4) Résolution numérique d'un PL

a) Intérêt des sommets de l'EA

- Pour P_1, la solution est :

$$B = \left(\frac{9}{14}; \frac{3}{7}\right) \Rightarrow z_{max} = \frac{72}{7} \ (en\ M€)$$

- En gardant les contraintes de P_1, donc avec le même EA, on peut modifier l'objectif z

$$z_2 = 10x_1 + 5x_2$$
$$v_2 = \begin{bmatrix} 10 \\ 5 \end{bmatrix} // \begin{bmatrix} 1 \\ 0{,}5 \end{bmatrix}$$

  La nouvelle solution se situe en $c = (0{,}8\,; 0{,}24)$
  Le montage des voitures S rapporte moins, donc on privilégie les voitures L.

$$z_2^{max} = 9{,}2$$
$$z_3 = 10x_1 + 30x_2$$
$$v_3 = \begin{bmatrix} 10 \\ 30 \end{bmatrix} // \begin{bmatrix} 0{,}5 \\ 1{,}5 \end{bmatrix}$$

  La nouvelle solution se situe en $A = (0; 0{,}75)$
  On arrête la production de L

$$z_3^{max} = 22{,}5$$

- Plus surprenant :

$$z_4 = 10x_1 - 5x_2$$
$$v_4 = \begin{bmatrix} 10 \\ -5 \end{bmatrix} // \begin{bmatrix} 1 \\ -0{,}5 \end{bmatrix}$$

  La nouvelle solution se situe en $D = (0{,}8; 0)$
  On arrête la production de S

$$z_4^{max} = 8$$
$$z_5 = -10x_1 - 9x_2$$

  La solution est $O = (0; 0)$
  C'est le chômage technique

$$z_5^{max} = 0$$

- Enfin, dernier cas (peu probable) :

$$z_6 = 6x_1 + 5x_2$$
$$v_6 = \begin{bmatrix} 6 \\ 5 \end{bmatrix} // \begin{bmatrix} 0{,}6 \\ 0{,}5 \end{bmatrix} \perp (BC)$$

  Tous les points du segment (BC) sont solutions

$$z_6^{max} = 6$$

- De façon générale, si elle existe, la solution d'un PL se trouve sur un sommet de l'EA. Même s'il existe plusieurs solutions (ex : z_6), il y a quand même un sommet qui est solution.

b) Algorithmes de recherche de la solution

- Première idée : on teste tous les sommets. Deux difficultés :
    - Difficile de calculer les coordonnées des sommets
    - Si n et m sont élevés, le nombre de sommets devient très élevé

- On va utiliser le fait que l'EA est <u>convexe</u>.
  En partant d'un sommet quelconque de l'EA, et en se déplaçant le long des arêtes de l'EA, de telle sorte que z croît, on arrive forcément à la solution B.

- Cela ne serait pas vrai si l'EA n'était pas convexe.
  En partant de D, on resterait bloqué en C.

- On est donc sûr d'arriver à une solution en se déplaçant le long d'une séquence d'arêtes de l'EA, à partir d'un sommet quelconque. Au lieu de visiter tous les sommets, on en visite une partie seulement.

c) Variables d'écart et forme standard

- Pour résoudre P_1 numériquement, on introduit m=3 nouvelles variables, appelées <u>variables d'écart</u> (VE), notées e_1, e_2 et e_3 :
$$\begin{cases} 6x_1 + 5x_2 + e_1 = 6 \\ x_1 + 2x_2 + e_2 = 1{,}5 \\ x_1 + e_3 = 0{,}8 \end{cases}$$
  De par leur définition : $e_1 \geq 0, e_2 \geq 0, e_3 \geq 0$
  Cela ne serait pas vrai si les m=3 contraintes étaient de type ≥ et non ≤

- On appelle <u>forme standard</u> d'un PL la forme canonique de ce PL, où les contraintes ≤ ont été transformées en contraintes = grâce aux VE :
$$\begin{array}{ll} max\ z = 10x_1 + 9x_2 & \\ sc & 6x_1 + 5x_2 + e_1 = 6 \\ & x_1 + 2x_2 + e_2 = 1{,}5 \\ & x_1 + e_3 = 0{,}8 \\ & x_1, x_2, e_1, e_2, e_3 \geq 0 \end{array}$$

- L'intérêt de la forme standard est qu'elle permet de caractériser facilement les sommets de l'EA. En effet, pour chaque sommet, il y a exactement 2 variables (VD ou VE) qui sont nulles.
  Les 2 variables nulles sont appelées les <u>variables de base</u> (VB) : en A ce sont x_1 et e_2.
  Les autres variables sont appelées <u>variables hors base</u> (VH) : en A ce sont x_2, e_1 et e_3.
  En chaque sommet, on peut exprimer les VH et l'objectif z en fonction des VB.

d) Algorithme du simplexe (1947)

Algorithme itératif. On part d'un sommet de l'EA : ici on part de O = (0 ;0).

