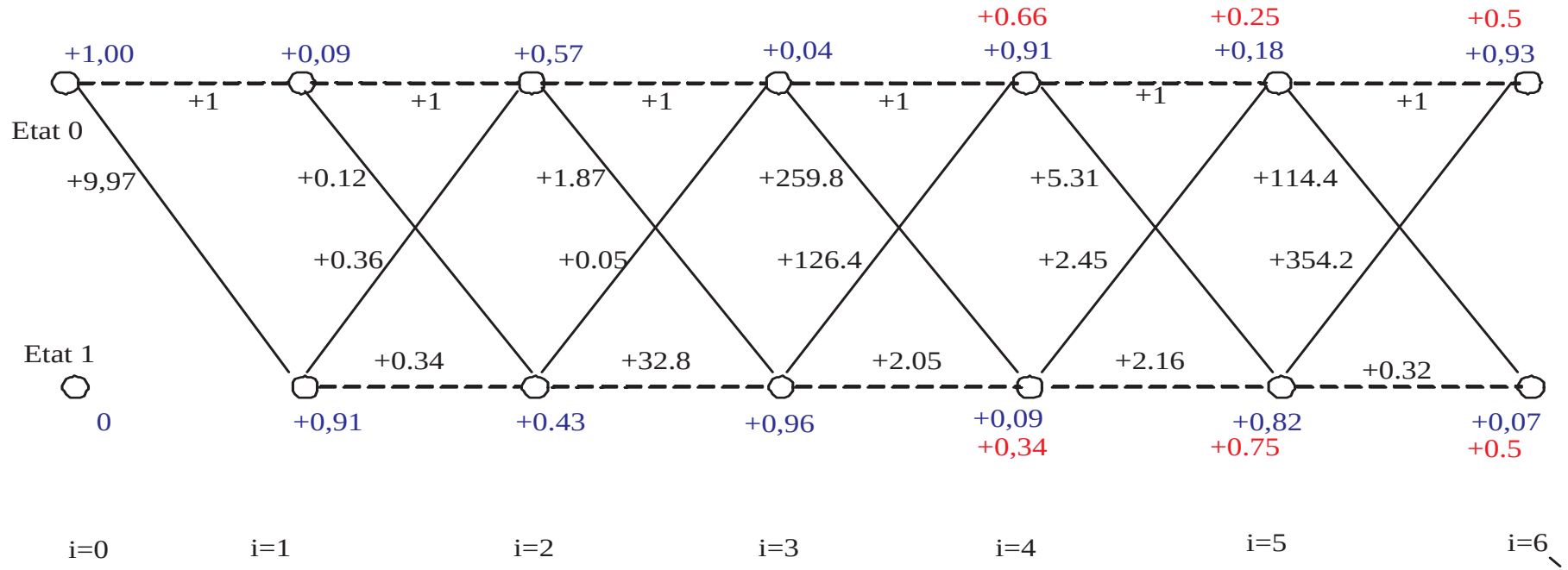
$i = 6$: initialisation

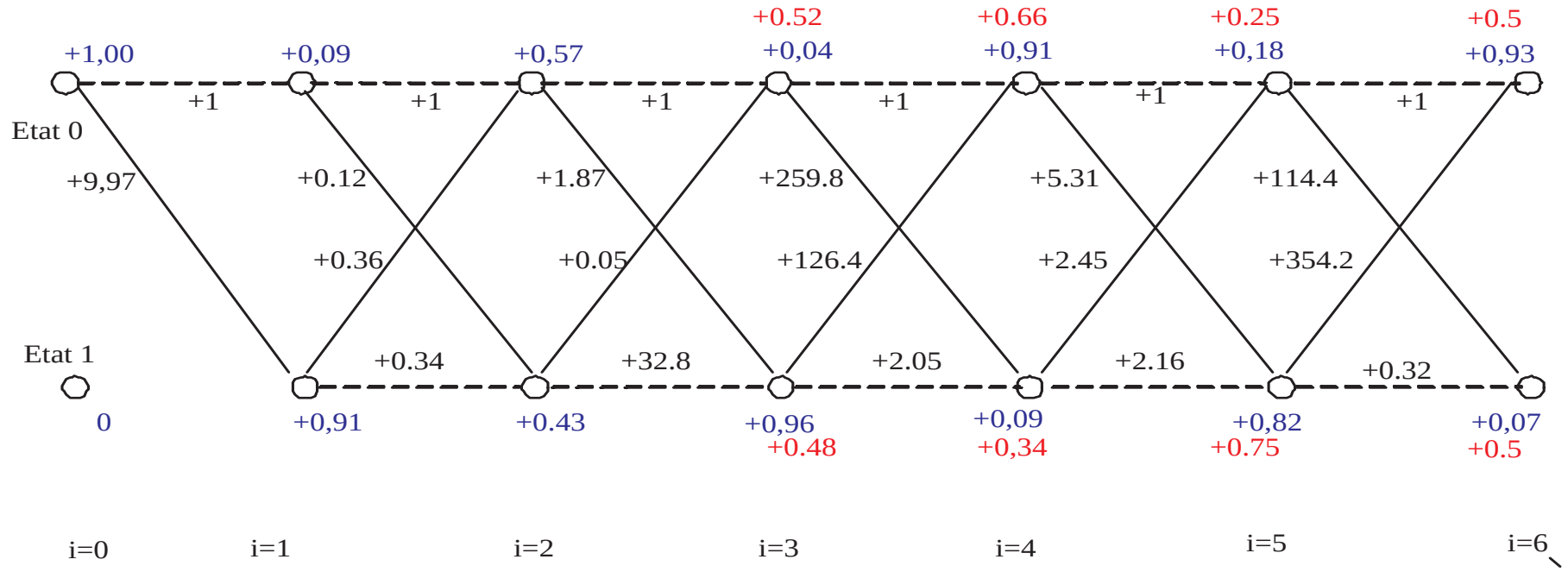
$\beta_6(0) = 0.5$
$\beta_6(1) = 0.5$ car le trellis n'a pas été terminé.

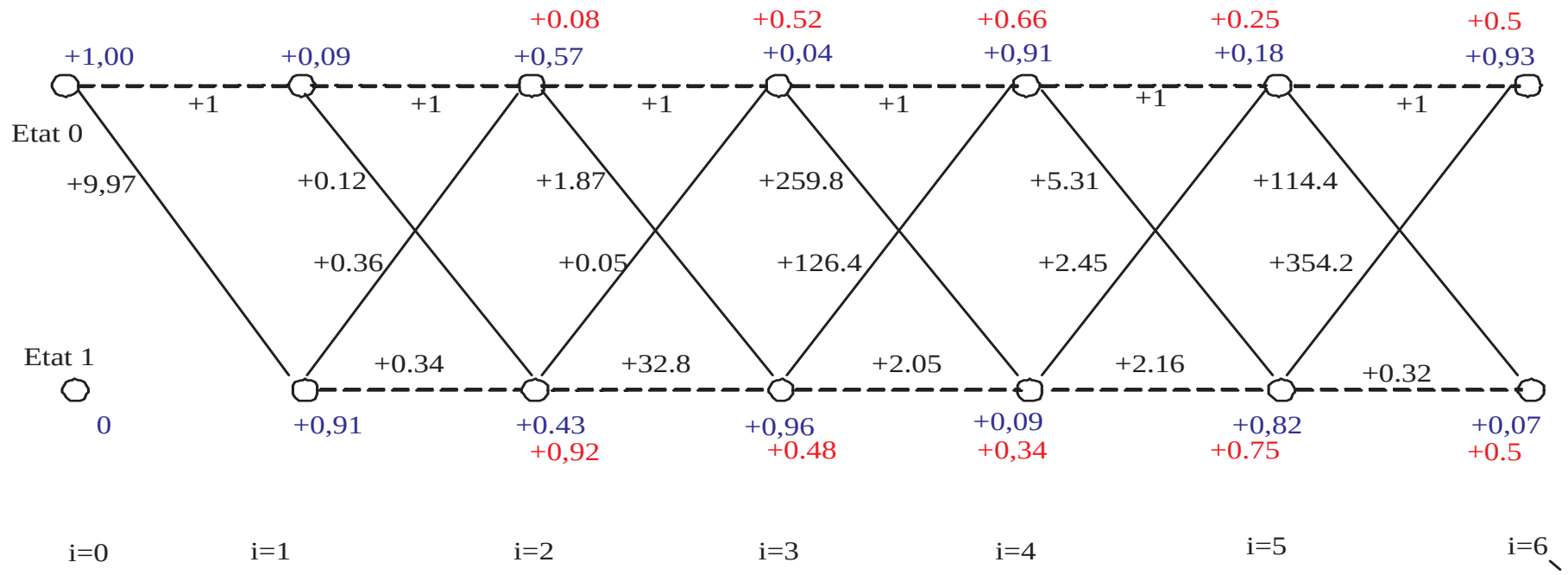


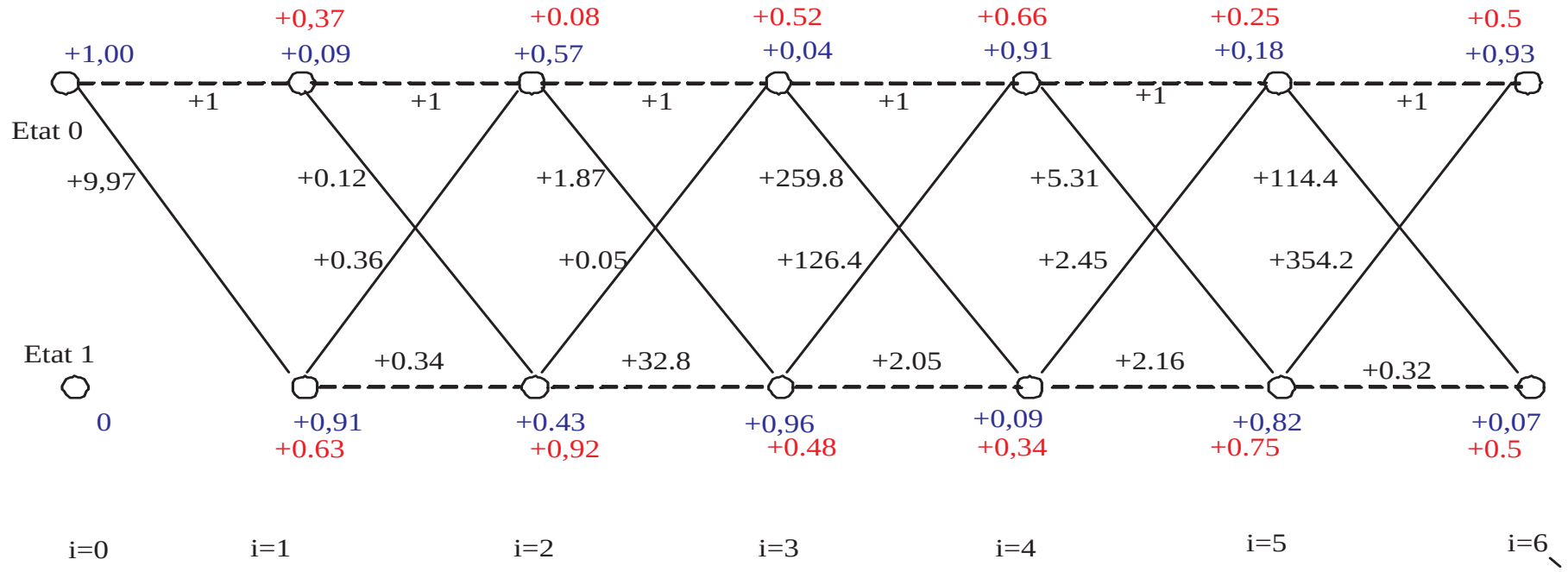
$i = 5$

$$\beta_5(0) \propto \beta_6(0)\gamma_5^{00} + \beta_6(1)\gamma_5^{11} = 57.71$$
$$\beta_5(1) \propto \beta_6(0)\gamma_5^{10} + \beta_6(1)\gamma_5^{01} = 177.28$$
$$\beta_5(0) = 0.25$$
$$\beta_5(1) = 0.75$$


+1,00
+0,09
+0,57
+0,04
+0.66
+0,91
+0.25
+0,18
+0.5
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
+0,09
+0,34
+0,82
+0.75
+0,07
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+1,00
+0,09
+0,57
+0.52
+0,04
+0.66
+0,91
+0.25
+0,18
+0.5
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
+0.48
+0,09
+0,34
+0,82
+0.75
+0,07
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6

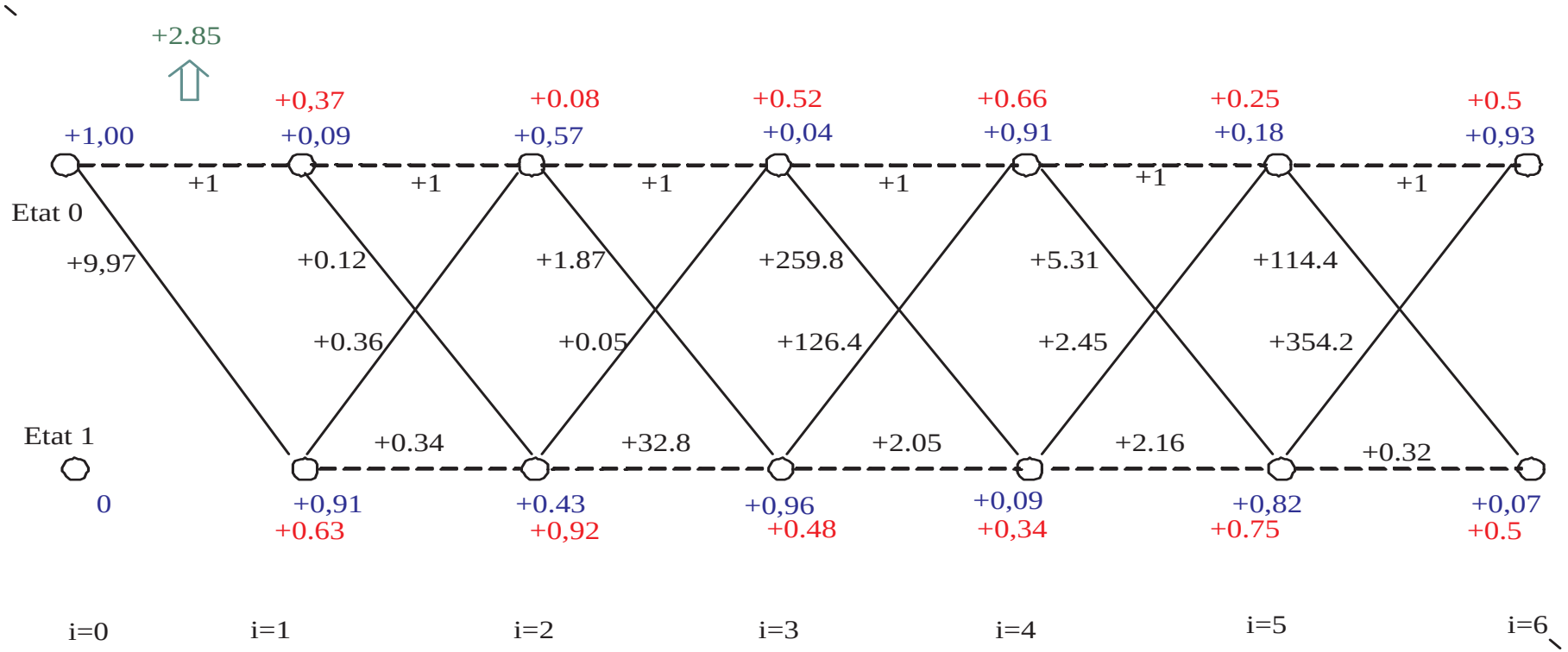

+0.08
+0.52
+0.66
+0.25
+0.5
+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
+0,09
+0,82
+0,07
+0,92
+0.48
+0,34
+0.75
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+0,37
+0.08
+0.52
+0.66
+0.25
+0.5
+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.43
+0,96
+0,09
+0,82
+0,07
+0.63
+0,92
+0.48
+0,34
+0.75
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6

4) calcul de $L_{APP}(x_i^S)$

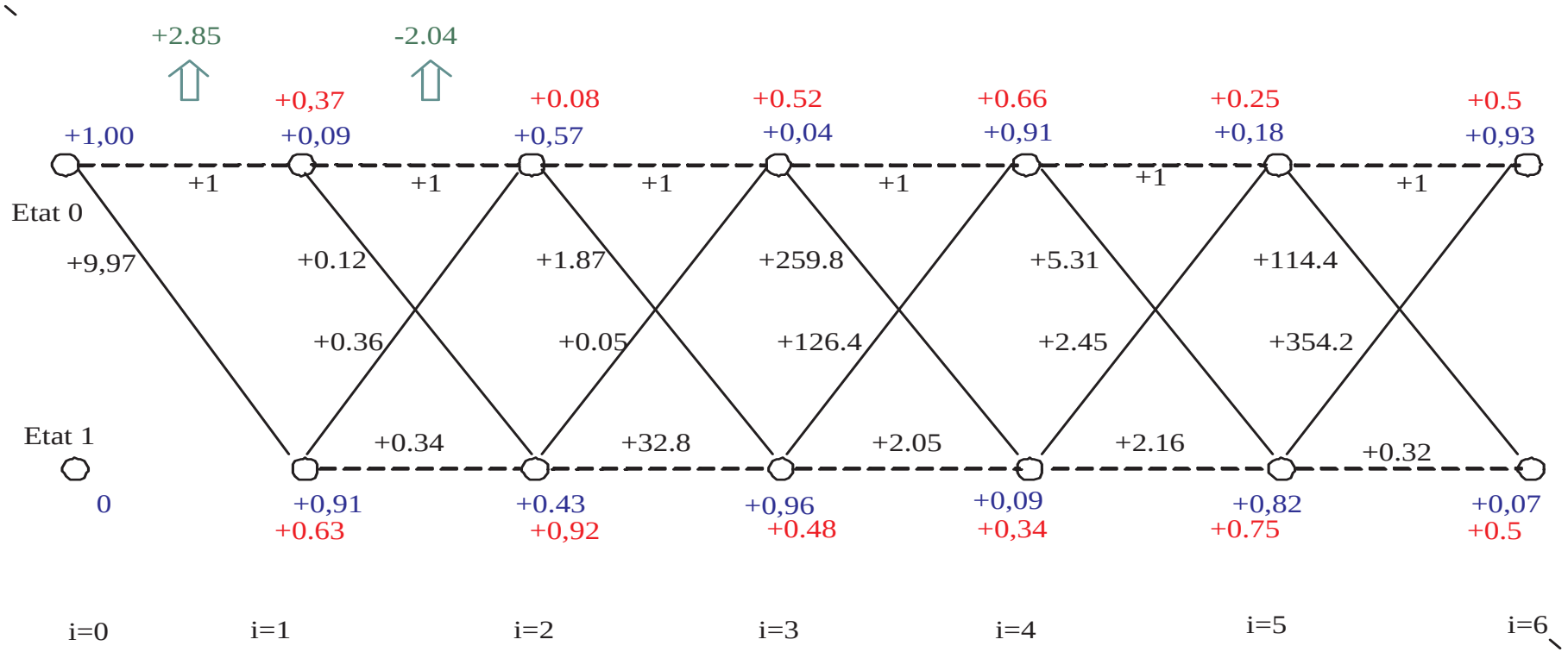
On utilise la relation

$$APP(x_i^S = a) \propto \sum_{m',m/x_i^S=a} \alpha_t(m') \times \gamma_t(m',m) \times \beta_{t+1}(m) \tag{27}$$



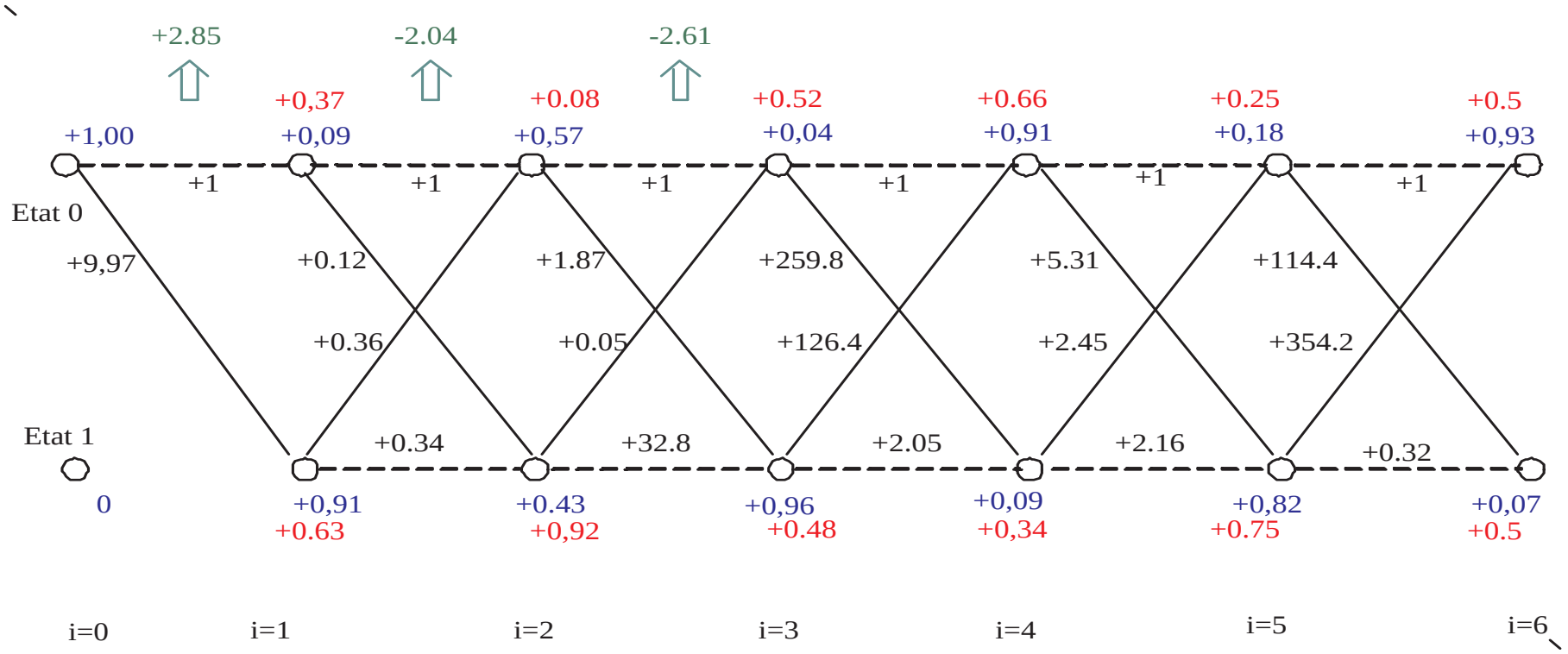
$$i = 0$$

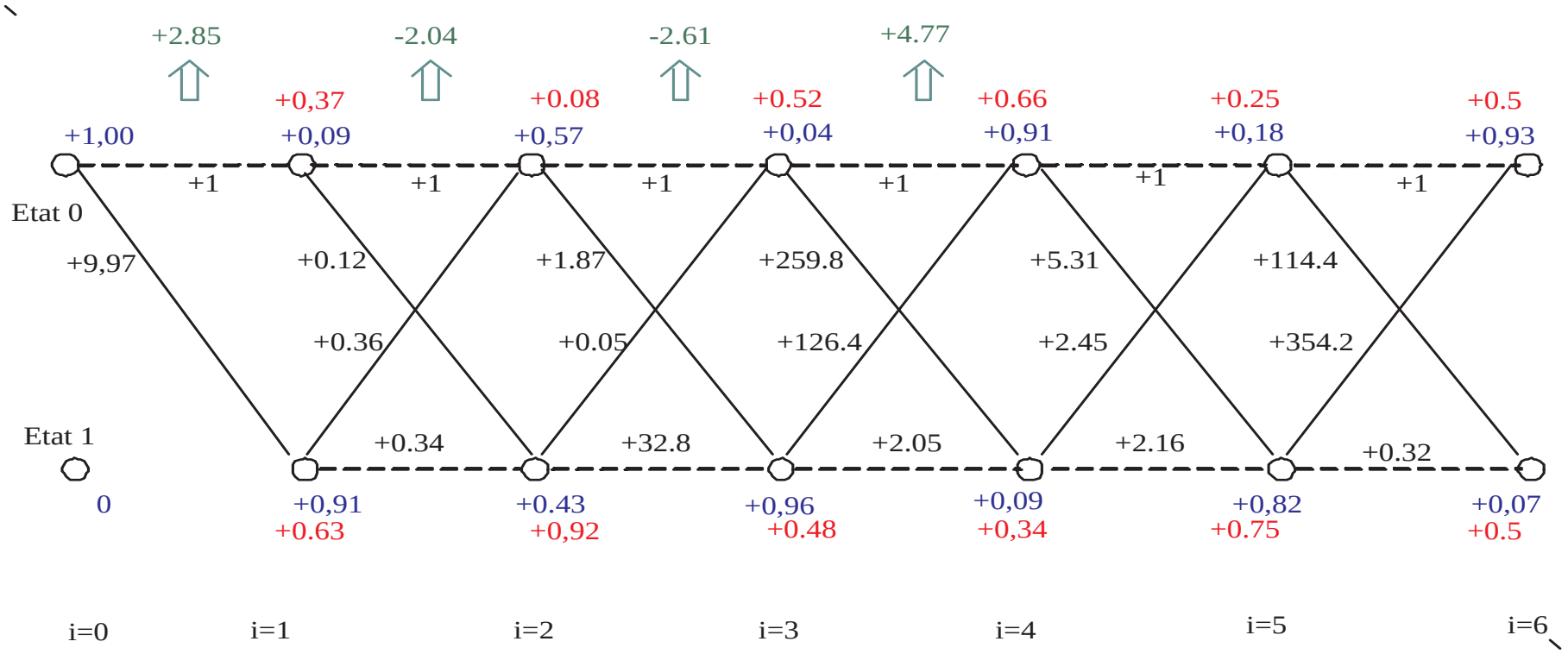
$$APP(x_0^S = -1) \propto \alpha_0(0) \times \gamma_0^{00} \times \beta_1(0) = 0.37$$
$$APP(x_0^S = +1) \propto \alpha_0(0) \times \gamma_0^{11} \times \beta_1(1) = 6.28$$
$$L_{APP}(x_0^S) = +2.85$$

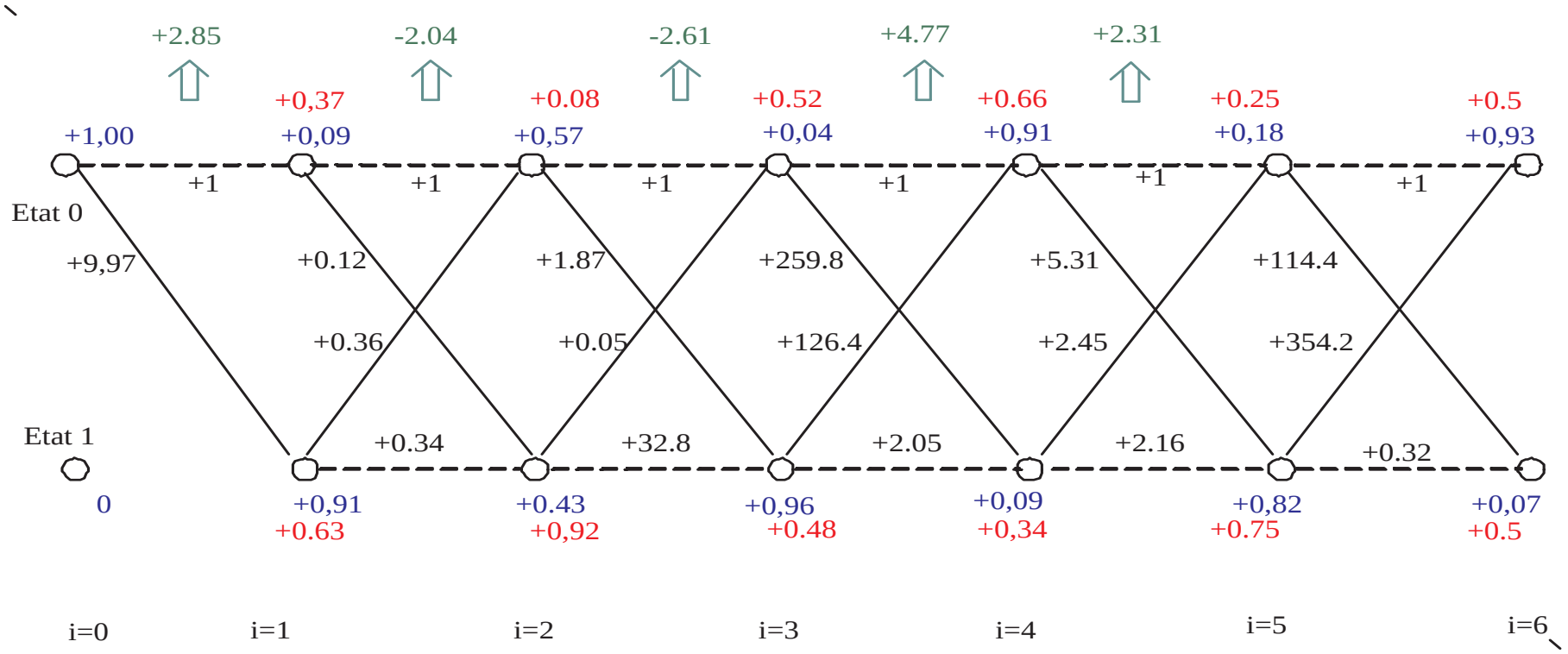


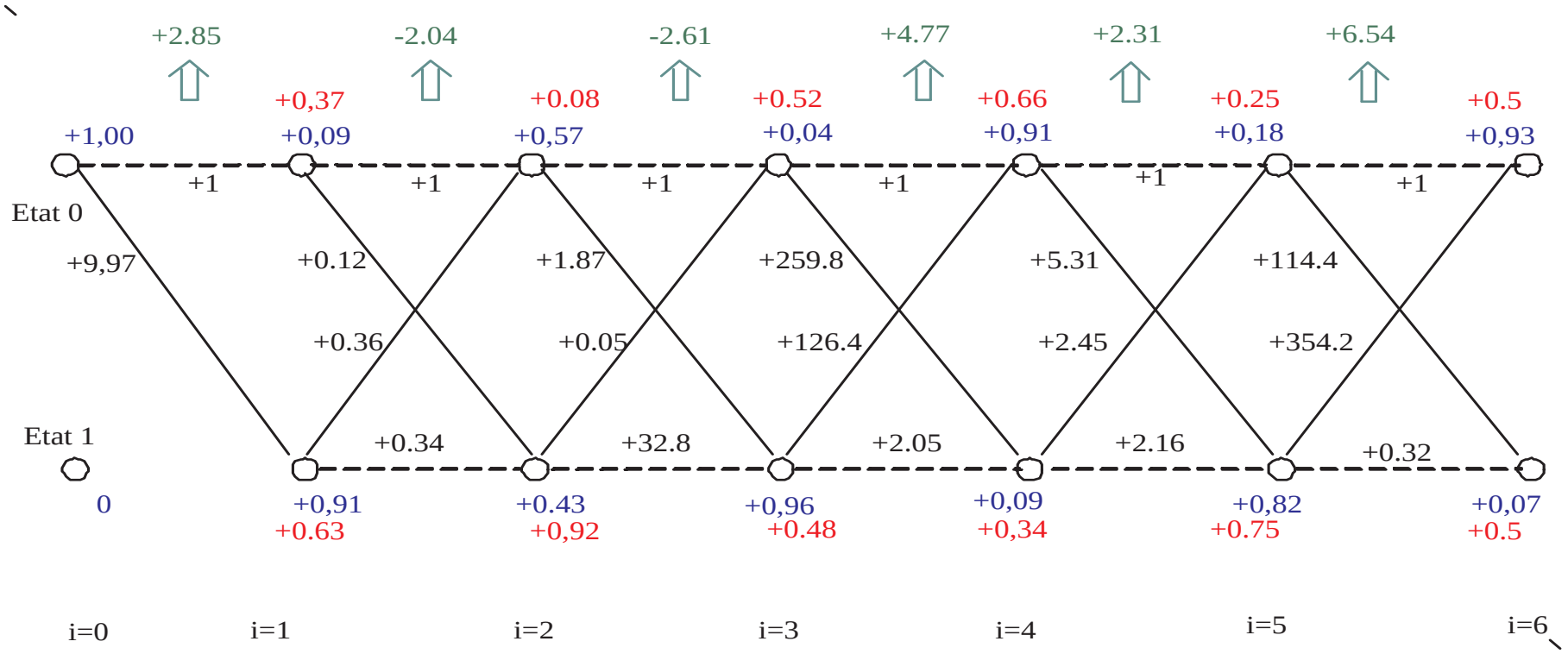
$$i = 1$$

$$APP(x_1^S = -1) \propto \alpha_1(0) \times \gamma_1^{00} \times \beta_2(0) + \alpha_1(1) \times \gamma_1^{01} \times \beta_2(1) = 0.2918$$
$$APP(x_1^S = +1) \propto \alpha_1(0) \times \gamma_1^{11} \times \beta_2(1) + \alpha_1(1) \times \gamma_1^{10} \times \beta_2(0) = 0.0361$$
$$L_{APP}(x_1^S) = -2.04$$


+2.85
-2.04
-2.61
+0,37
+0.08
+0.52
+0.66
+0.25
+0.5
+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.63
+0.43
+0,92
+0,96
+0.48
+0,09
+0,34
+0,82
+0.75
+0,07
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+2.85
-2.04
-2.61
+4.77
+0,37
+0.08
+0.52
+0.66
+0.25
+0.5
+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.63
+0.43
+0,92
+0,96
+0.48
+0,09
+0,34
+0,82
+0.75
+0,07
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+2.85
-2.04
-2.61
+4.77
+2.31
+0,37
+0.08
+0.52
+0.66
+0.25
+0.5
+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.63
+0.43
+0,92
+0,96
+0.48
+0,09
+0,34
+0,82
+0.75
+0,07
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+2.85
-2.04
-2.61
+4.77
+2.31
+6.54
+0,37
+0.08
+0.52
+0.66
+0.25
+0.5
+1,00
+0,09
+0,57
+0,04
+0,91
+0,18
+0,93
+1
+1
+1
+1
+1
+1
Etat 0
+9,97
+0.12
+1.87
+259.8
+5.31
+114.4
+0.36
+0.05
+126.4
+2.45
+354.2
Etat 1
+0.34
+32.8
+2.05
+2.16
+0.32
0
+0,91
+0.63
+0.43
+0,92
+0,96
+0.48
+0,09
+0,34
+0,82
+0.75
+0,07
+0.5
i=0
i=1
i=2
i=3
i=4
i=5
i=6

Finalement après calcul, on obtient les informations a posteriori suivantes :

$L_{APP}(x_i^S)$	+2.85	-2.04	-2.61	+4.77	+2.31	+6.54

EXEMPLE DECODEUR MAX LOG MAP

En pratique, l'utilisation de logarithme de probabilité permet de simplifier les calculs.

1) calcul des $\ln \gamma$

Les métriques de branches $\ln \gamma_i^{pq}$ se calculent directement à partir des informations L_{INT}

$\ln \gamma_i^{00} = 0$

$\ln \gamma_i^{10} = L_{INT}(x_i^S)$

$\ln \gamma_i^{01} = L_{INT}(x_i^P)$

$\ln \gamma_i^{11} = L_{INT}(x_i^S) + L_{INT}(x_i^P)$

On obtient :

$\ln \gamma_i^{00}$	0	0	0	0	0	0
$\ln \gamma_i^{10}$	+1.21	-1.00	-2.86	+4.84	+0.90	+5.87
$\ln \gamma_i^{01}$	+1.09	-1.07	+3.49	+0.72	+0.77	-1.13
$\ln \gamma_i^{11}$	+2.30	-2.07	+0.63	+5.56	+1.67	+4.74

De la même façon, au lieu de calculer les α et β nous calculerons $\ln \alpha$ et des $\ln \beta$

DECODEUR MAX LOG MAP

• calcul exact et approximatif de $\ln \alpha$ et $\ln \beta$:

calcul exact :

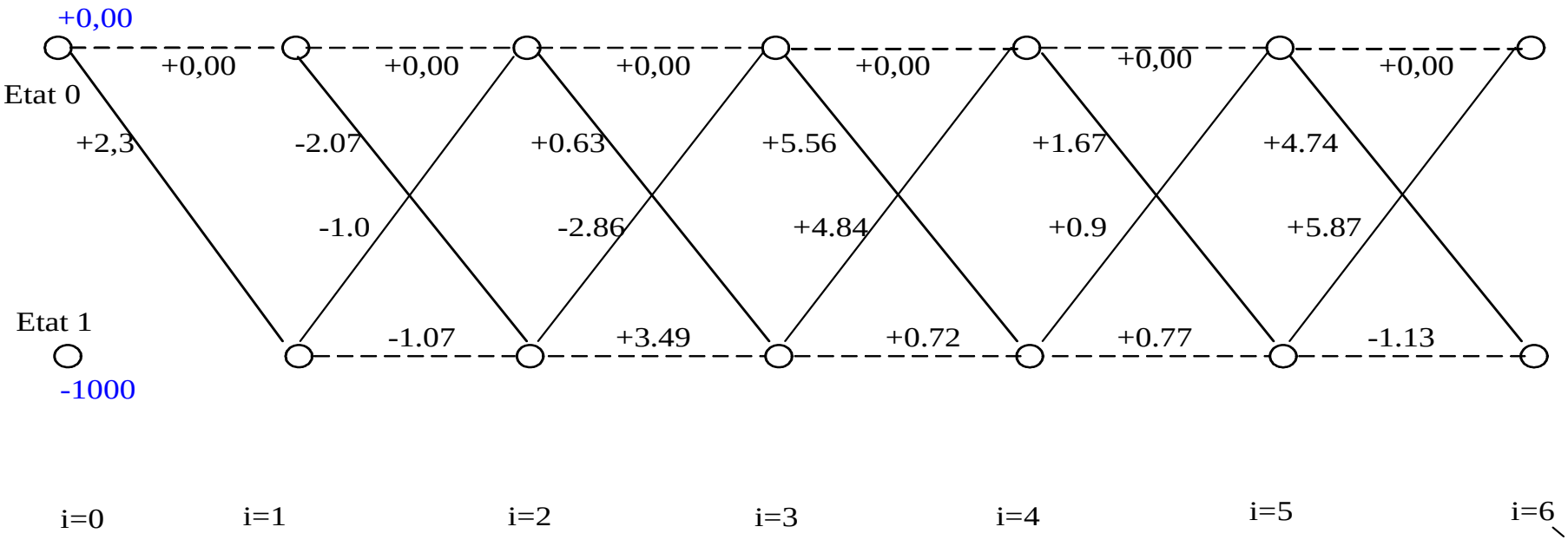
$$\alpha = \alpha_1 + \alpha_2$$

$$\begin{aligned} \ln \alpha &= \ln(\alpha_1 + \alpha_2) \\ &= \ln(\exp(\ln \alpha_1) + \exp(\ln \alpha_2)) \\ &= \ln \alpha_1 (1 + \exp(\ln \alpha_2 - \ln \alpha_1)) \end{aligned}$$

approximation MAX LOG MAP :

$$\ln \alpha \approx \max(\ln \alpha_1, \ln \alpha_2)$$

• même courbes de contour que l'algorithme somme produit

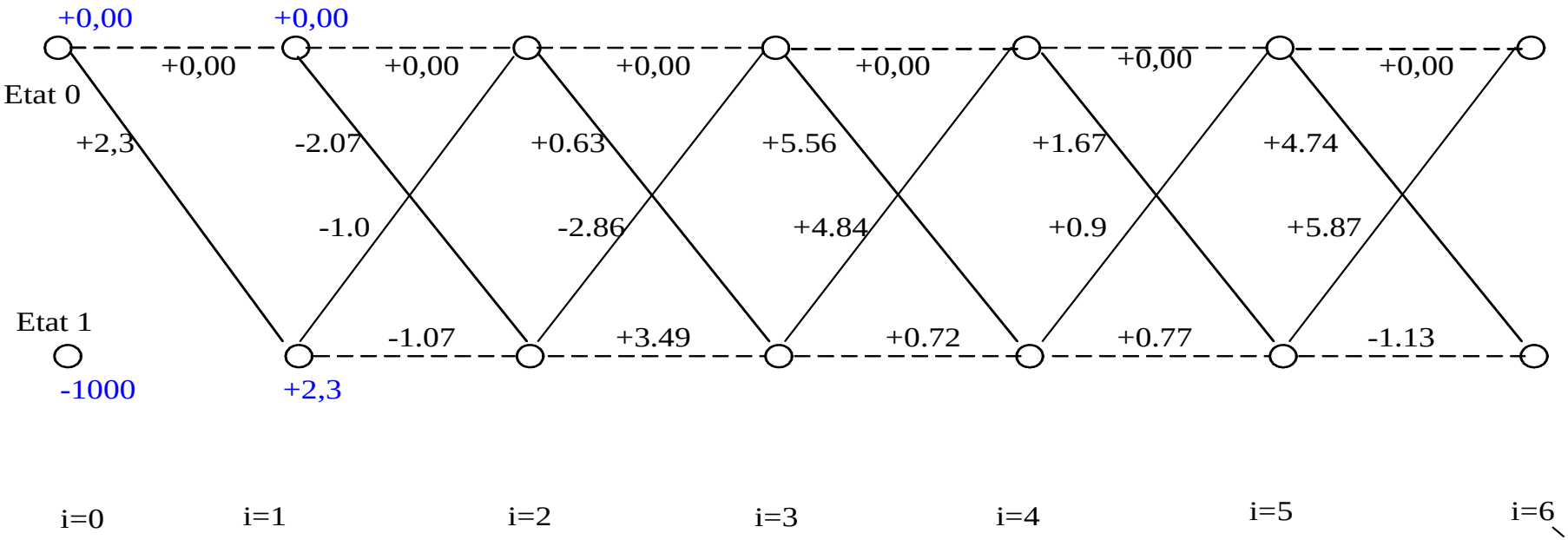


2) calcul des $\ln \alpha$

$i = 0$: initialisation

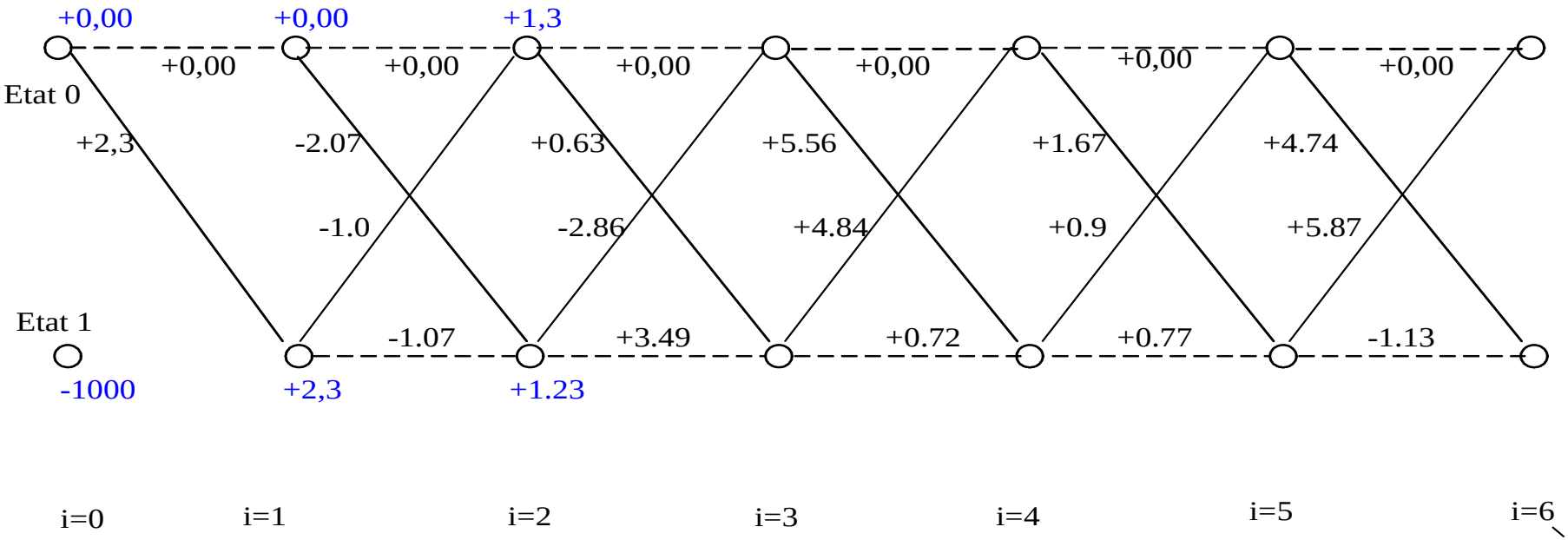
$\ln \alpha_0(0) = 0$

$\ln \alpha_0(1) = -\infty$



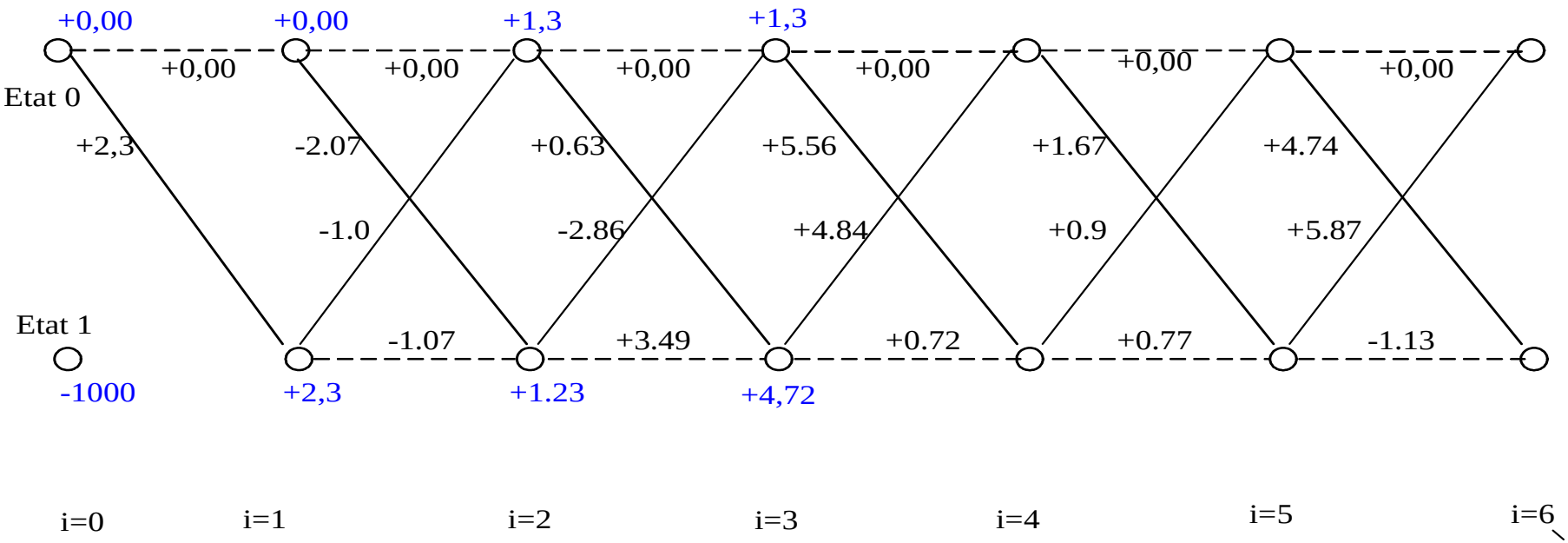
$$i = 1$$

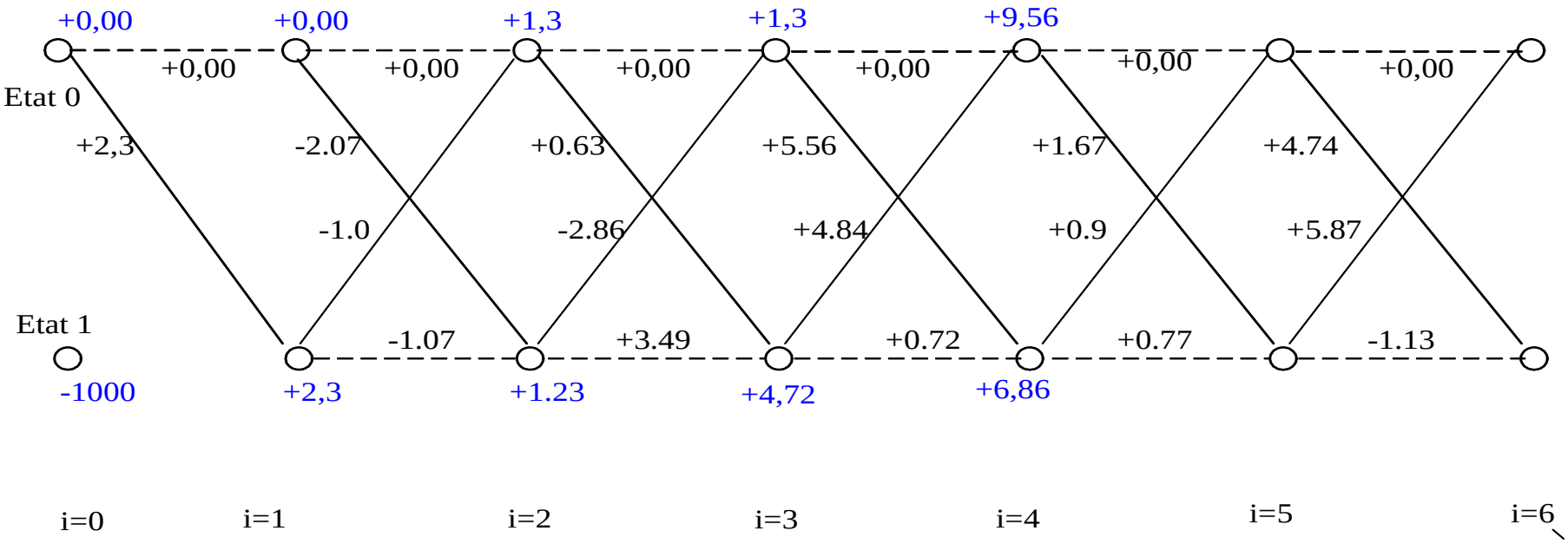
$$\ln \alpha_1(0) = \ln \alpha_0(0) + \ln \gamma_0^{00} = 0$$
$$\ln \alpha_1(1) = \ln \alpha_0(0) + \ln \gamma_0^{11} = 2.3$$

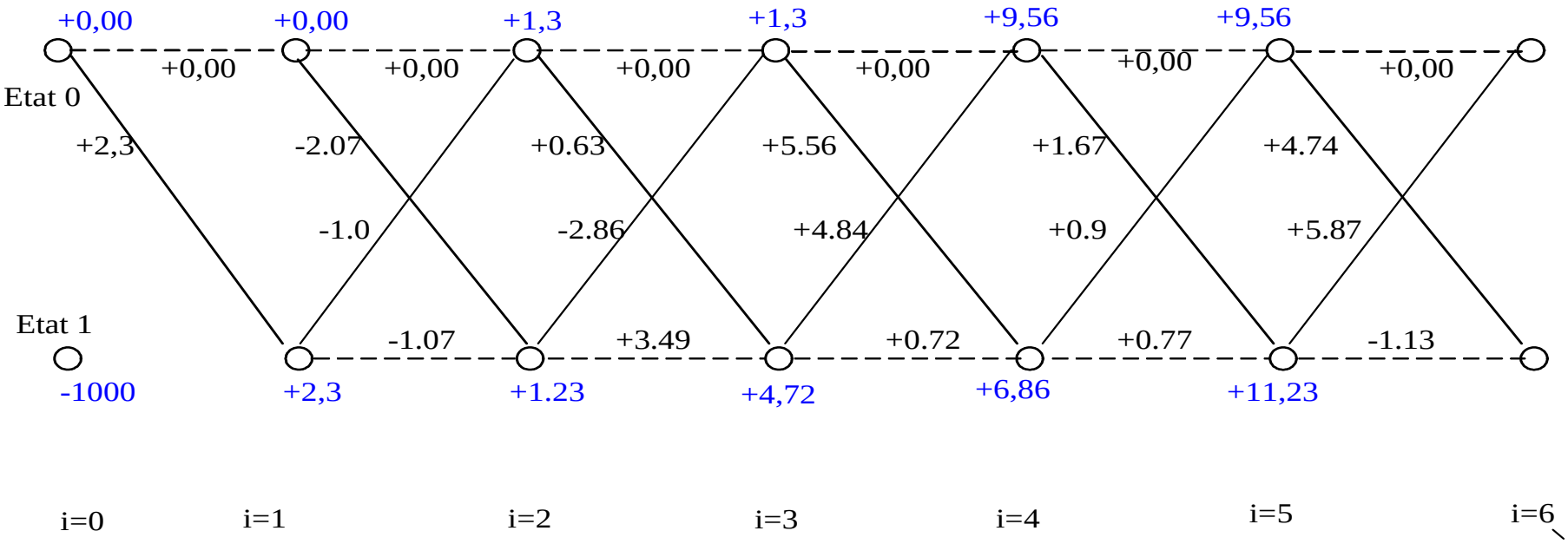


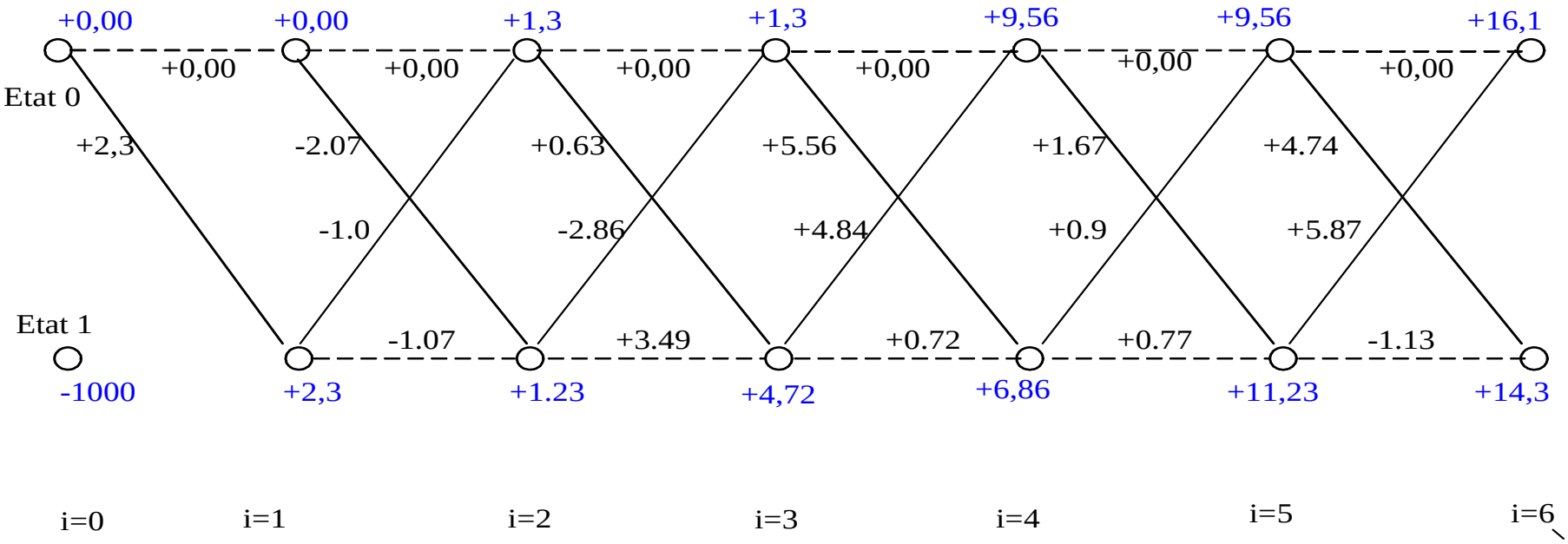
$i = 2$

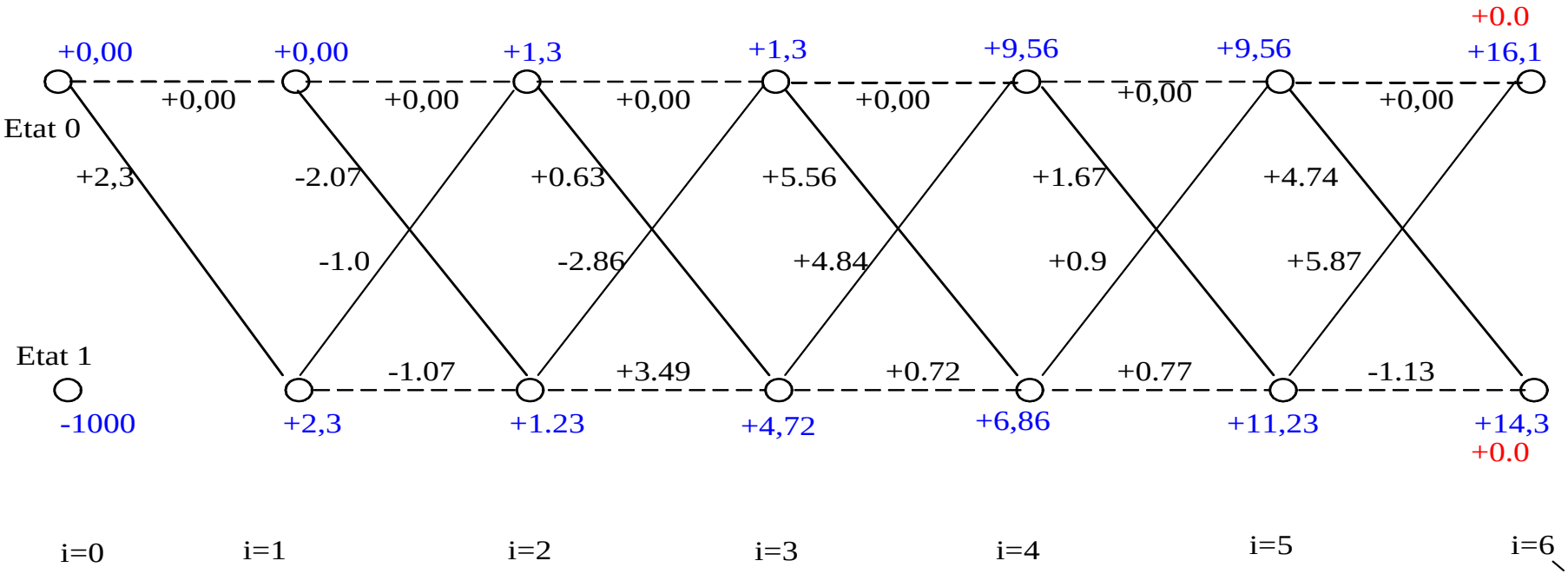
$$\ln \alpha_2(0) = \max(\ln \alpha_1(0) + \ln \gamma_1^{00}; \ln \alpha_1(1) + \ln \gamma_1^{10}) = \max(0; 1.3) = 1.3$$
$$\ln \alpha_2(1) = \max(\ln \alpha_1(0) + \ln \gamma_1^{11}; \ln \alpha_1(1) + \ln \gamma_1^{01}) = \max(-2.07; 1.23) = 1.23$$


+0,00
+0,00
+1,3
+1,3
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+0,00
+0,00
+1,3
+1,3
+9,56
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
+6,86
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
+6,86
+11,23
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
+6,86
+11,23
+14,3
i=0
i=1
i=2
i=3
i=4
i=5
i=6

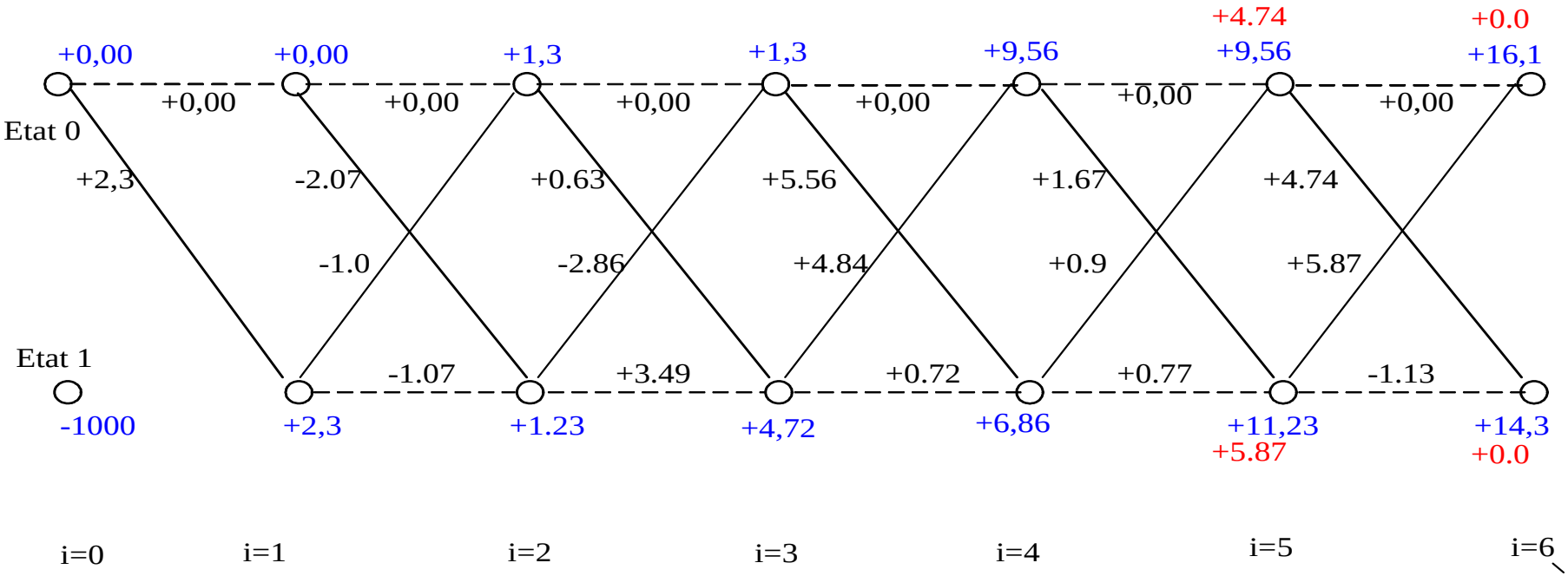


3) calcul des $\ln \beta$

$i = 6$: initialisation

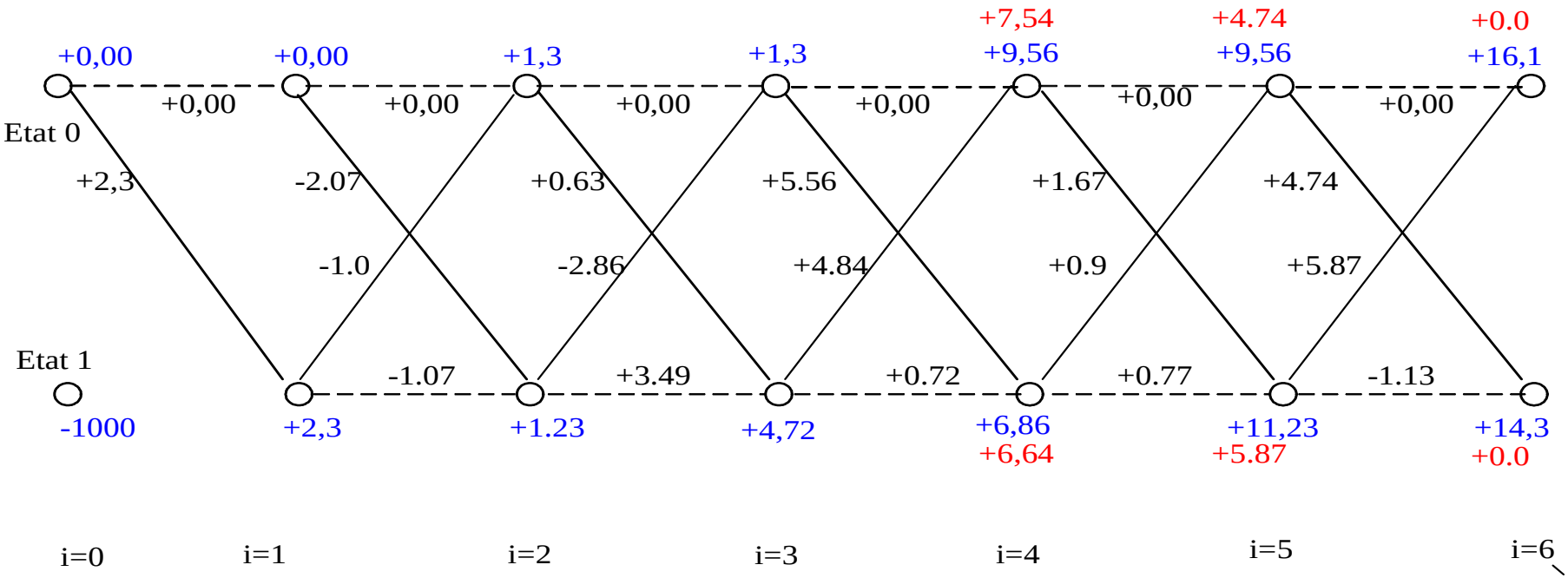
$\ln \beta_6(0) = 0$

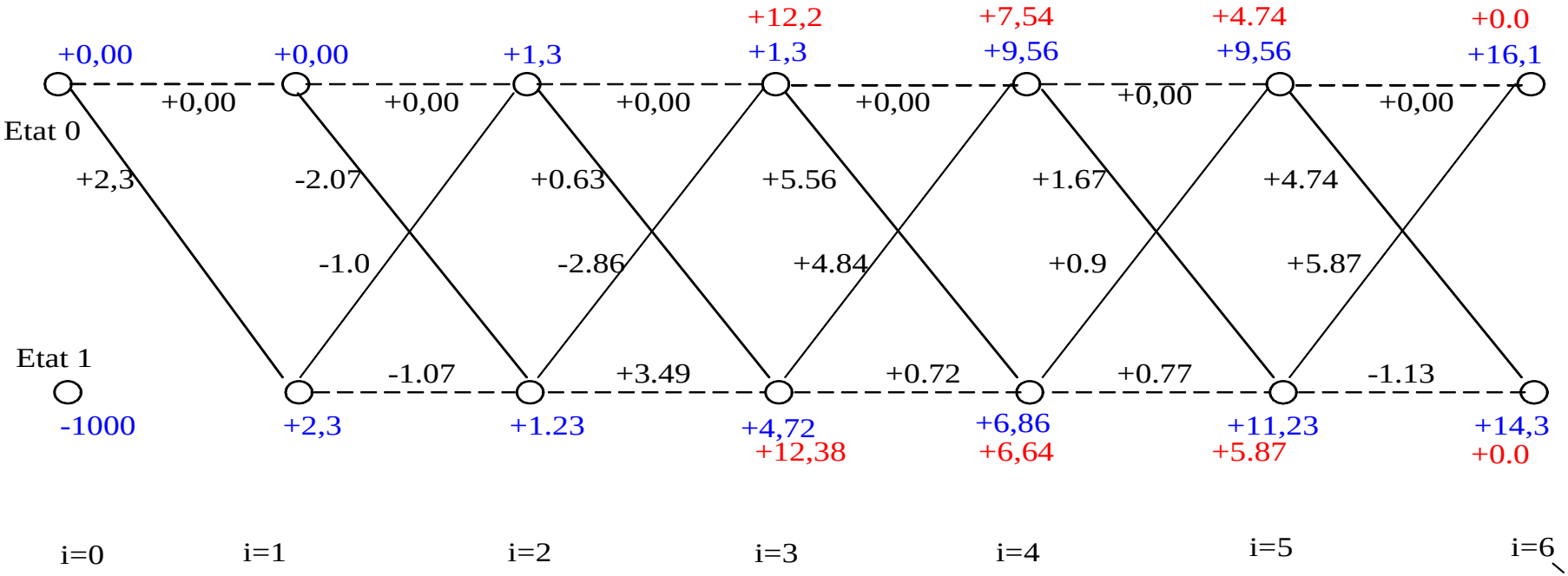
$\ln \beta_6(1) = 0$ car le trellis n'a pas été terminé.

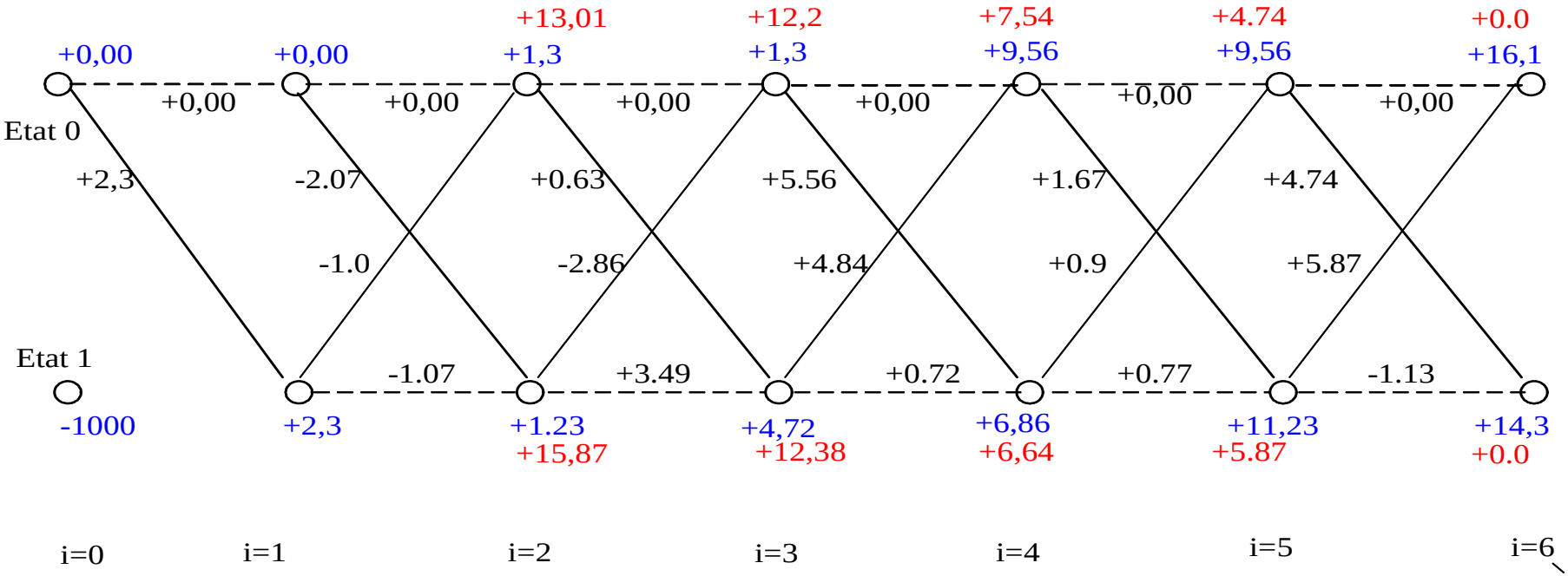


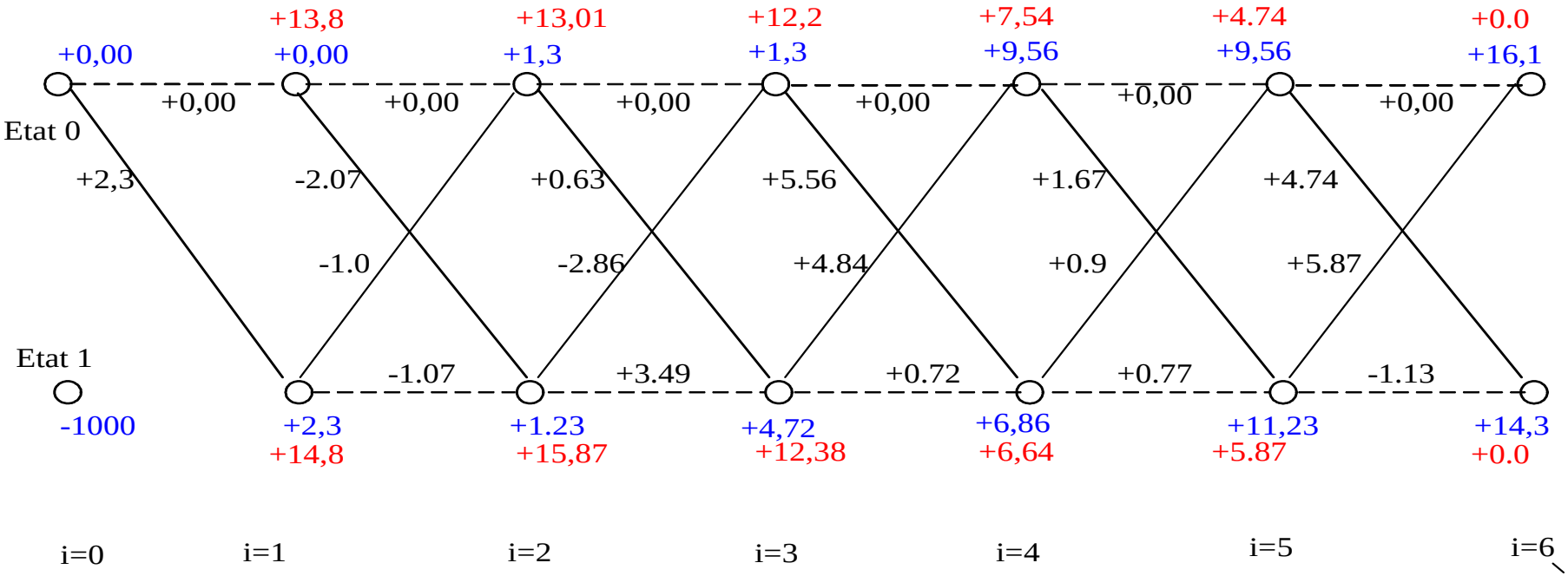
$$i = 5$$

$$\ln \beta_5(0) = \max(\ln \beta_6(0) + \ln \gamma_5^{00}; \ln \beta_6(1) + \ln \gamma_5^{11} = \max(0; +4.74) = +4.74$$
$$\ln \beta_5(1) = \max(\ln \beta_6(0) + \ln \gamma_5^{10}; \ln \beta_6(1) + \ln \gamma_5^{01}) = \max(5.87; -1.13) = +5.87$$


+0,00
+0,00
+1,3
+1,3
+7,54
+9,56
+4.74
+9,56
+0.0
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
+6,86
+6,64
+11,23
+5.87
+14,3
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+0,00
+0,00
+1,3
+12,2
+1,3
+7,54
+9,56
+4.74
+9,56
+0.0
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
+12,38
+6,86
+6,64
+11,23
+5.87
+14,3
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6

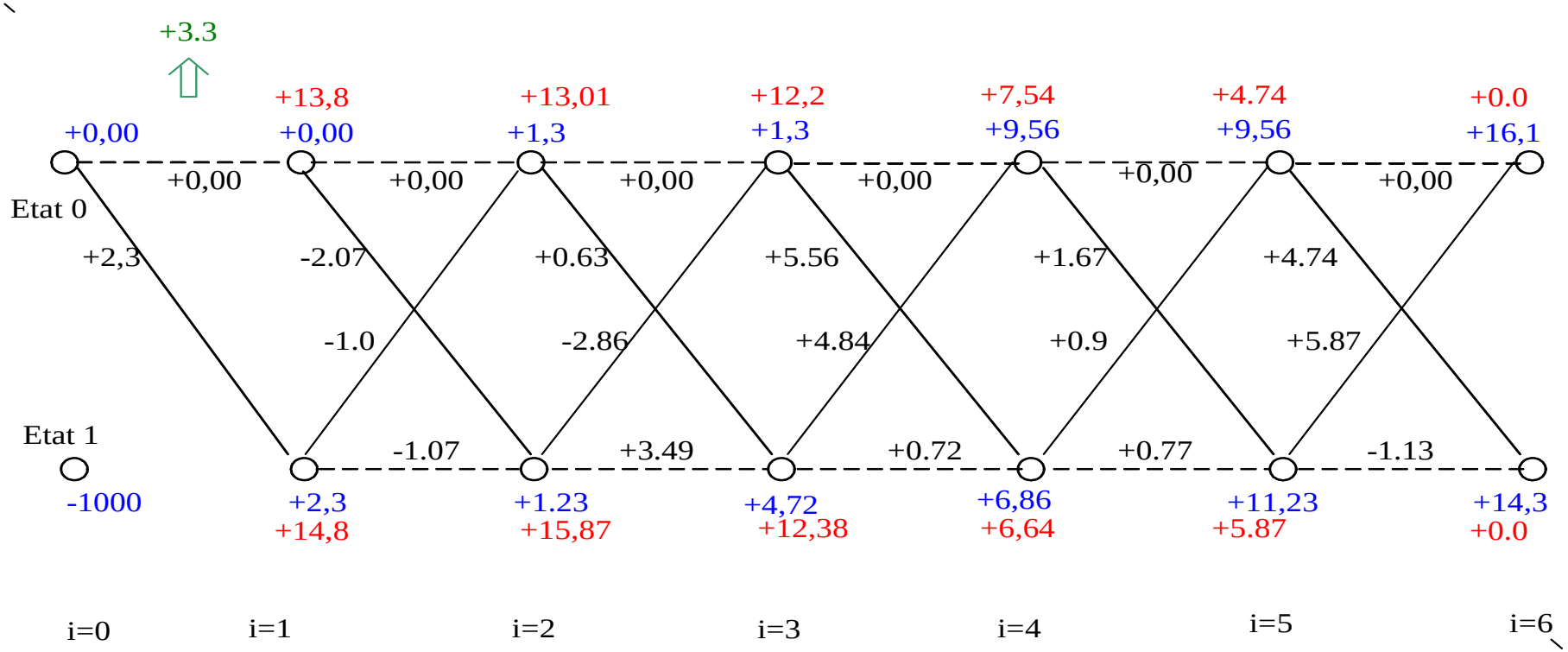

+13,01
+12,2
+7,54
+4.74
+0.0
+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
+6,86
+11,23
+14,3
+15,87
+12,38
+6,64
+5.87
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+13,8
+13,01
+12,2
+7,54
+4.74
+0.0
+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+1.23
+4,72
+6,86
+11,23
+14,3
+14,8
+15,87
+12,38
+6,64
+5.87
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6

4) calcul de $L_{APP}(x_i^S)$

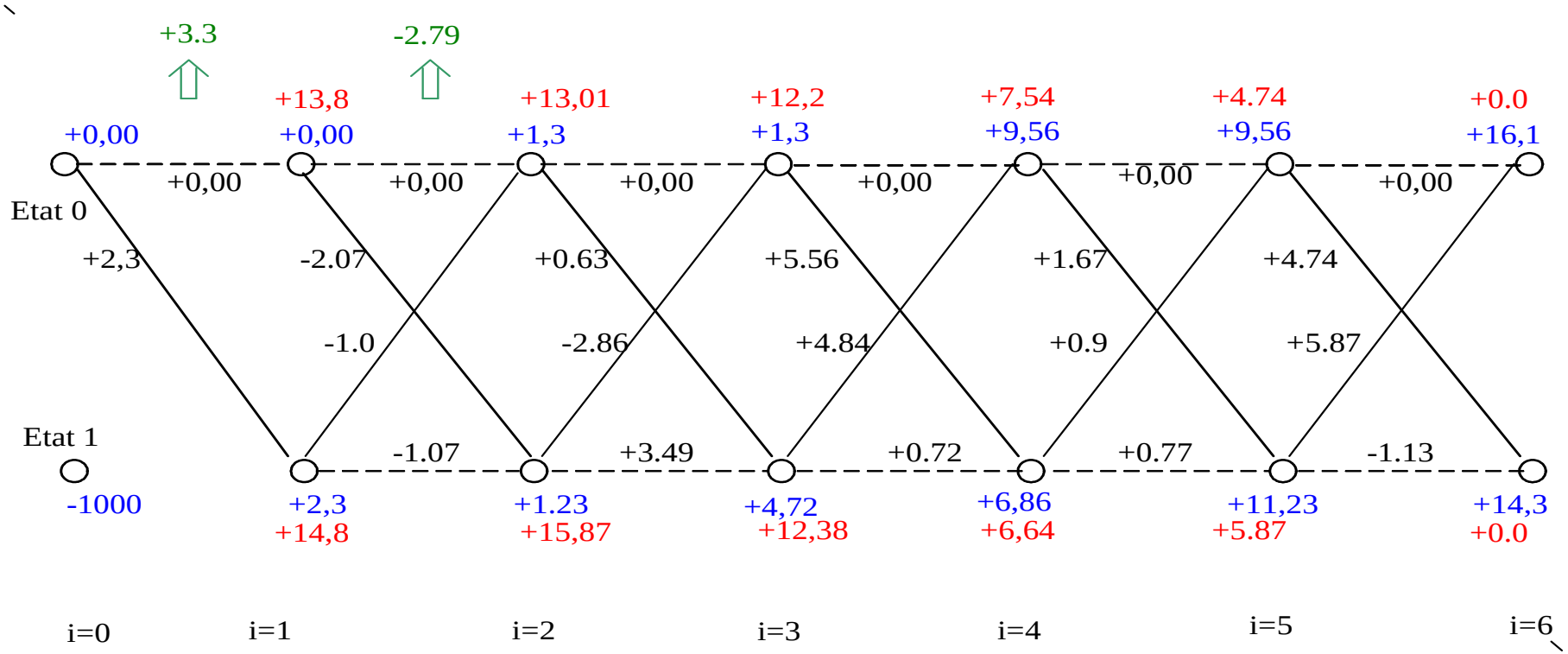
On utilise la relation

$$L_{APP}(x_i^S) = \max_{m',m/x_i^S=+1} \left(\ln \alpha_t(m') + \ln \gamma_t(m',m) + \ln \beta_{t+1}(m) \right) - \max_{m',m/x_i^S=-1} \left(\ln \alpha_t(m') + \ln \gamma_t(m',m) + \ln \beta_{t+1}(m) \right) \quad (28)$$



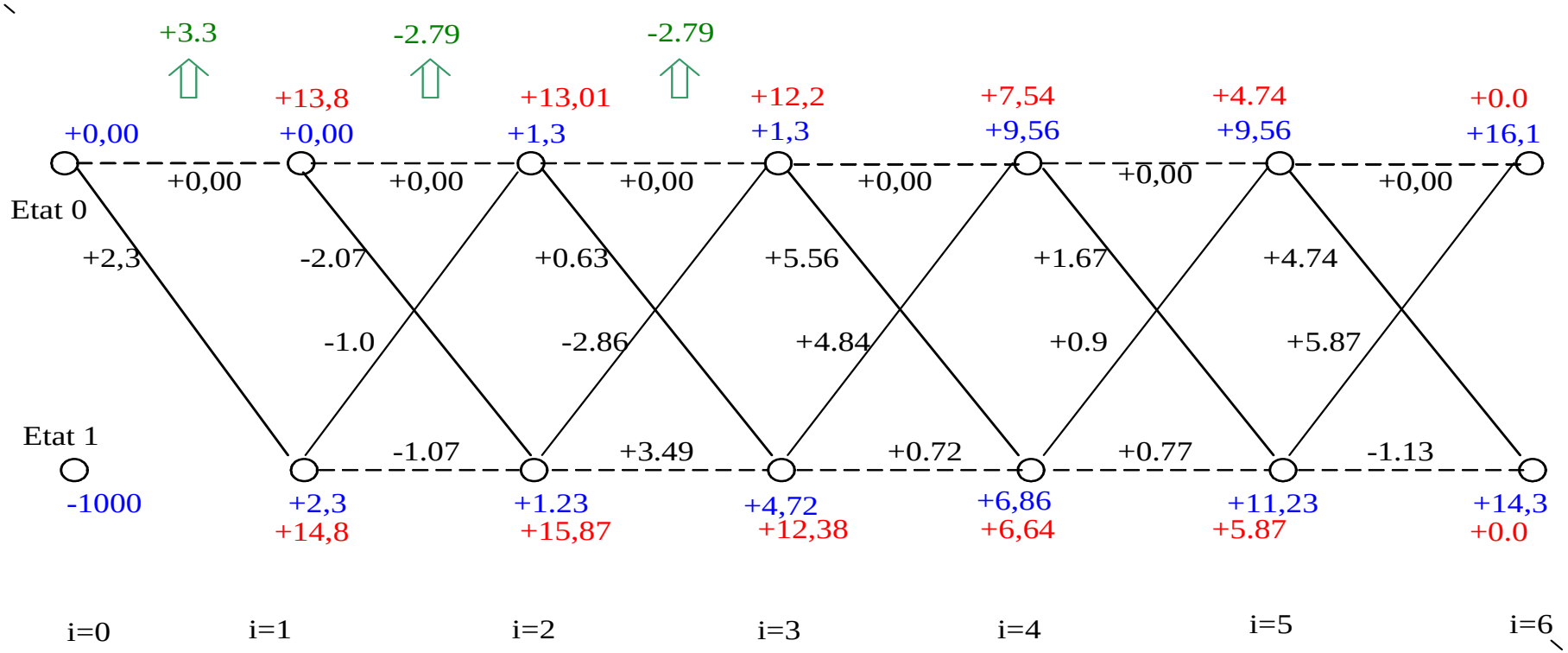
$i = 0$

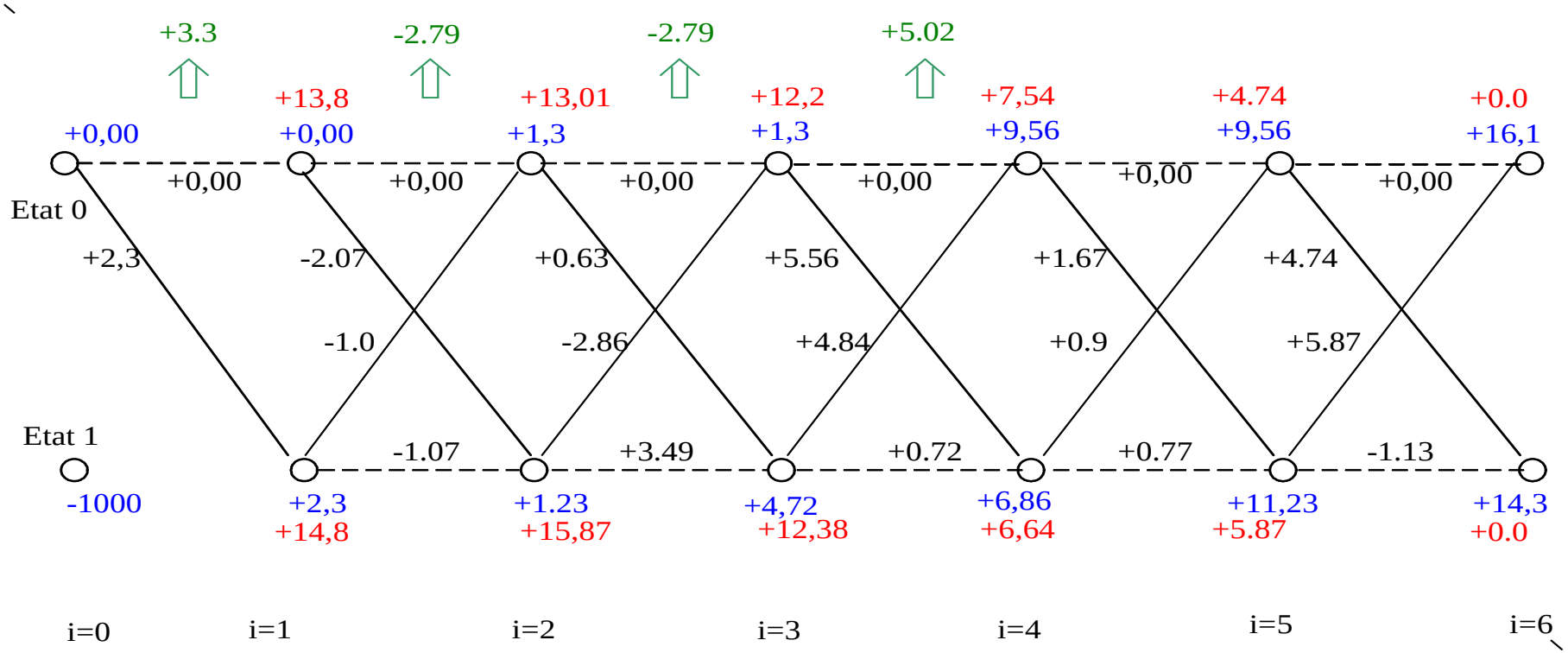
$$
\begin{aligned}
L_{APP}(x_0^S) =& (\ln \alpha_0(0) + \ln \gamma_0^{11} + \ln \beta_1(1)) \\
& - (\ln \alpha_0(0) + \ln \gamma_0^{00} + \ln \beta_1(0)) \\
& = (0 + 2.3 + 14.8) - (0 + 0 + 13.8) = 3.3
\end{aligned}
$$

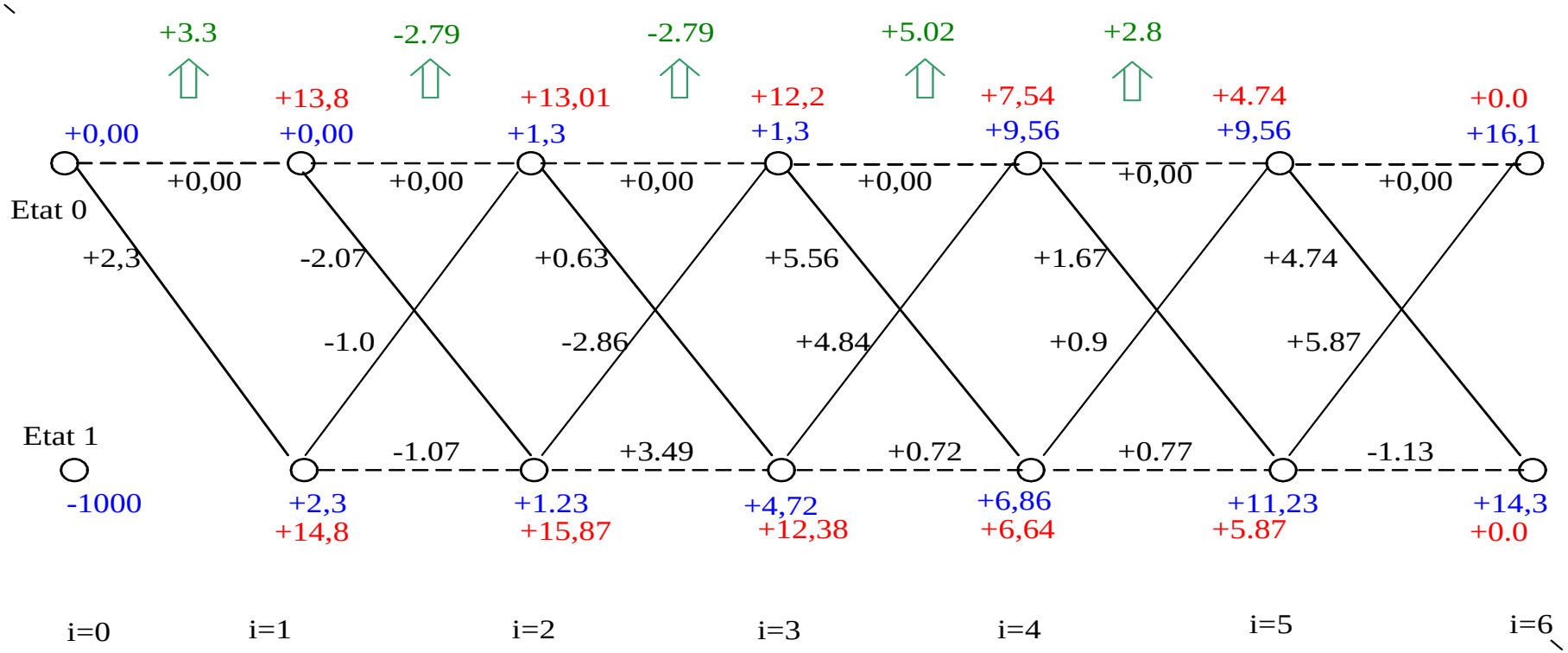


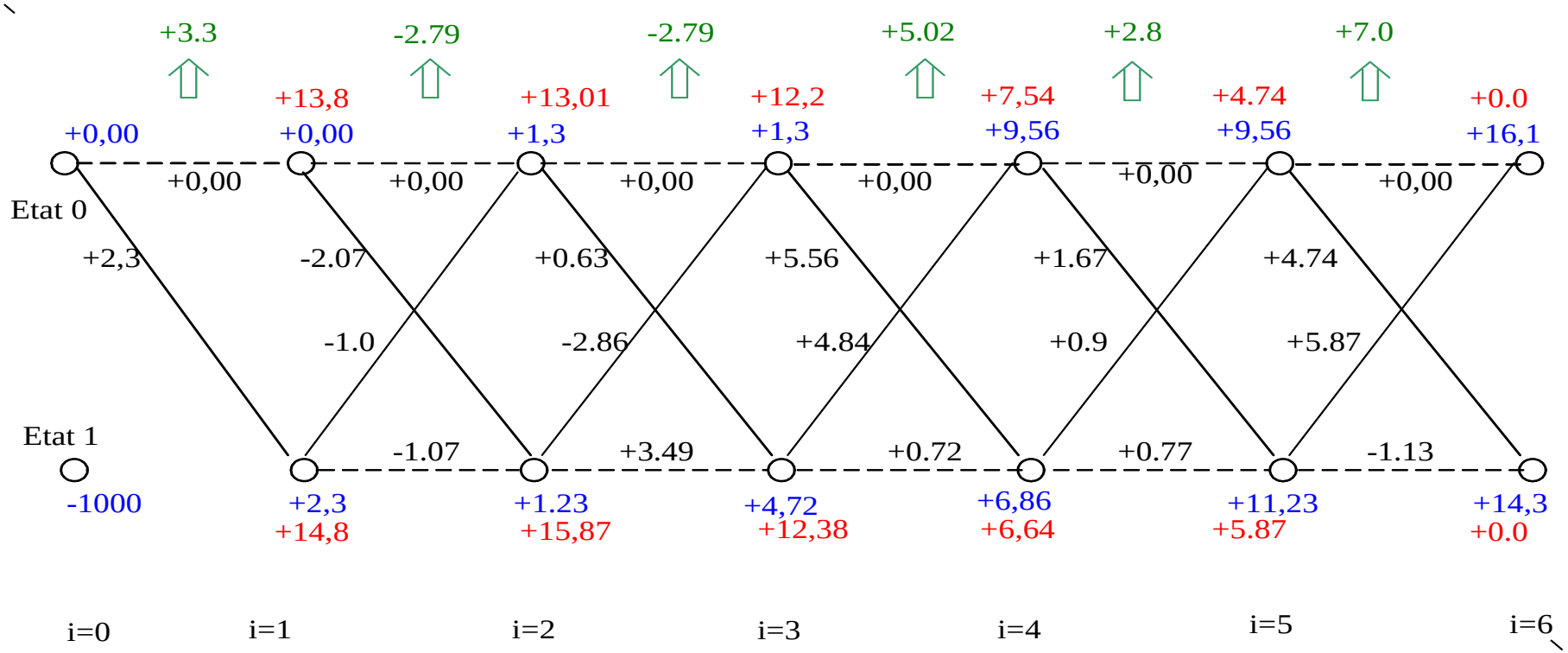
$i = 1$

$$
\begin{aligned}
L_{APP}(x_1^S) &= \max(\ln \alpha_1(0) + \ln \gamma_1^{11} + \ln \beta_2(1); \ln \alpha_1(1) + \ln \gamma_1^{10} + \ln \beta_2(0)) \\
&\quad - \max(\ln \alpha_1(0) + \ln \gamma_1^{00} + \ln \beta_2(0); \ln \alpha_1(1) + \ln \gamma_1^{01} + \ln \beta_2(1)) \\
&= \max(0 - 2.07 + 15.87; 2.3 - 1 + 13.01) \\
&\quad - \max(0 + 0 + 13.01; +2.3 - 1.07 + 15.87) \\
&= 14.31 - 17.1 = -2.79
\end{aligned}
$$


+3.3
-2.79
-2.79
+13,8
+13,01
+12,2
+7,54
+4.74
+0.0
+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+14,8
+1.23
+15,87
+4,72
+12,38
+6,86
+6,64
+11,23
+5.87
+14,3
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+3.3
-2.79
-2.79
+5.02
+13,8
+13,01
+12,2
+7,54
+4.74
+0.0
+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+14,8
+1.23
+15,87
+4,72
+12,38
+6,86
+6,64
+11,23
+5.87
+14,3
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+3.3
-2.79
-2.79
+5.02
+2.8
+13,8
+13,01
+12,2
+7,54
+4.74
+0.0
+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+14,8
+1.23
+15,87
+4,72
+12,38
+6,86
+6,64
+11,23
+5.87
+14,3
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6


+3.3
-2.79
-2.79
+5.02
+2.8
+7.0
+13,8
+13,01
+12,2
+7,54
+4.74
+0.0
+0,00
+0,00
+1,3
+1,3
+9,56
+9,56
+16,1
+0,00
+0,00
+0,00
+0,00
+0,00
+0,00
Etat 0
+2,3
-2.07
+0.63
+5.56
+1.67
+4.74
-1.0
-2.86
+4.84
+0.9
+5.87
Etat 1
-1.07
+3.49
+0.72
+0.77
-1.13
-1000
+2,3
+14,8
+1.23
+15,87
+4,72
+12,38
+6,86
+6,64
+11,23
+5.87
+14,3
+0.0
i=0
i=1
i=2
i=3
i=4
i=5
i=6

Finalement après calcul, on obtient les informations a posteriori suivantes:

$L_{APP}(x_i^S)$	+3.3	-2.79	-2.79	+5.02	+2.8	+7.0

Ainsi le décodeur max log MAP donne des valeurs sensiblement différentes par rapport au décodeur MAP. Ces différences proviennent de l'approximation max. En pratique on préfére cependant utiliser le décodeur max log MAP car il est plus facile à mettre en oeuvre.

CODE LDPC

• Les codes LPDC (low density parity check codes en anglais) ont été introduits par Gallager en 1963 et ont été redécouverts en 1996 par MacKay

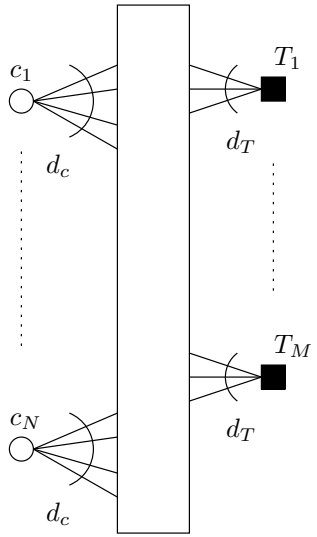
• Un code LDPC (N, d_c, d_T) est un code linéaire binaire défini par sa matrice de parité $\mathbf{H}$

• Cette matrice est de faible densité c'est-à-dire que le nombre de "1" sur chacune de ses lignes est petit devant la taille du bloc de bits informations. Chaque ligne de la matrice $\mathbf{H}$ contient d_T "1" et chaque colonne contient d_c "1" et des "0" partout ailleurs.

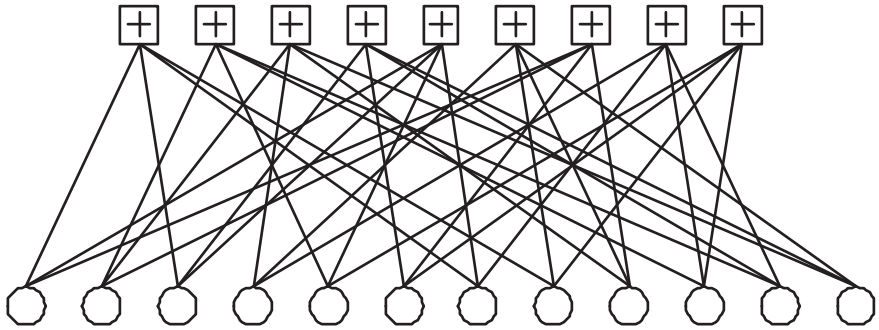
• Lorsque toutes les lignes de la matrice $\mathbf{H}$ sont indépendantes, le rendement du code LDPC est $R = 1 - \frac{d_c}{d_T}$

• Le graphe de Tanner associé à ce code est régulier si les nœuds de variable ont tous le même degré d_c et si les nœuds de contrôle ont aussi tous le même degré d_T.

• Graphe de Tanner d'un code LDPC (N, d_c, d_T) régulier :



exemple. Soit le code LDPC (12,3,4) dont le graphe de Tanner est le suivant :



La matrice de parité associée à ce code est la suivante :

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{29}$$

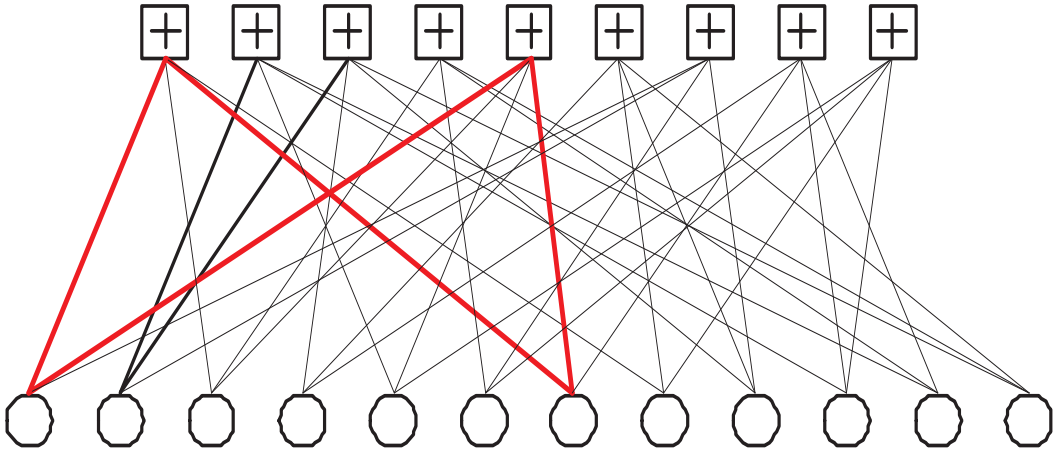
• Le rendement de ce code est égal à $R = \frac{1}{4}$, et la longueur du mot d'information est $K = 3$.

GIRTH D'UN GRAPHE

• Lorsque le graphe contient des cycles, l'algorithme somme produit devient sous optimal

=> nous devons éviter les cycles courts

• Girth : longueur du cycle le plus court dans le graphe



• Le girth détermine le nombre d'itération pour qu'un message envoyé par un noeud soit lui-même renvoyé à ce noeud

• Le girth est relié à la distance minimale du code

CONSTRUCTION D'UN CODE LDPC

- Concevoir une matrice de parité avec un girth élevé est un problème combinatoire :
  - Méthode de recherche exhaustive
  - Géométrie finie (Kou, Lin and Fossorier)
  - Steiner or Kirkman triple systems (MacKay and Davey, Johnson and Weller, Vasic)
  - LDPC Quasi cyclique LDPC basé sur une matrice de permutation (Song and Kumar)
  - ...

QUELQUES PROPRIETES DES LDPC

• La distance minimale moyenne croit seulement logarithmiquement avec K lorsque $d_c = 2$.

• La distance minimale moyenne croit linéairement avec K lorsque $d_c > 2$.

• Les codes LDPC irréguliers permettent de s'approcher de la limite de Shannon pour une longueur de mot de code très grande.

DECODAGE ITERATIF A DECISIONS DURES

On considère la transmission d'un mot de code LDPC sur un canal BSC. Soit y_m le bit reçu (0 ou 1) associé au nœud de variable c_m.

Définissons $\mu^i_{c_n \to T_m}(c_n)$ le bit transmis du nœud de variable c_n vers le nœud de contrôle T_m et soit $\mu^i_{T_m \to c_n}(c_n)$ le bit transmis du nœud de contrôle T_m vers le nœud de variable c_n à l'itération i.

Pour chaque branche (c_n, T_m):

- **Mise à jour des bits** $\mu^i_{c_n \to T_m}(c_n)$:

  A l'itération 0, $\mu^0_{c_n \to T_m}(c_n) = y_m$

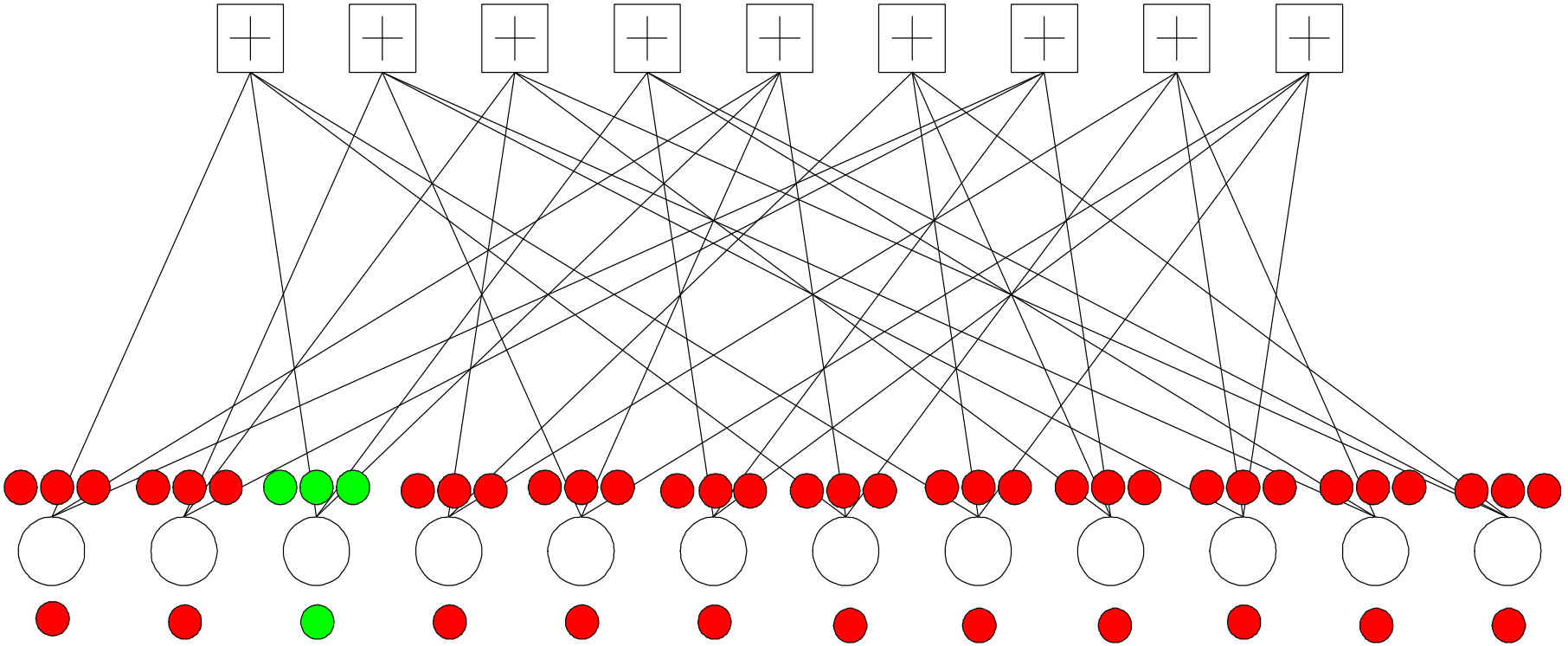
  A l'itération i avec $i > 0$,

  Si tous les bits $\mu^{i-1}_{T_{m'} \to c_n}(c_n) = b$ où $T_{m'}$ sont les nœuds de controle reliés à c_n excepté T_m(soit $d_c - 1$ bits) alors $\mu^i_{c_n \to T_m}(c_n) = b$

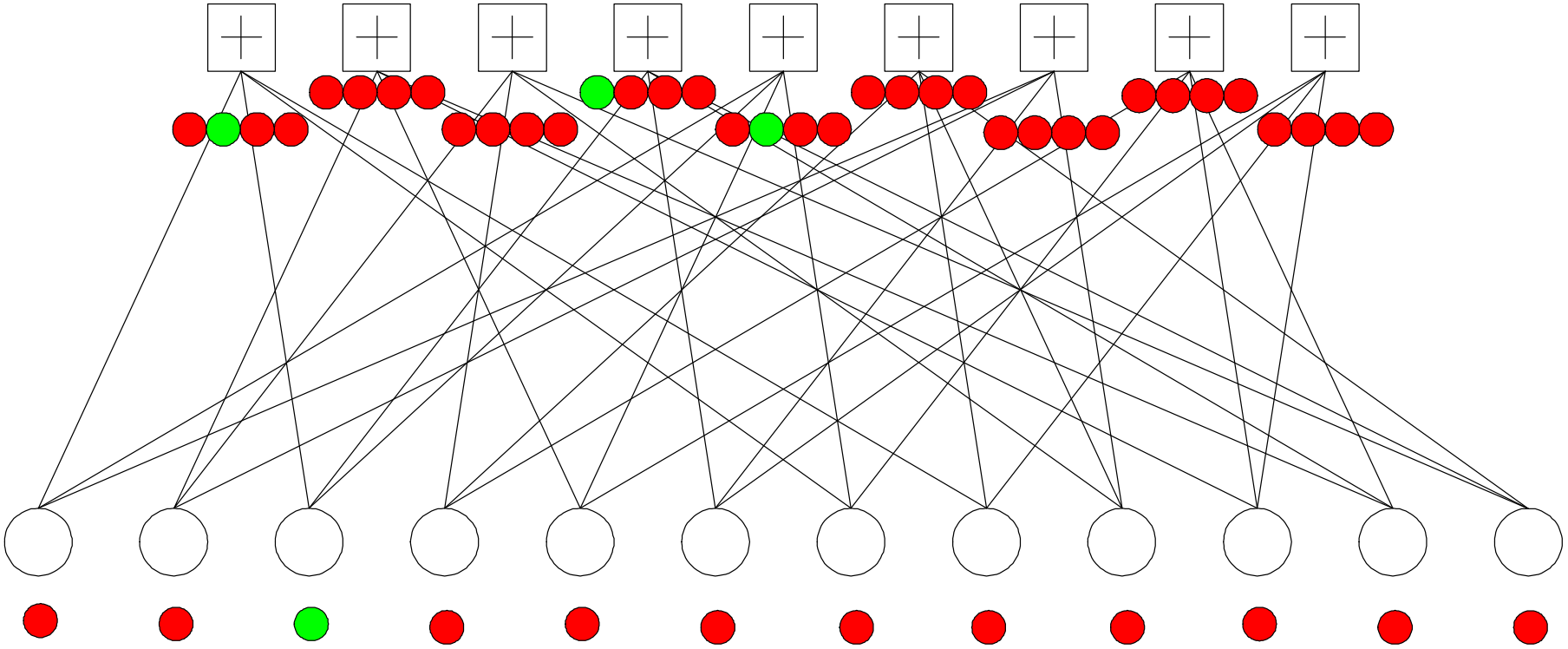
  Sinon, $\mu^i_{c_n \to T_m}(c_n) = y_m$

- **Mise à jour des bits** $\mu^i_{T_m \to c_n}(c_n)$:

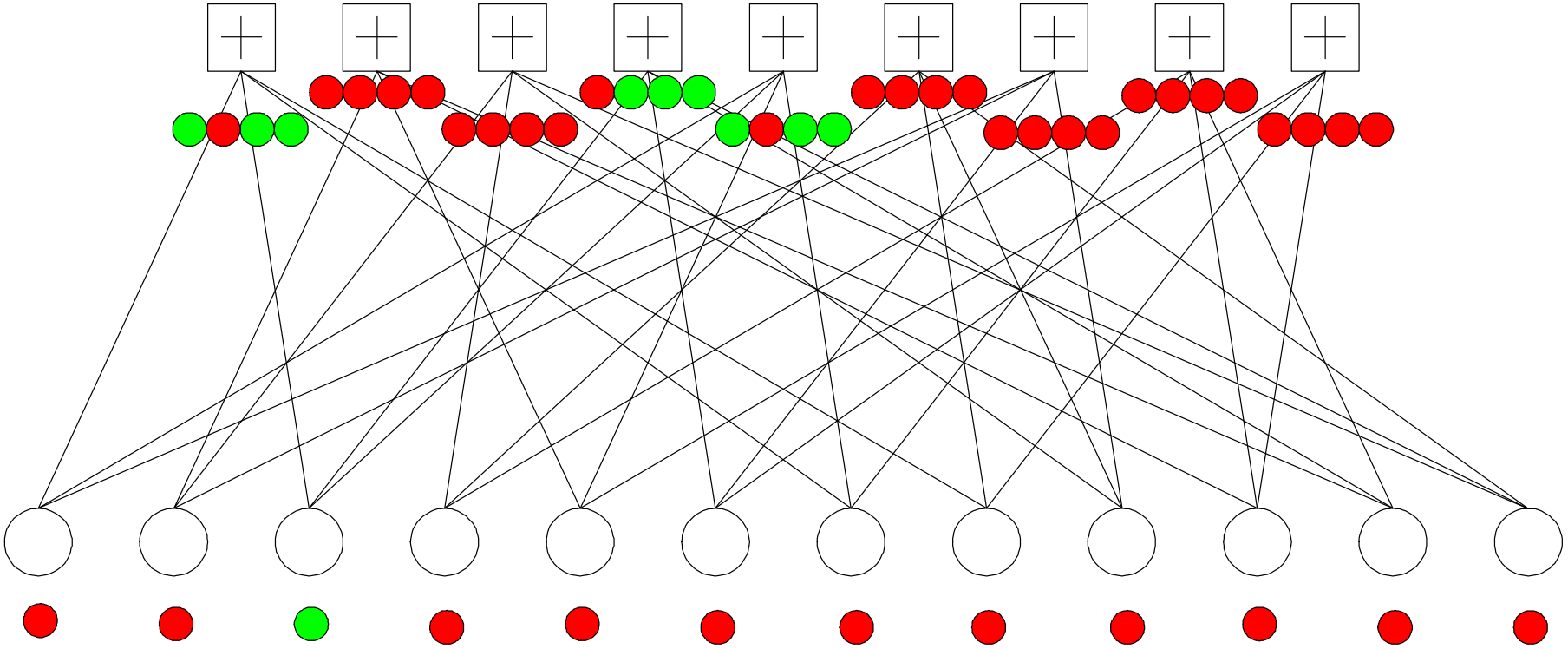
  $\mu^i_{T_m \to c_n}(c_n)$ est la somme modulo 2 des bits $\mu^i_{c_{n'} \to T_m}(c_{n'})$ où $c_{n'}$ sont les nœuds de variable reliés à T_m excepté c_n(soit $d_T - 1$ bits)



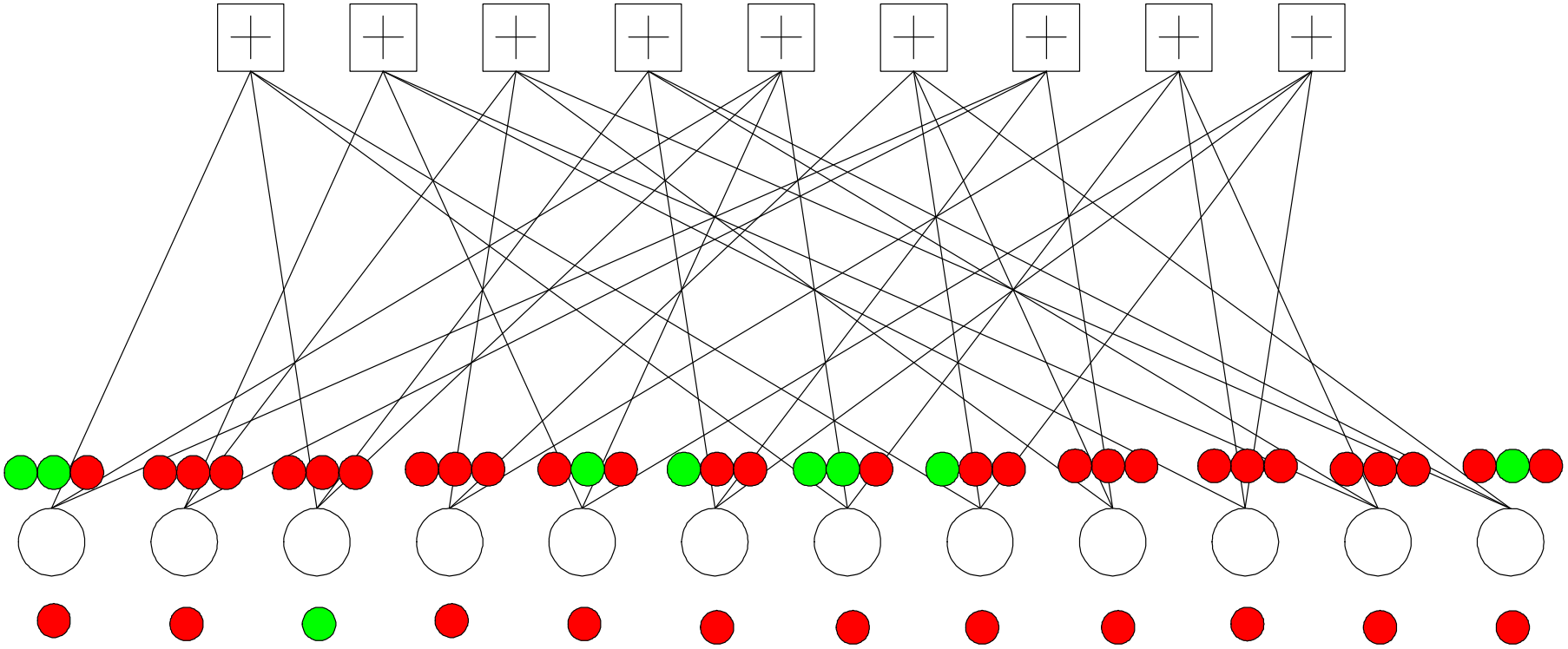
- Soit le code LDPC ($N = 12$, d_C=3, d_T=4)
- On émet le mot de code "00...0".
- On a une erreur de transmission
- On applique l'algorithme A de Gallager :



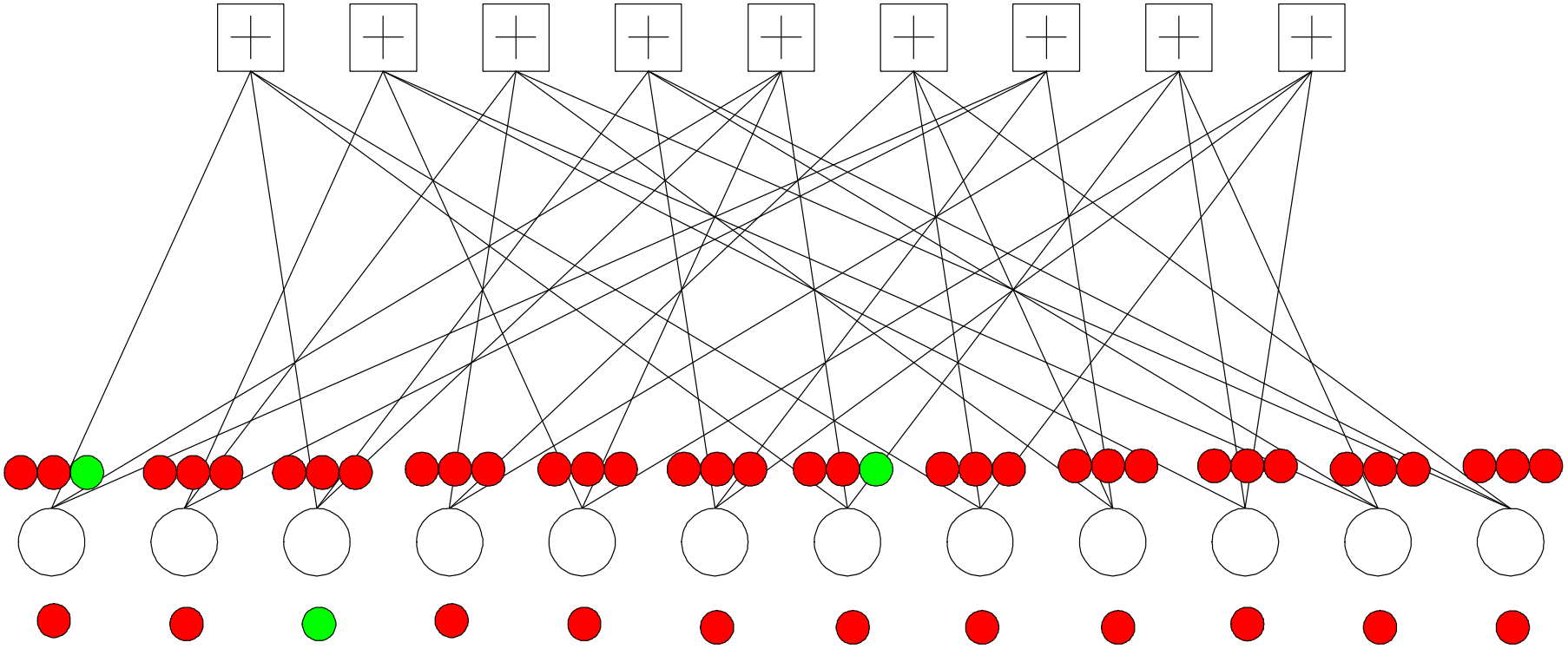
- Itération 1



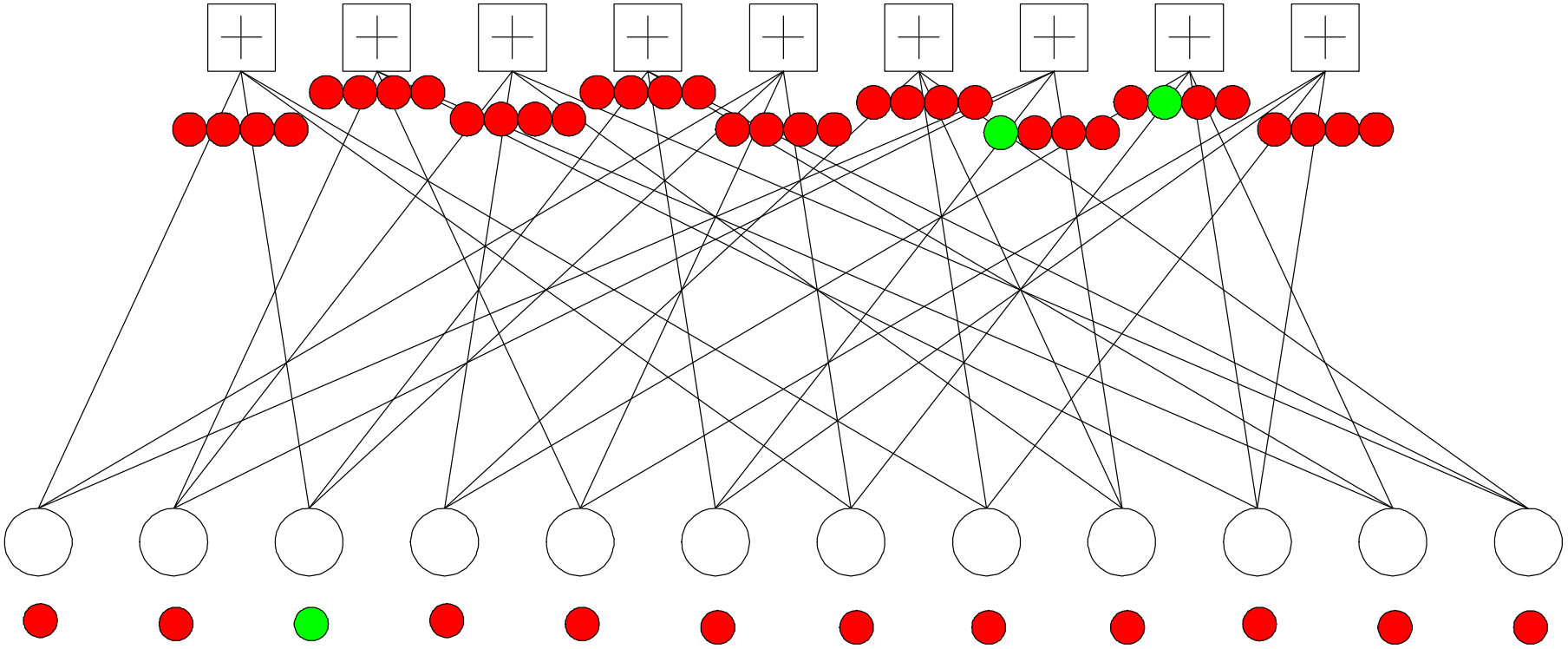
- Itération 1 : décodage horizontal



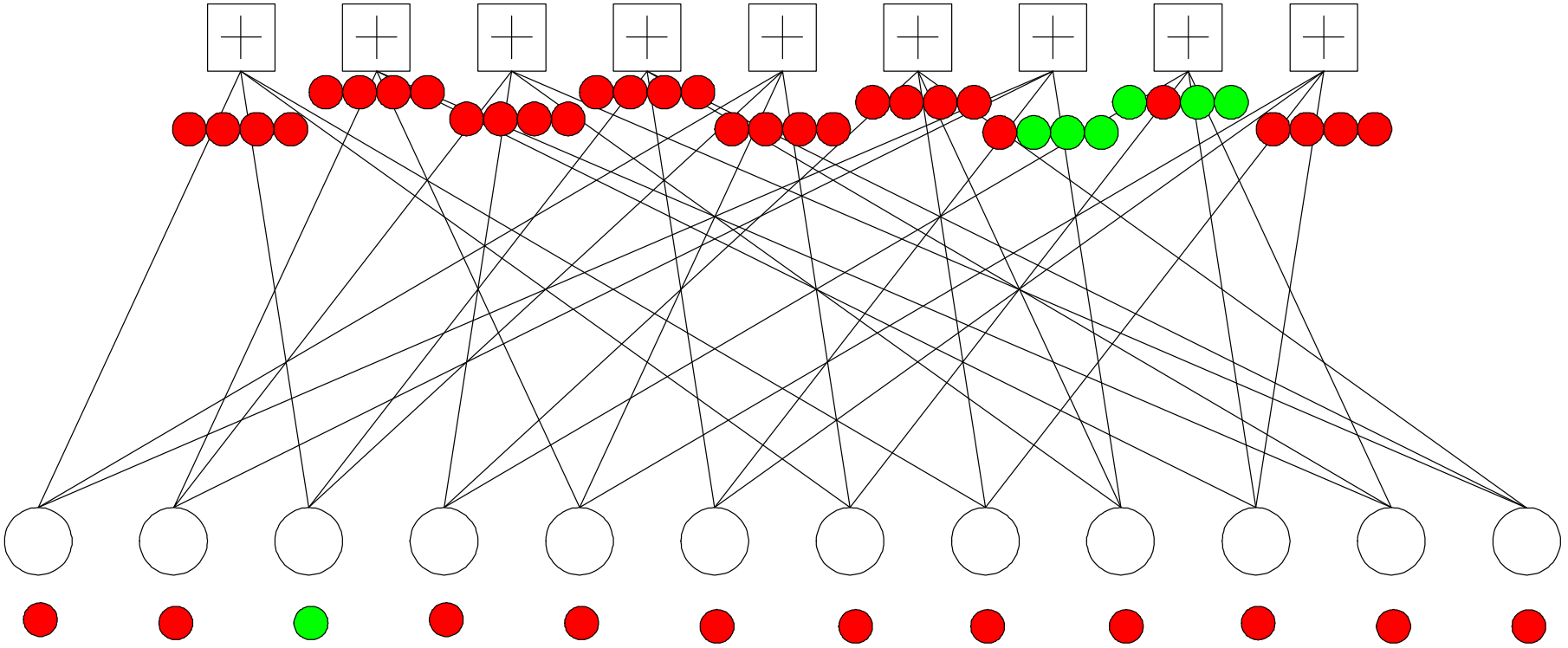
- Itération 1



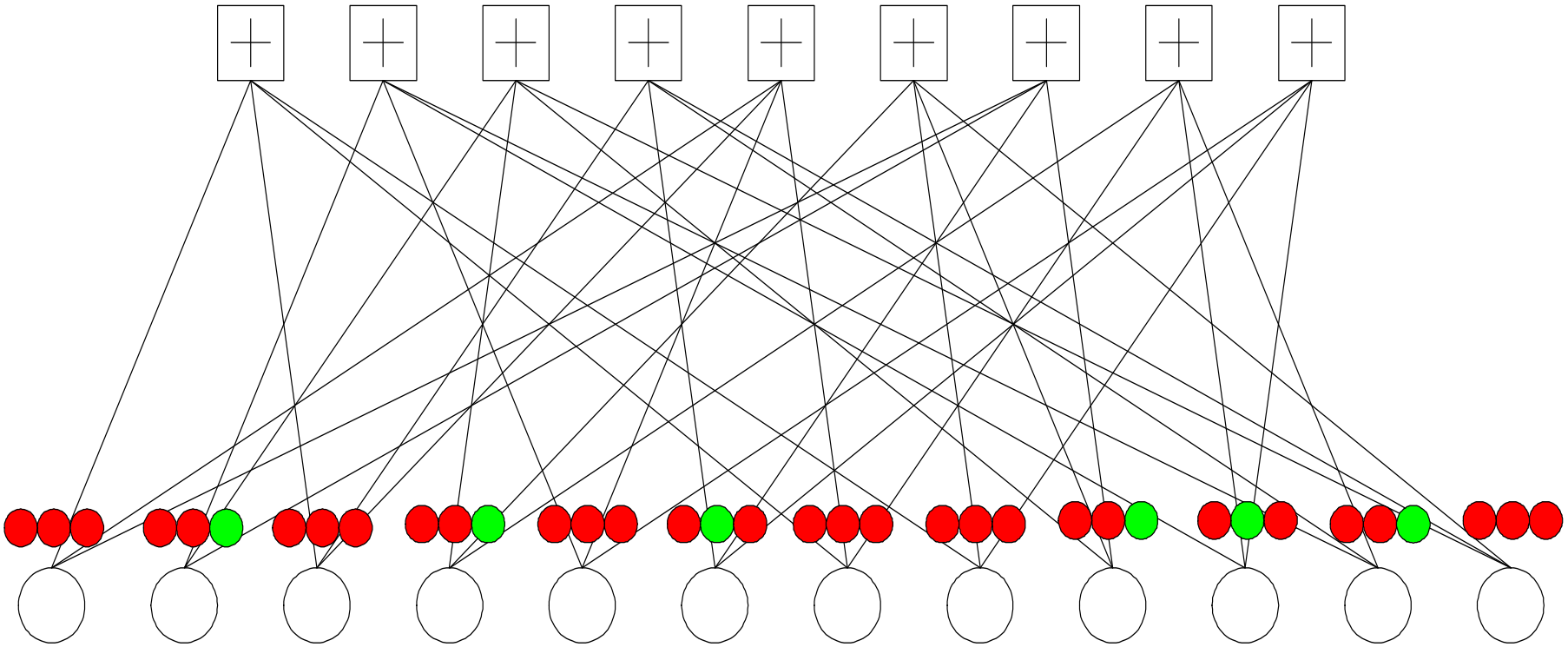
- Itération 1 : décodage vertical



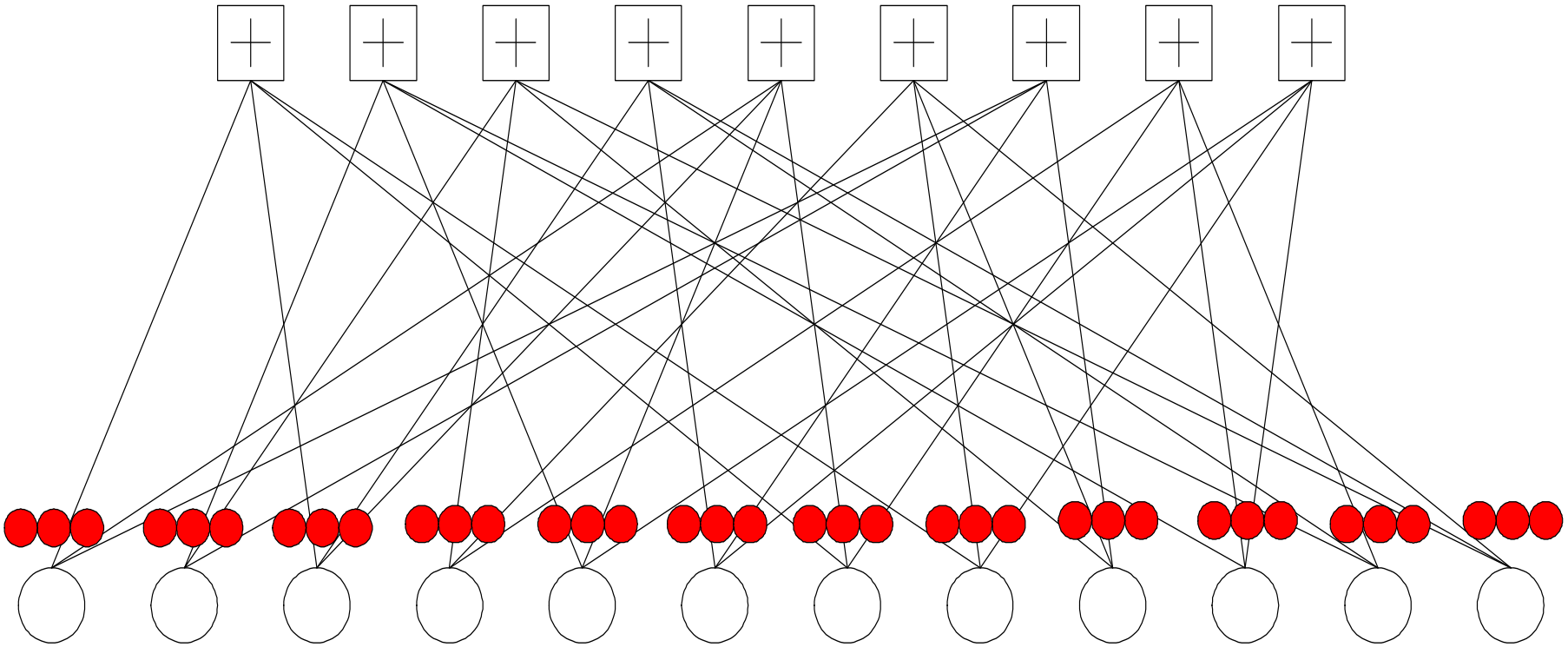
- Itération 2



- Itération 2 : décodage horizontal



- Itération 2

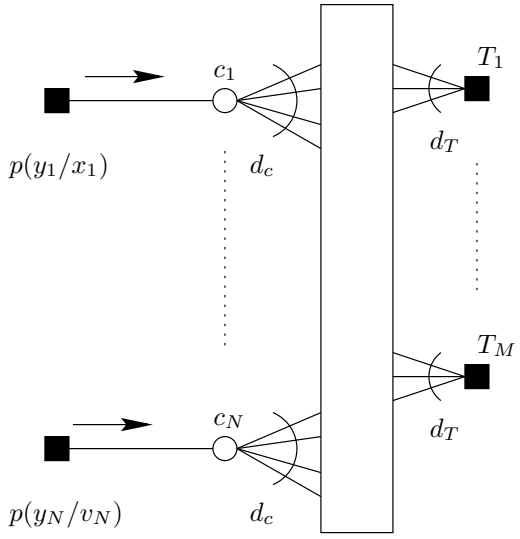


- Itération 2 : décodage vertical

DECODAGE ITERATIF A DECISIONS PONDEREES

• On suppose que la taille des mot de code est très grande.

• Le décodage d'un code LPDC consiste à appliquer l'algorithme Somme-Produit sur le graphe de Tanner du code.



• Pour des tailles de bloc d'information K très grandes, les messages se propagent sur un arbre avec une probabilité $1 - \mathbb{O}(\frac{1}{K})$.

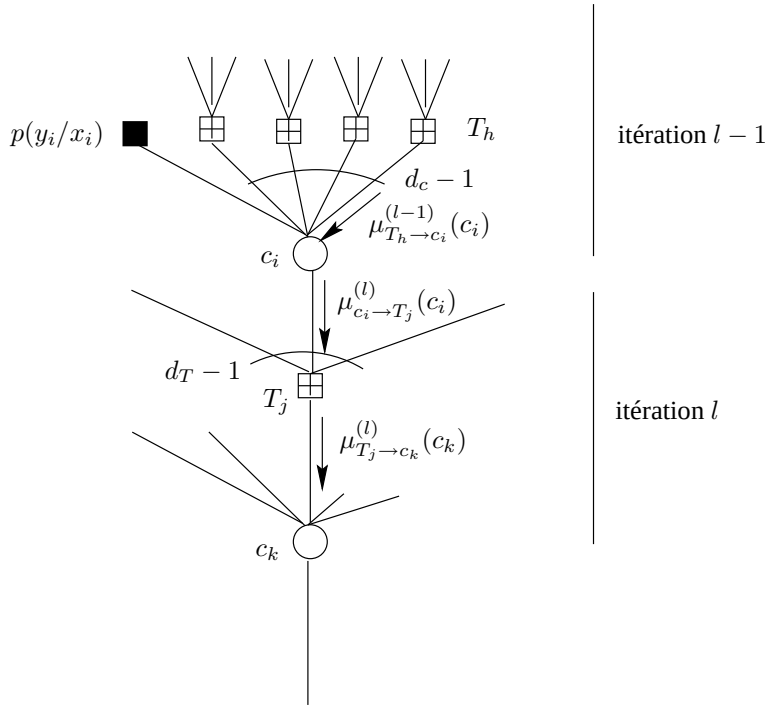
• Chaque itération comprend deux phases :

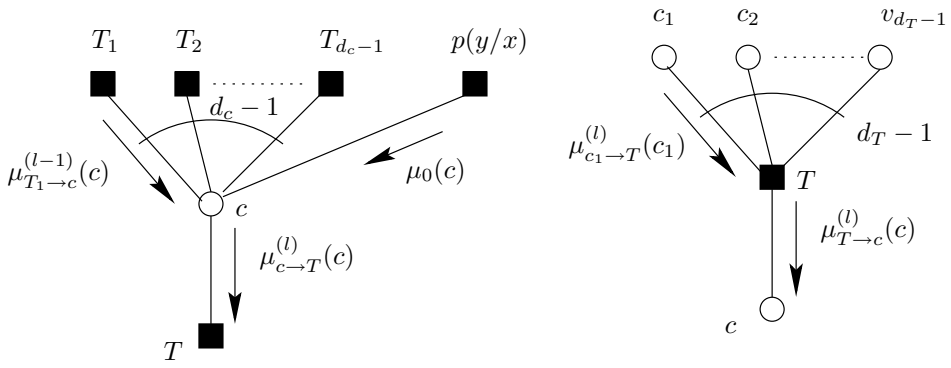
-calcul des messages nœud de variable vers nœud de contrôle

-calcul des messages nœud de contrôle vers nœud de variable

• Dans les deux phases, chaque message est calculé à partir de tous les messages entrant à l'exception du message issu de la branche considérée.

• En l'absence de cycle, le décodage est appliqué sur un arbre :





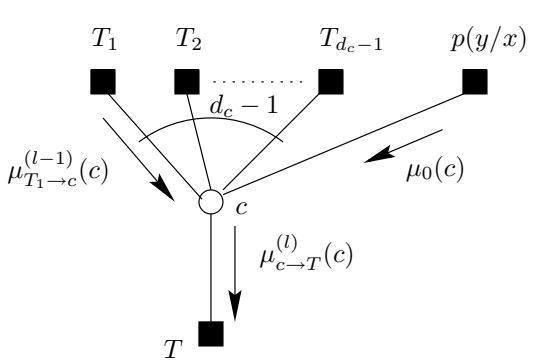
• On utilise les 2 règles de calcul de l'algorithme Somme Produit (ou Minimum Somme)

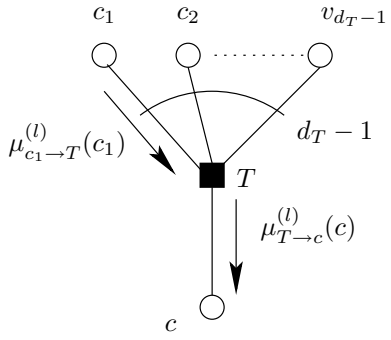
-calcul des messages nœud de variable vers nœud de contrôle $\mu^{(l)}_{c\to T}(c)$:

$$\mu^{(l)}_{c\to T}(c) = \sum_{i=1}^{d_T-1} \mu^{(l-1)}_{T_i\to c}(c) + \mu_0(c) \tag{30}$$

Lors de la première itération, nous n'avons aucune information a priori:

$$\mu^{(1)}_{c\to T}(c) = \mu_0(c) = \ln \frac{p(y/x=+1)}{p(y/x=-1)} = \frac{2}{\sigma^2} y$$





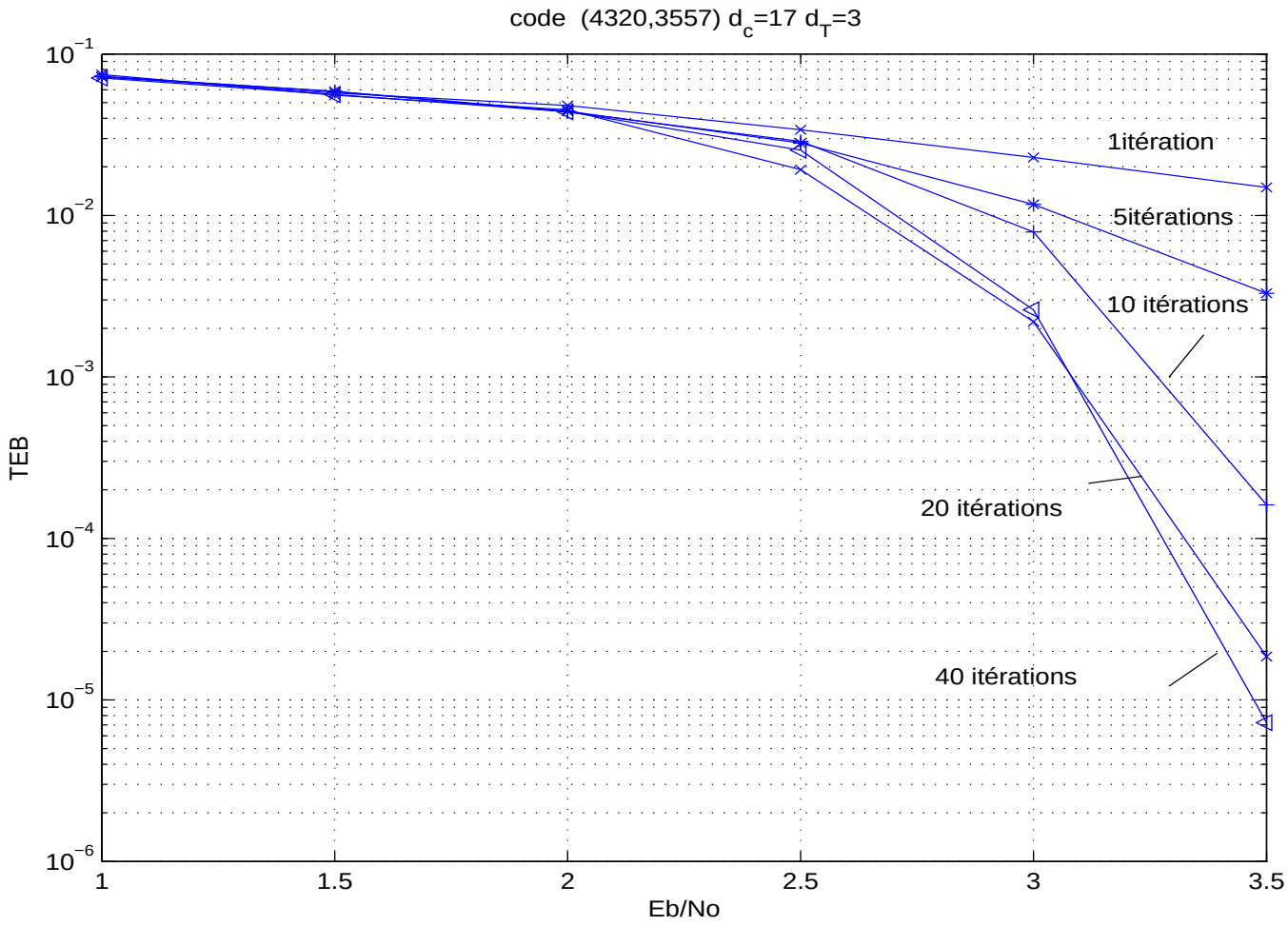
-calcul des messages contrôle vers variable $\mu^{(l)}_{T\to c}(c)$:

$$\mu^{(l)}_{T\to c}(c) = -2\tanh^{-1}\left(\prod_{i=1}^{d_c-1}\tanh\left(\frac{\mu^{(l)}_{c_i\to T}(c_i)}{2}\right)\right) \tag{31}$$

Finalement, le calcul de l'information a posteriori $APP(u)$ est effectué en sommant les informations issues de tous les nœuds de contrôle et du canal:

$$APP(x) = \sum_{i=1}^{d_c}\mu^{(l)}_{T_i\to c}(c) + \mu_0(c) \tag{32}$$

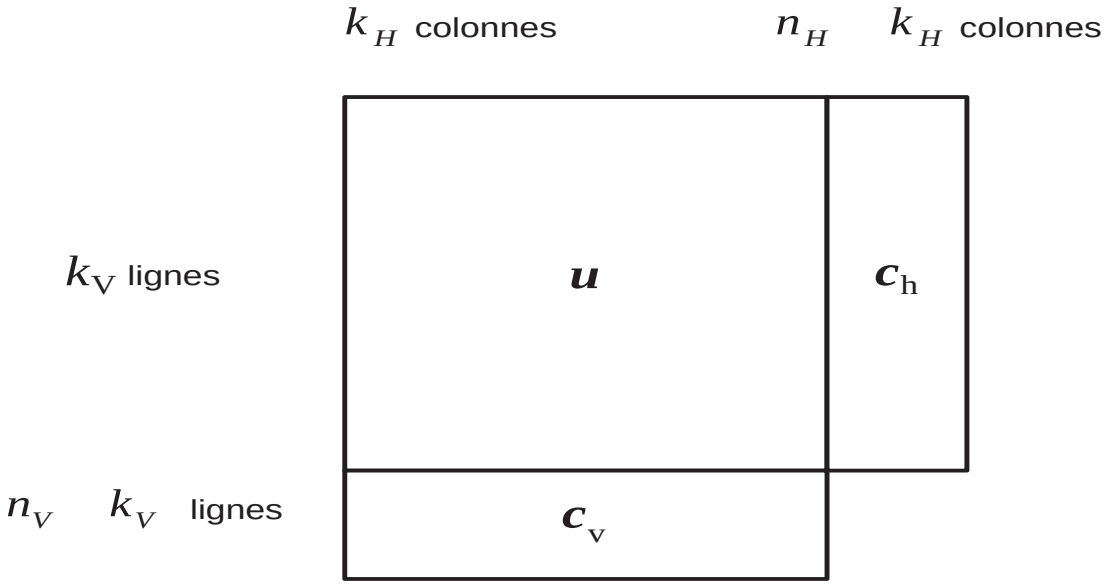
• Exemple de performance d'un code LDPC sur canal BBAG :



• Un grand nombre d'itérations est nécessaire pour atteindre la convergence du décodeur itératif

CODES PCBC

• Codes linéaires en blocs concaténés en parallèle PCBC (parallel concatenated block codes en anglais)



• $k_V k_H$ bits d'information

• $k_V(n_H - k_H)$ + $(n_V - k_V)k_H$ bits de parité

• Le rendement du code PCBC est $R_T = \frac{k_V k_H}{k_V n_H + n_V k_H - k_V k_H}$

• Chaque ligne et colonne correspond à un code en bloc linéaire systématique

EXEMPLE CODE PCBC $(8, 4)$

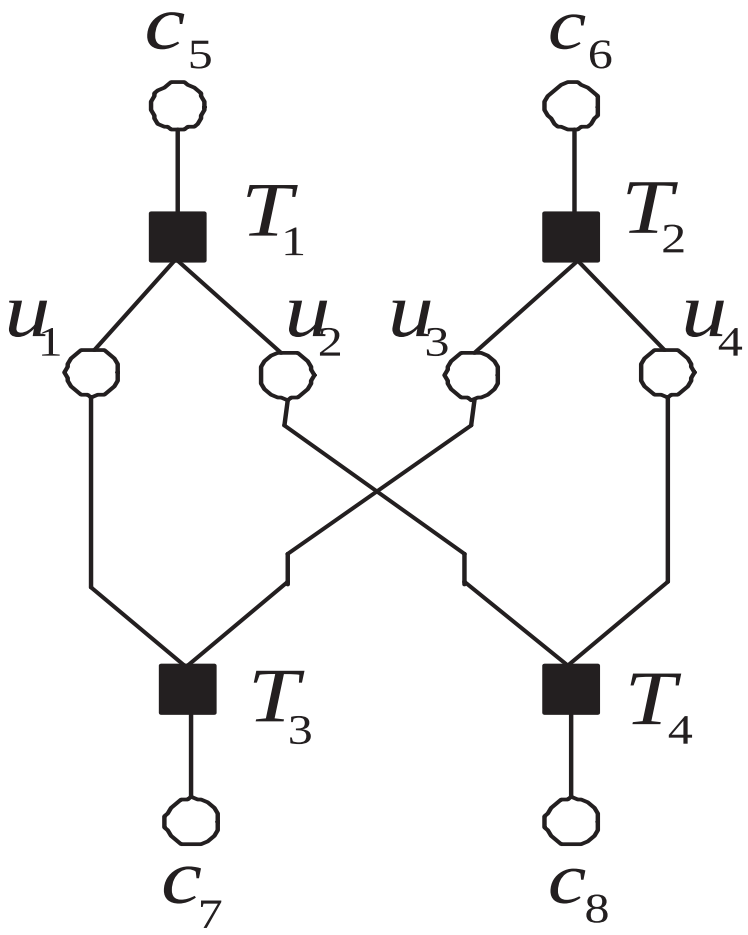
• Code défini par les 4 équations de parité :

$$\begin{aligned} u_1 + u_2 + c_5 &= 0 \qquad T_1 \\ u_3 + u_4 + c_6 &= 0 \qquad T_2 \\ u_1 + u_3 + c_7 &= 0 \qquad T_3 \\ u_2 + u_4 + c_8 &= 0 \qquad T_4 \end{aligned} \tag{33}$$

La structure de ce code PCBC est la suivante :

u_1	u_2	c_5
u_3	u_4	c_6
c_7	c_8	

Le graphe de Tanner associé est le suivant :



Considérons le mot d'information binaire suivant :

$$\mathbf{u} = [1001]$$

Le mot émis est égal à :

$$\mathbf{x} = [+1 - 1 - 1 + 1 + 1 + 1 + 1 + 1]$$

mot de code binaire

$u_1 = 1$	$u_2 = 0$	$c_5 = 1$
$u_3 = 0$	$u_4 = 1$	$c_6 = 1$
$c_7 = 1$	$c_8 = 1$	

mot émis sur le canal

$x_1 = +1$	$x_2 = -1$	$x_5 = +1$
$x_3 = -1$	$x_4 = +1$	$x_6 = +1$
$x_7 = +1$	$x_8 = +1$	

Soit la séquence reçue en sortie du canal bruit blanc additif gaussien de variance $\sigma^2 = 1$ $\mathbf{y} = \mathbf{x} + \mathbf{n}$:

séquence reçue

$y_1 = 0.75$	$y_2 = 0.05$	$y_5 = 1.25$
$y_3 = 0.1$	$y_4 = 0.15$	$y_6 = 1.0$
$y_7 = 3.0$	$y_8 = 0.05$	

Les informations intrinsèques sont calculées en utilisant la relation $L_{INT}(x_i) = \frac{2y_i}{\sigma^2}$.

Informations intrinsèques $L_{INT}(x_i)$

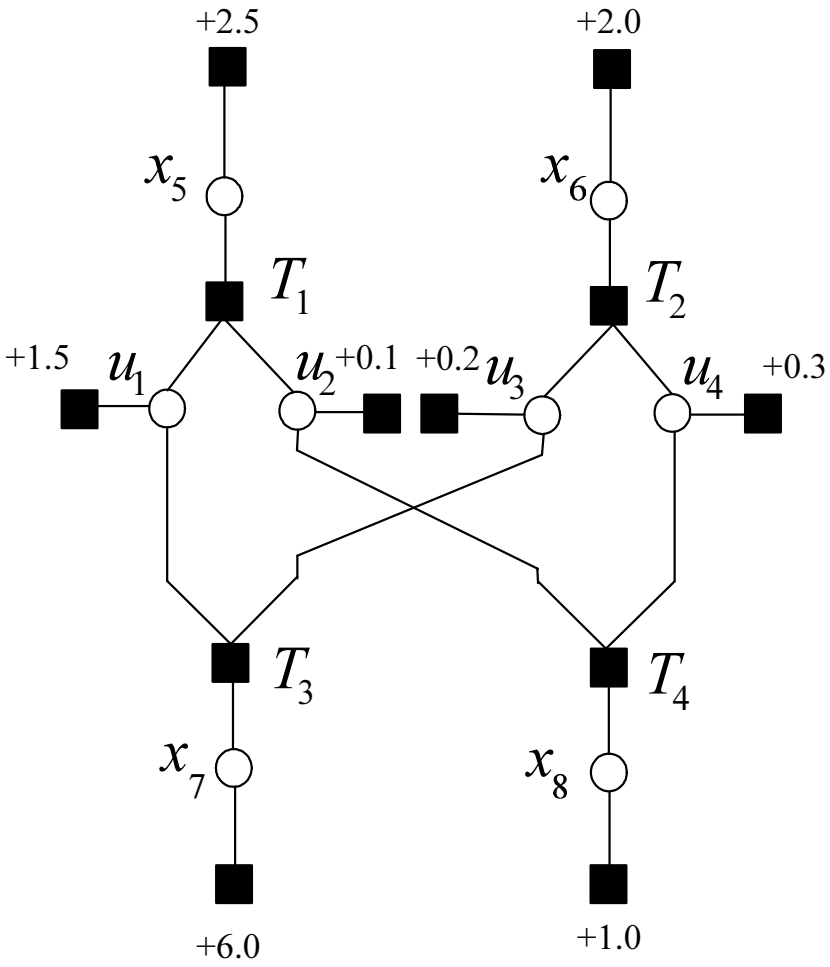
$+1.5$	$+0.1$	$+2.5$
$+0.2$	$+0.3$	$+2.0$
$+6.0$	$+1.0$	

A chaque itération on déterminera tout d'abord les informations extrinsèques $L^H_{EXTR}(x_i)$ (décodage horizontal) :

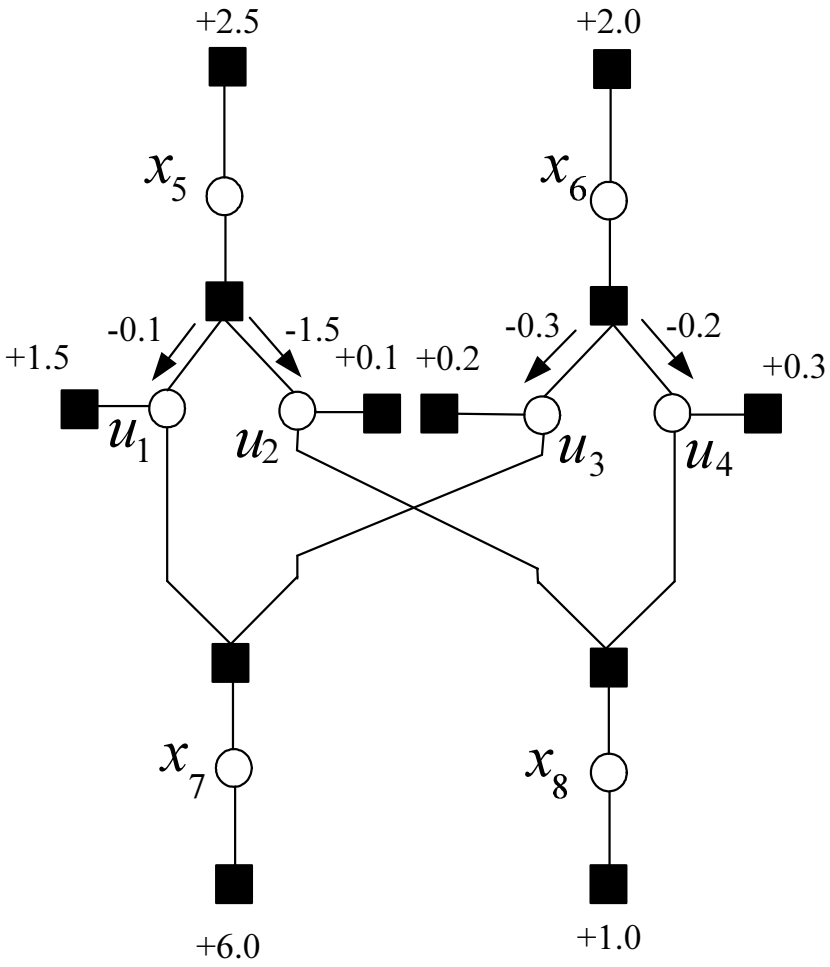
$$
\begin{aligned}
L^H_{EXTR}(x_1) &= -(L_{APRI}(x_2) + L_{INTR}(x_2)) \boxplus L_{INTR}(x_5) \\
L^H_{EXTR}(x_2) &= -(L_{APRI}(x_1) + L_{INTR}(x_1)) \boxplus L_{INTR}(x_5) \\
L^H_{EXTR}(x_3) &= -(L_{APRI}(x_4) + L_{INTR}(x_4)) \boxplus L_{INTR}(x_6) \\
L^H_{EXTR}(x_4) &= -(L_{APRI}(x_4) + L_{INTR}(x_4)) \boxplus L_{INTR}(x_6)
\end{aligned}
$$

puis ensuite les informations $L^V_{EXTR}(x_i)$ (décodage vertical)

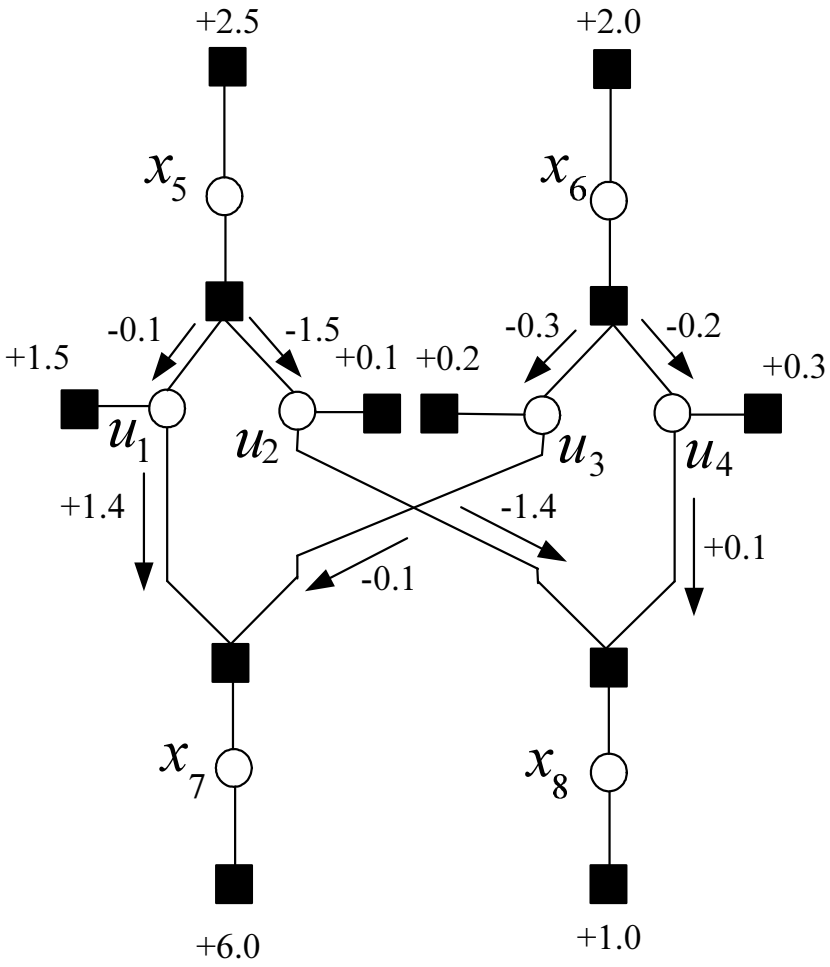
$$
\begin{aligned}
L^V_{EXTR}(x_1) &= -(L_{APRI}(x_3) + L_{INTR}(x_3)) \boxplus L_{INTR}(x_7) \\
L^V_{EXTR}(x_3) &= -(L_{APRI}(x_1) + L_{INTR}(x_1)) \boxplus L_{INTR}(x_7) \\
L^V_{EXTR}(x_2) &= -(L_{APRI}(x_4) + L_{INTR}(x_4)) \boxplus L_{INTR}(x_8) \\
L^V_{EXTR}(x_4) &= -(L_{APRI}(x_2) + L_{INTR}(x_2)) \boxplus L_{INTR}(x_8)
\end{aligned}
$$



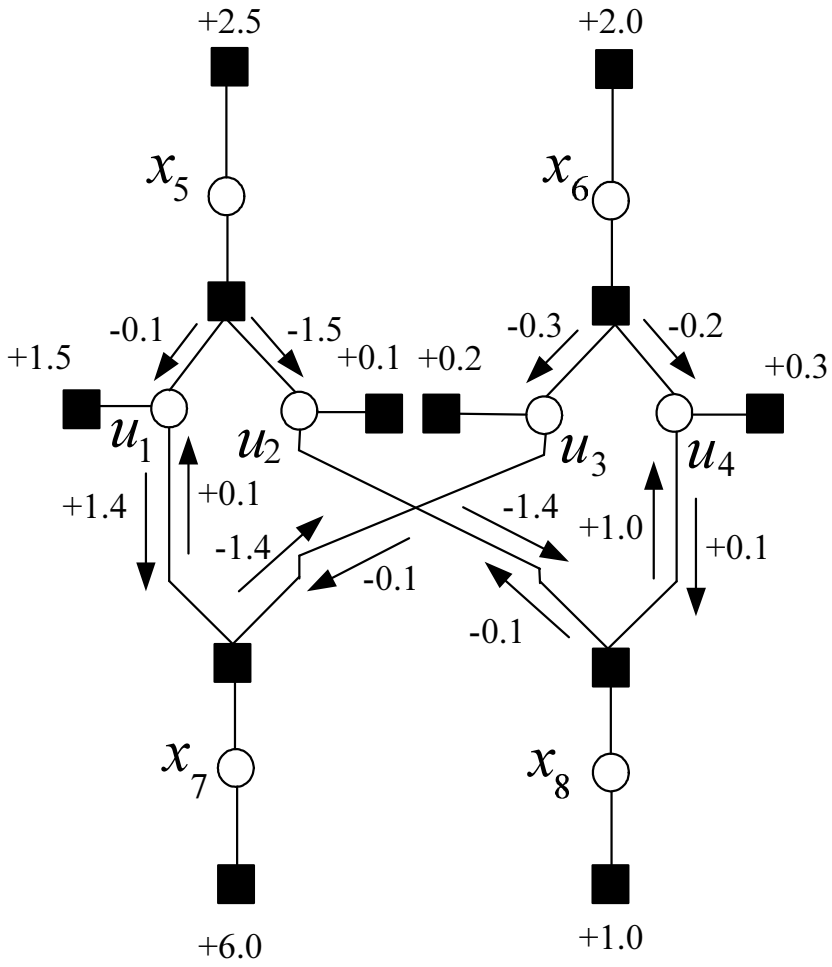
- Itération 1



- Itération 1 : décodage horizontal



- Itération 1



- Itération 1 : décodage vertical

Première itération

1.5	0.1
0.2	0.3

$L_{INTR}(x_i)$

-0.1	-1.5
-0.3	-0.2

$L_{APRI}(x_i)$

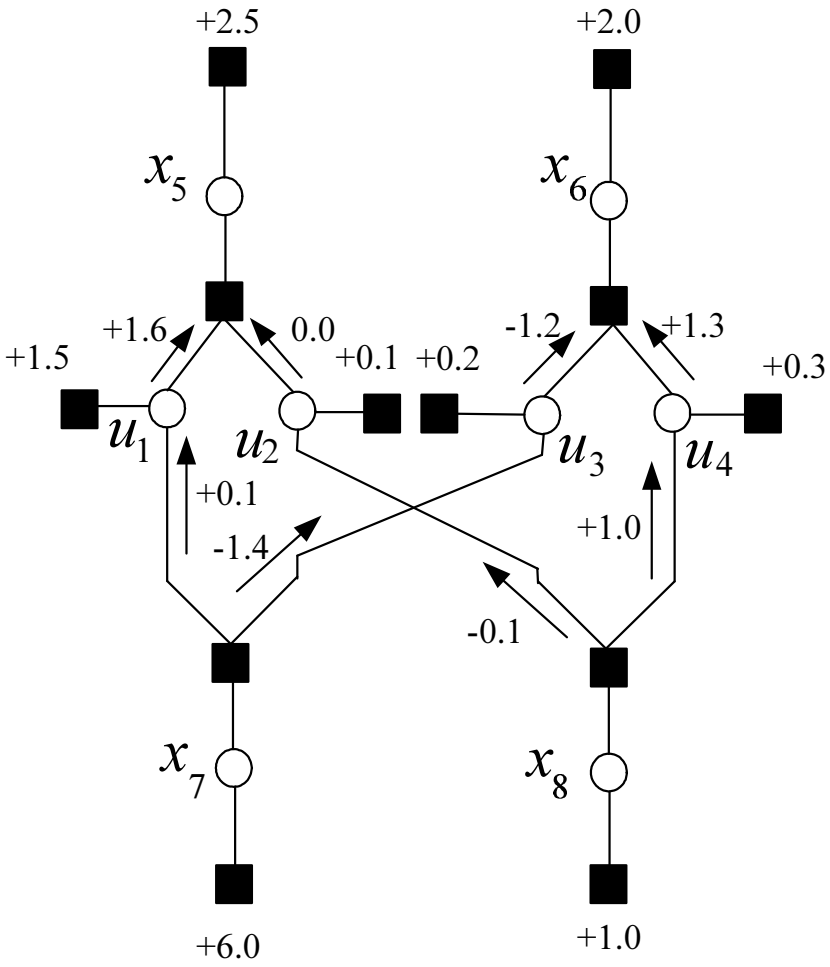
0.1	-0.1
-1.4	1.0

$L^V_{EXTR}(x_i)$

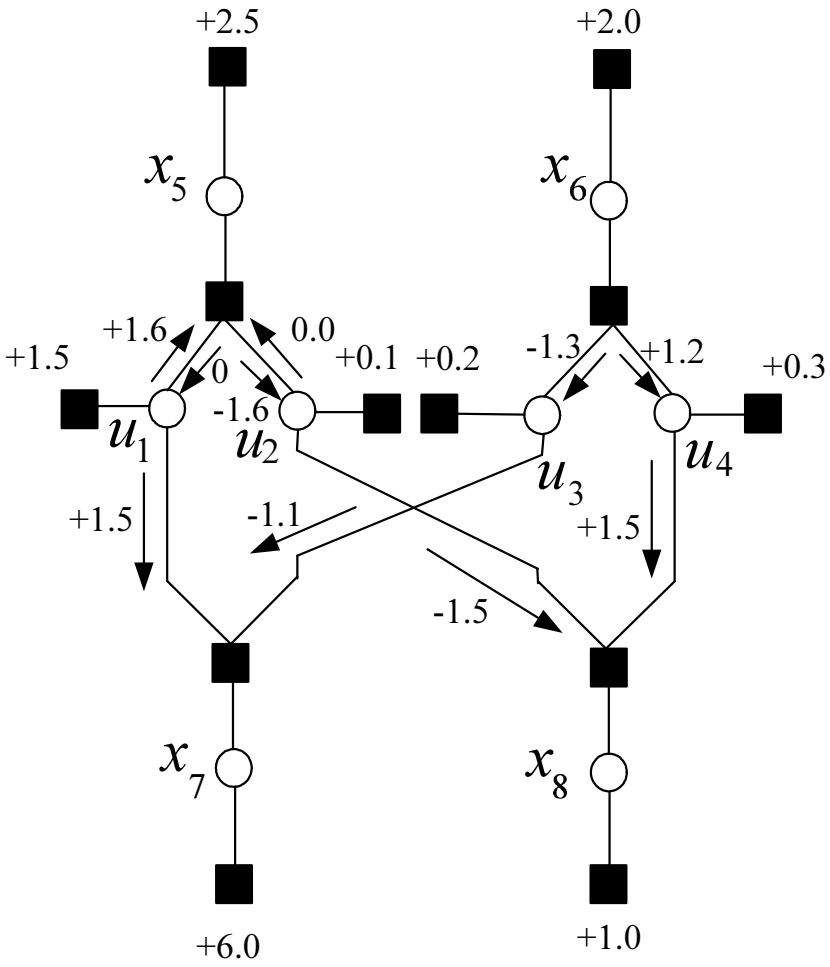
1.5	-1.5
-1.5	1.1

$L_{APP}(x_i)$

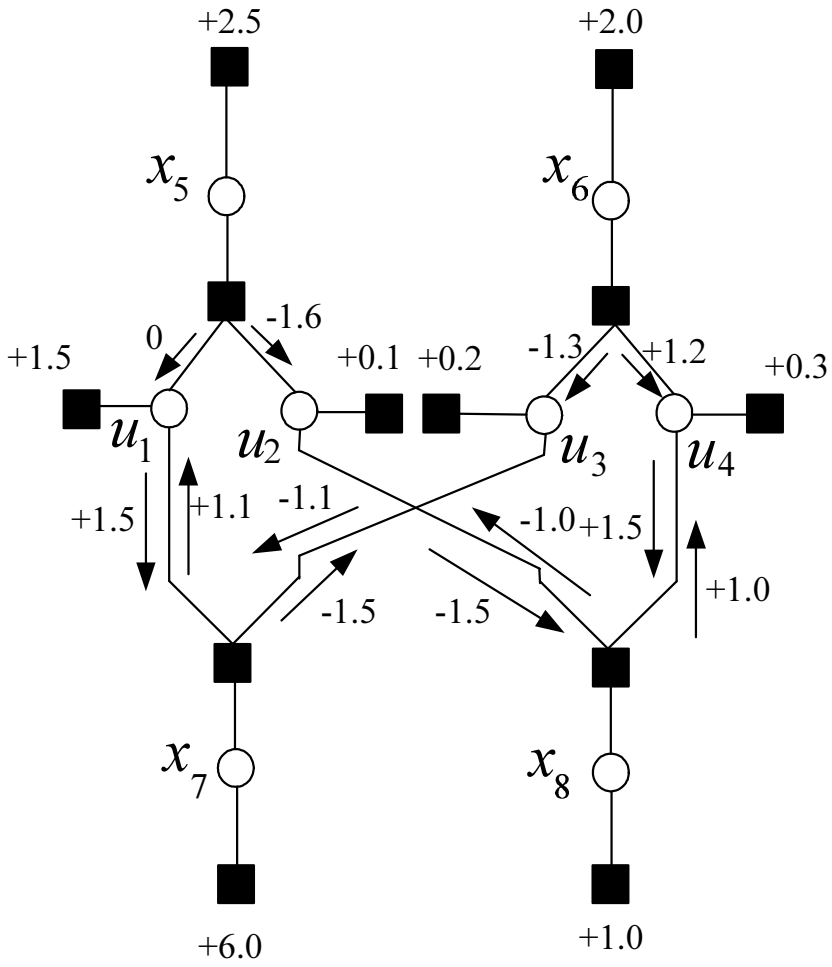
- On utilise la relation $L_{APP}(x_i) = L_{APRI}(x_i) + L_{INTR}(x_i)) + L^V_{EXTR}(x_i)$



- Itération 2



- Itération 2 : décodage horizontal



- Itération 2 : décodage

Seconde itération :

1.5	0.1
0.2	0.3

$L_{INTR}(x_i)$

0.0	-1.6
-1.3	1.2

$L_{APRI}(x_i)$

1.1	-1.0
-1.5	1.0

$L^V_{EXTR}(x_i)$

2.6	-2.5
-2.6	2.5

$L_{APP}(x_i)$

• Les fiabilités sur les informations a posteriori $L_{APP}(x_i)$ obtenues en fin de seconde itération sont meilleures.

• Cependant les informations a priori sont de plus en plus corrélées et une troisième itération n'apportera pas de gain supplémentaire.

• En pratique, 5 à 6 itérations suffisent pour obtenir des performances proches de celles d'un décodeur à maximum de vraisemblance.

CODES PRODUITS

• Les codes produits ont été introduits en 1954 par Elias.

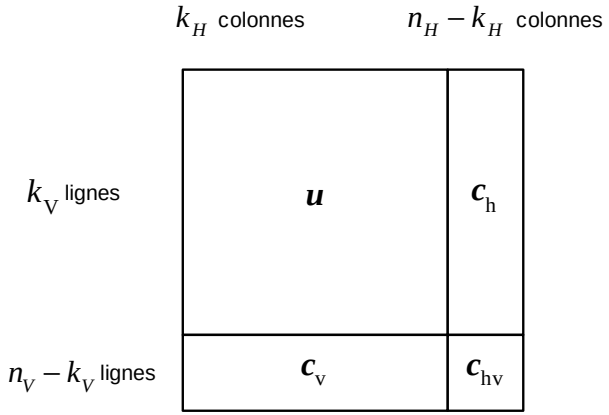


Figure 1: Code Produit.

Le code ainsi obtenu est un code linéaire en bloc systématique (N, K) avec $N = n_V n_H$ et $K = k_V k_H$.

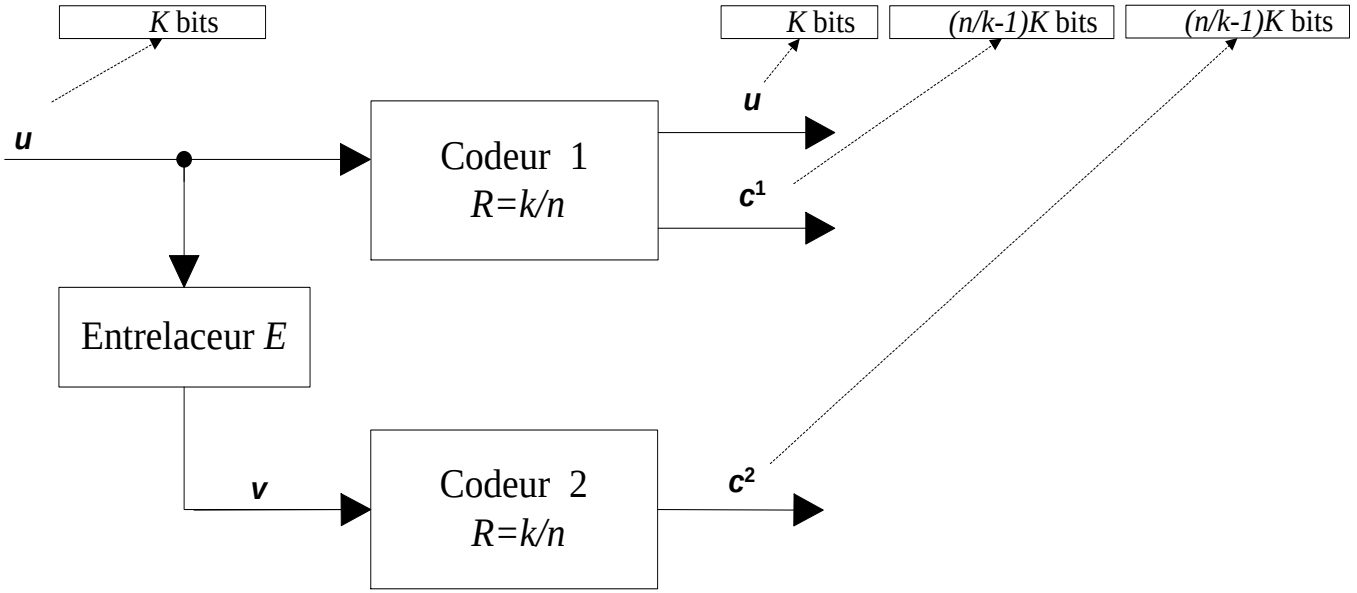
Le rendement global du code produit est égal à :

$$R_T = \frac{k_V k_H}{n_V n_H}$$

• La distance minimale de ces codes est égale au produit $d_V \times d_H$ où d_V et d_H sont respectivement la distance minimale des codes constituants C_V et C_H.

CODES CONVOLUTIFS CONCATENES EN PARALLELE (PCC) OU TURBO CODES

- Structure du turbo codeur :



- Le rendement global du code est égal à :

$$R_T = \frac{1}{2\frac{n-k}{k} + 1}$$

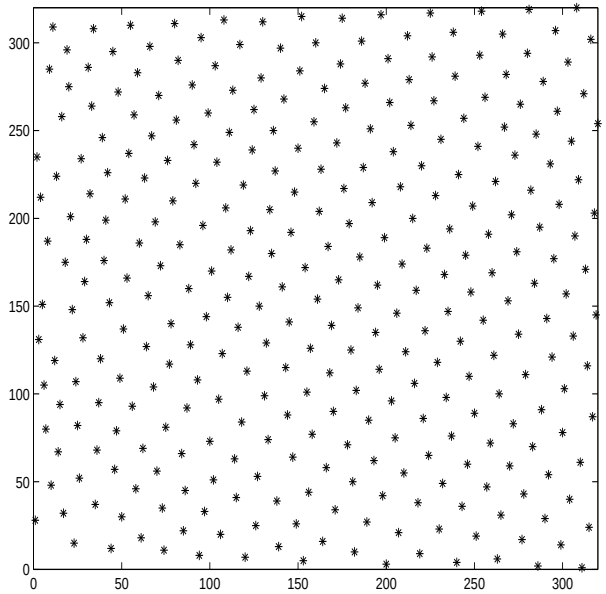
- La fonction d'entrelacement modifie l'ordre de sortie des K bits d'information.

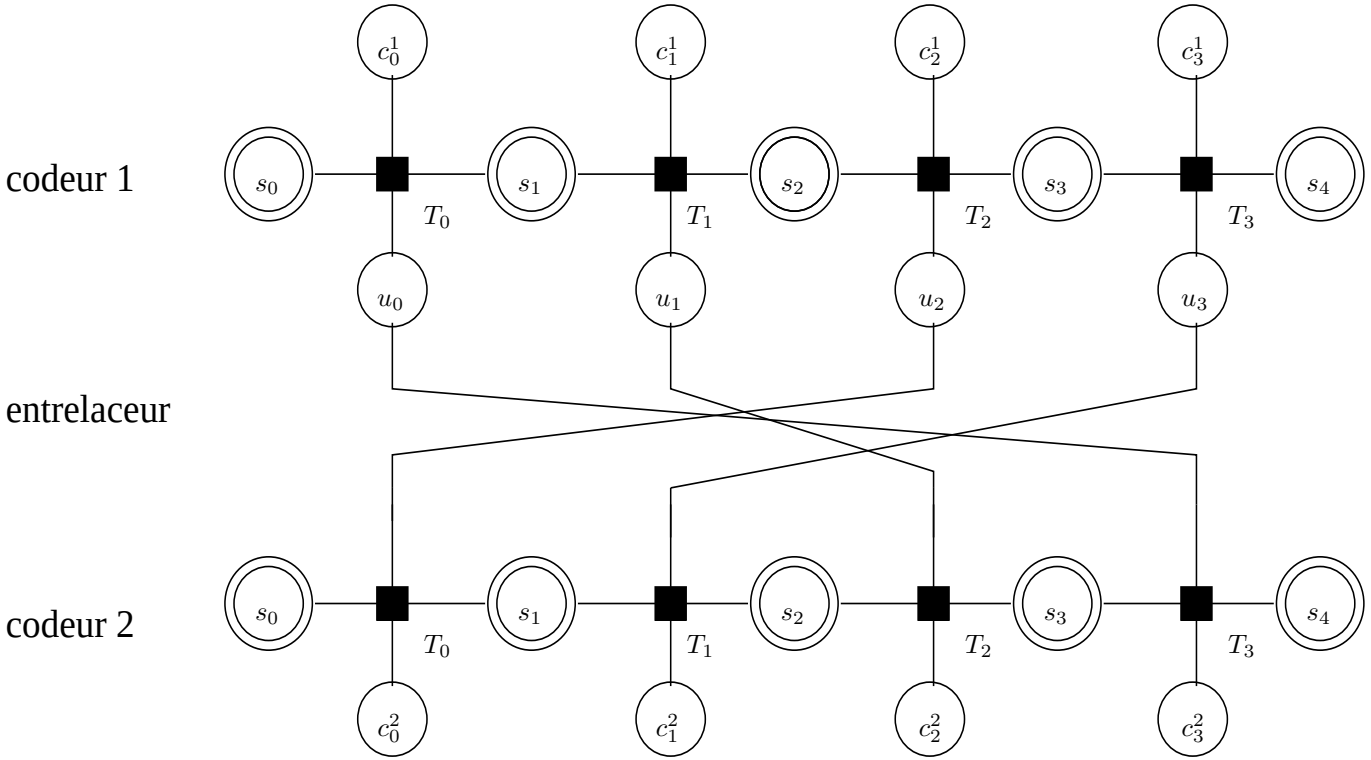
Un entrelaceur E de taille K est défini par sa matrice de permutation Π de dimension $K \times K$:

$$\mathbf{v} = \mathbf{u}\Pi \qquad \text{avec} \quad \mathbf{u} = \left[u_0, u_1, \ldots, u_{K-1}\right]$$
$$\mathbf{v} = \left[v_0, v_1, \ldots, v_{K-1}\right]$$

$$\Pi = \left\{a_{ij}\right\}_{K \times K} \quad \text{avec} \quad a_{ij} \in \{0, 1\}$$

Si a_{ij}=1 alors le bit u_i est associé au bit v_j.





• Il est possible d'effectuer une perforation en sortie du codeur.

• 2 critères de construction de l'entrelaceur :

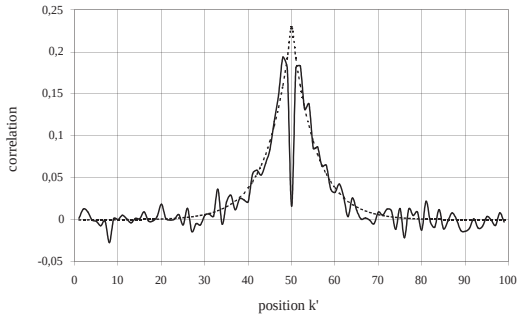
- L'entrelaceur doit permettre d'améliorer la distribution de poids des codes concaténés et d'augmenter leurs distances minimales.

- L'entrelaceur doit garantir le meilleur passage de l'information extrinsèque d'un décodeur à l'autre.

• Le rôle de la terminaison est d’éviter les mots de code de faible poids en ajoutant en fin de séquence plusieurs bits afin de ramener l’état interne du ou des codeurs convolutifs à l’état zéro.

CONSTRUCTION DE L'ENTRELACEUR

- Correlation entre Le et l'information du canal



$$\rho_{Le_i^{(k)},x_j} = \frac{\text{cov}[Le_i^{(k)}, x_j]}{\sqrt{\text{Var}[Le_i^{(k)}]\text{Var}[x_j]}}$$

- Un entrelaceur optimisant les cycles de parametre L est définit comme suit: deux bits séparés par X bits ($X \leq L-2$) dans la séquence d'entrée u doivent être séparés avec au moins $L-2-X$ bits après entrelacement.

$$I = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

ETUDE DES PERFORMANCES MOYENNES DES TURBO CODES

• On utilise un entrelaceur uniforme qui associe une séquence d'entrée de poids w avec les $\binom{K}{w}$ séquences distinctes de poids w, chacune avec la même probabilité $p = 1/\binom{K}{w}$.

• La fonction IRWEF d'un codeur RSC C dont la séquence d'entrée est terminée et de longueur K bits est la suivante :

$$A^C(W,Z) = \sum_{w=w_{min}}^{K} W^w A^C(w,Z) \quad \text{avec} \quad A^C(w,Z) = \sum_{z=z_{min}}^{\frac{n-k}{k}K} A^C_{w,z} Z^z$$

• $A^C_{w,z}$ est le nombre de mots de code de C dont le poids de la séquence d'entrée est égal à w et dont le poids de la séquence des bits de redondance est égal à z. w_{min} est le poids minimal des séquences d'entrée.

• Les coefficients de la fonction d'énumération de poids IRWEF moyenne $A^{C_P}(W, Z)$ s'exprime comme suit :

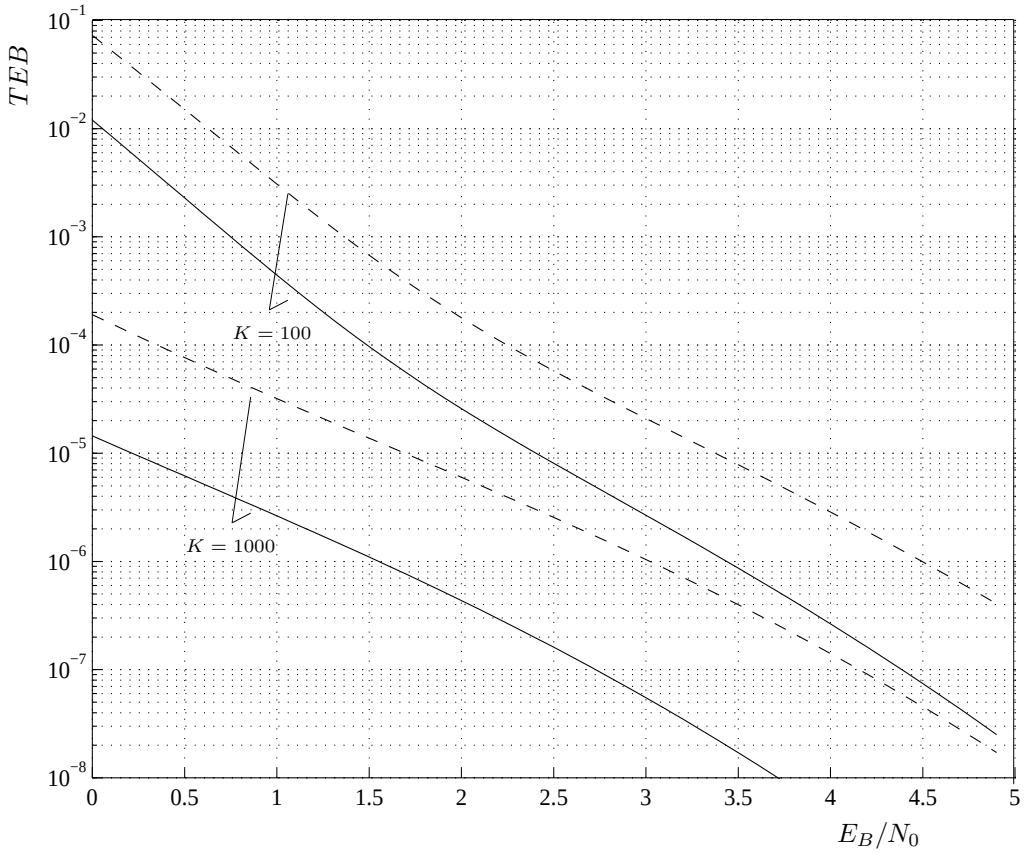
$$A^{C_P}(w, Z) = \frac{[A^C(w, Z)]^2}{\binom{K}{w}} \tag{34}$$

• Pour les codes convolutifs concaténés en parallèle, le gain d'entrelacement est égal à $K^{1-w_{min}}$.

• Dans le cas des codeurs convolutifs récursifs, comme $w_{min} = 2$, le gain d'entrelacement est égal à $1/K$.

• Dans le cas des codeurs convolutifs non récursifs, $w_{min} = 1$, il n'y a aucun gain d'entrelacement.

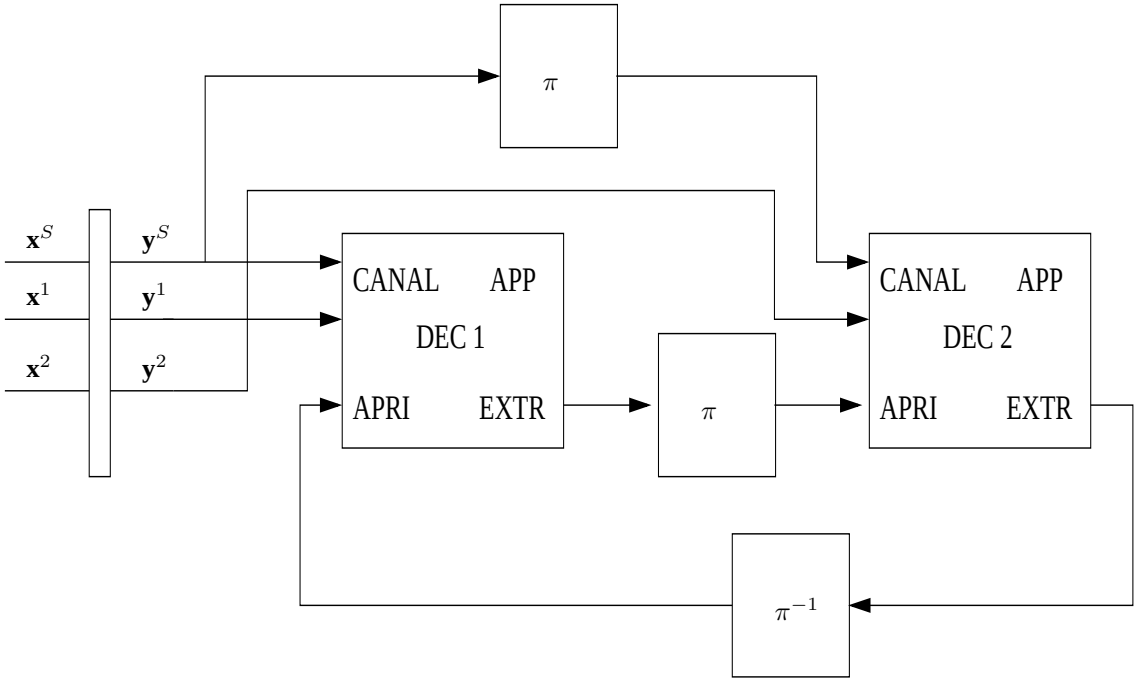
Comparaison des performances moyennes de codes PCC de rendement $R_t = 1/3$ composé de codes RSC(7,5) (en trait pointillé) et RSC(15,17) (en trait continu) et d'un entrelaceur uniforme $K = 100$ et $K = 1000$



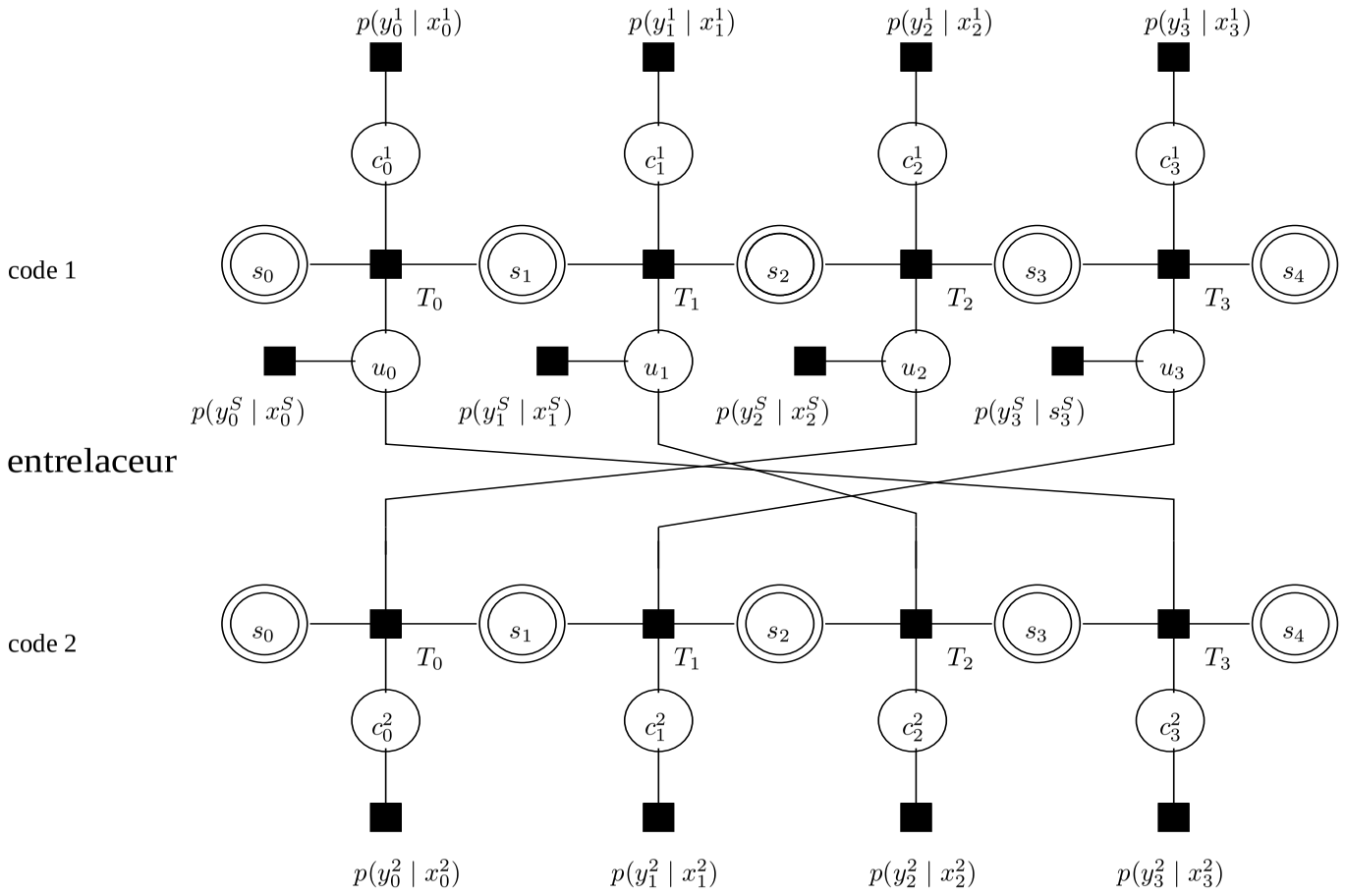
DECODEUR ITERATIF POUR LES TURBO CODES

• La complexité du calcul exact de $APP(u_i)$ dans une structure concaténée croit exponentionellement avec la longueur de la séquence d'information.

• L'objectif du décodage itératif est d'exploiter la structure du graphe du codeur PCC composé de 2 graphes élémentaires sans cycle pour factoriser la probabilité a posteriori $APP(u_i = a) = Pr\{u_i = a \mid \mathbf{y^S}, \mathbf{y^1}, \mathbf{y^2}\}$.



GRAPHE FACTORISE



DECODAGE ITERATIF DES TURBO CODES

• A chaque itération l, $1 \leq l \leq l_T$:

Le décodeur 1 calcule alors les probabilités extrinsèques $EXTR^{1(l)}(x_i^S)$ pour les K bits d'information à partir des observations relatives au premier codeur et des probabilités a priori $APRI^{1(l)}(x_i^S)$:

$$EXTR^{1(l)}(x_i^S = a) = \sum_{\mathbf{x^S}: x_i^S = a} \prod_{\substack{j=0 \\ j \neq i}}^{K-1} APRI^{1(l)}(x_j^S) p(y_j^S \mid x_j^S) \prod_{j'=0}^{N-K-1} p(y_{j'}^1 \mid x_{j'}^1) \qquad (35)$$

$EXTR^{1(l)}(x_i^S = a)$ est la probabilité extrinsèque et $APRI^{1(l)}(x_i^S = a)$ est la probabilité a priori sur le i-ième bit de la séquence d'entrée $\mathbf{u}$. $APRI^{1(l)}(x_i^S = a)$ est obtenue comme suit :

$$APRI^{1(l)}(x_i^S) = \begin{cases} 1/2 \quad \forall i & \text{pour} \quad l = 1 \\ EXTR^{2(l-1)}(x_i^S) & \text{pour} \quad l > 1 \end{cases}$$

De la même façon, le décodeur 2 calcule $EXTR^{2(l)}(x_i^S)$:

$$EXTR^{2(l)}(x_i^S = a) = \sum_{\mathbf{x^S}: x_i^S = a} \prod_{\substack{j=0 \\ j \neq i}}^{K-1} APRI^{2(l)}(x_j^S) p(y_j^S \mid x_j^S) \prod_{j'=0}^{N-K-1} p(y_{j'}^2 \mid x_{j'}^2) \qquad (36)$$

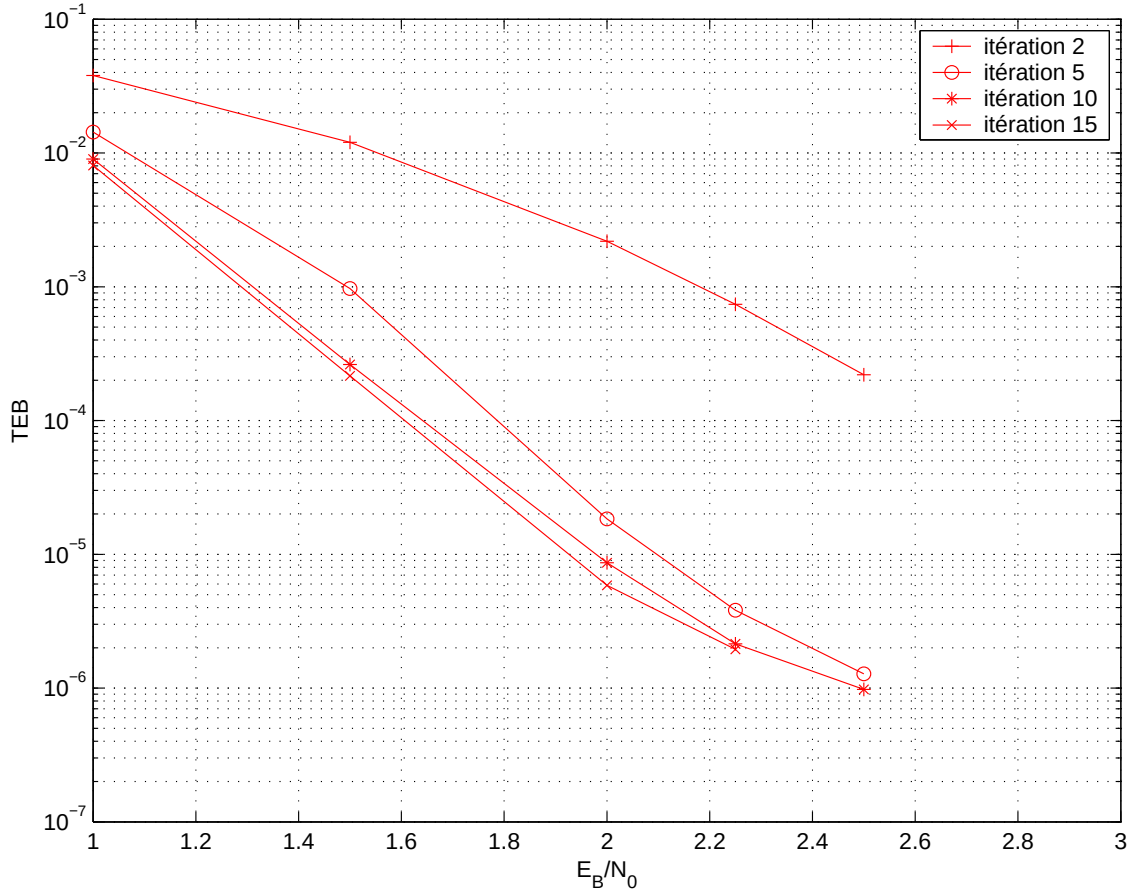
avec $APRI^{2(l)}(x_i^S) = EXTR^{1(l)}(x_i^S)$.

Finalement, pour réaliser la décision finale lors de la dernière itération l_T, on calcule $APP^{(l_T)}(x_i^S = a)$:

$$APP^{(l_T)}(x_i^S = a) \propto p(y_i^S \mid u_i) \times APRI^{2(l_T)}(x_i^S = a) \times EXTR^{2(l_T)}(x_i^S = a) \qquad (37)$$

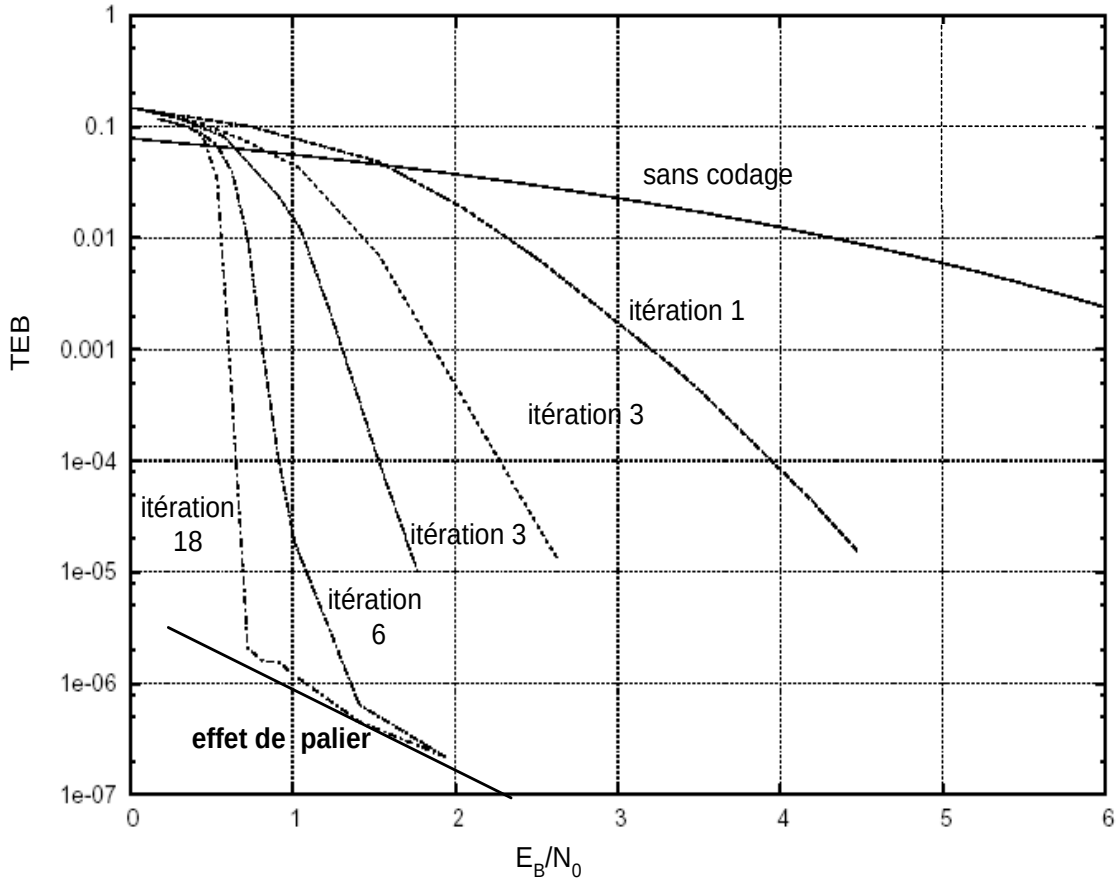
RESULTATS DE SIMULATION

• Turbo code de rendement $R_T = 1/2$ composé de 2 codes convolutifs RSC $(1, \frac{1+D^2}{1+D+D^2})$ et d'un entrelaceur optimisé S-random de dimension $K = 1024$.



RESULTATS DE SIMULATION

• Turbo code de rendement $R_T = 1/2$ composé de 2 codes convolutifs RSC $(1, \frac{1+D^4}{1+D+D^2+D^3+D^4})$ perforés et d'un entrelaceur pseudo aléatoire de dimension $K = 65536$.



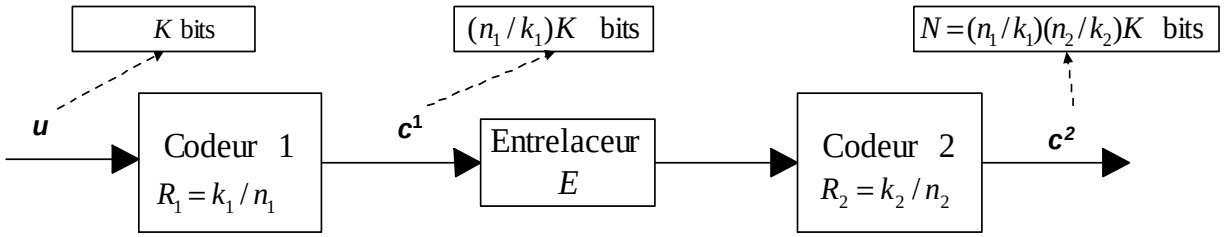
• Après 18 itérations, pour un TEB de 10^{-5}_{124} on a $E_B/N_0 = 0.7$ dB.

• Effet de palier qui apparait ici pour un TEB de l'ordre de 10^{-6} dû à la faible distance minimale des turbo codes.

• L'optimisation de l'entrelaceur permet de réduire cet effet.

CODEURS CONVOLUTIFS CONCATENES EN SERIE

• La structure du codeur est la suivante :



• Le code SCC est un code en bloc (N, K) avec $N = (\frac{n_2 n_1}{k_1 k_2})K$.

• Par exemple en choisissant un code C_1 de rendement 1/2 et un code C_2 de rendement $2/3$ on obtiendra un code SCC de rendement $R_T = 1/3$.

• Comme pour les turbo codes, le décodage est itératif.

• Pas d'effet de pallier mais seuil de déclenchement sensiblement moins bon que pour les turbo codes.

APPLICATIONS DES CODES CONCATENES

- Les codes LDPC eIRA ont été retenu dans la norme DVB S2.

- Les turbo codes ont été retenu dans les normes DVB-RCS et UMTS.

- Les codes produits ont été retenu dans la norme de réseau sans fil large bande IEEE 802.16. Les codes constituants sont des codes de parité (8,7),(16,15) et (32,31) ainsi que des codes de Hamming étendu (16,11), (32,26) et (64,57).