1- En O, les VB sont x_1 et x_2 :
$$\begin{cases} e_1 = 6 - 6x_1 - 5x_2 & (1) \\ e_2 = 1{,}5 - x_1 - 2x_2 & (2) \\ e_3 = 0{,}8 - x_1 & (3) \\ z = 10x_1 + 9x_2 & (4) \end{cases}$$
   z(0) = 0

On sort de la base la variable dont le coefficient dans z est le plus élevé > 0. Ici, c'est x_1. On se déplace en D, où la base est (x_2,e_3).

2- Pour exprimer les VH et z n fonction de (x_2,e_3), on commence par l'équation qui exprime la variable entrante (e_3) en fonction de la variable sortante (x_1) :

$$(3) \Longrightarrow x_1 = 0{,}8 - e_3$$

On remplace x_1 par cette expression dans les autres équations :

$$\begin{cases} e_1 = \dfrac{6}{5} + 6e_3 - 5x_2 & (5) \\ e_2 = \dfrac{7}{10} + e_3 - 2x_2 & (6) \\ x_1 = 0{,}8 - e_3 & (7) \\ z = 8 - 10e_3 + 9x_2 & (8) \end{cases}$$

$$z(D) = 8 > z(0)$$

C'est x_2 qui sort de la base : on se déplace de D en C, où les VB sont (e_1,e_3).

3- Ss

$$\begin{cases} x_2 = \dfrac{6}{25} + \dfrac{6}{5}e_3 - \dfrac{1}{5}e_1 & (9) \\ e_2 = \dfrac{11}{50} + \dfrac{2}{5}e_1 - \dfrac{7}{5}e_3 & (10) \\ x_1 = \dfrac{4}{5} - e_3 & (11) \\ z = \dfrac{254}{25} - \dfrac{9}{5}e_1 + \dfrac{4}{5}e_3 & (12) \end{cases}$$

$$z(C) = \frac{254}{25} > z(D)$$

C'est e_3 qui sort de la base : on se déplace de C en B, où la base est (e_1,e_2).

4- Ss

$$\begin{cases} e_3 = \dfrac{11}{70} + \dfrac{2}{7}e_1 - \dfrac{5}{7}e_1 & (13) \\ x_2 = \dfrac{3}{7} + \dfrac{1}{7}e_1 - \dfrac{6}{7}e_2 & (14) \\ x_1 = \dfrac{9}{14} - \dfrac{2}{7}e_1 + \dfrac{5}{7}e_2 & (15) \\ z = \dfrac{72}{7} - \dfrac{11}{7}e_1 - \dfrac{4}{7}e_2 & (16) \end{cases}$$

$$z(B) = \frac{72}{7}$$

C'est la solution car tous les coefficients des VB dans z sont ≤ 0.

$$\begin{cases} z_{max} = \dfrac{72}{7} \\ x_1^* = \dfrac{9}{14} \\ x_2^* = \dfrac{3}{7} \end{cases}$$

5) PL en nombre entiers

a) Exemple P3

- Forme canonique de P'_3

$$\begin{aligned}&max\ z = 8x_1 + 5x_2\\ &sc\ \ x_1 + x_2 \leq 6\\ &9x_1 + 5x_2 \leq 45\\ &x_1 \geq 0\\ &x_2 \geq 0\\ &x_1 \in \mathbb{N}\\ &x_2 \in \mathbb{N}\end{aligned}$$

- On appelle P_3 le PL de même forme canonique que P_3', mais à variables (x_1, x_2) continues. On dit que P_3 est le relaxé de P_3'.
  Résolution graphique de P_3 : $B = (3{,}75\,;2{,}25)\ donc$

$$\begin{cases}x_1^* = 3{,}75\\ x_2^* = 2{,}25\end{cases} \Longrightarrow z_{max} = 41{,}25\ (en\ €)$$

- La solution de P_3 n'est pas solution de P_3'. La résolution de P_3' est notamment plus difficile que celle de P_3. De façon générale, les problèmes de RO en nombres entiers sont beaucoup plus difficiles à résoudre que les problèmes à variables continues. Cela peut sembler paradoxal, car le nombre de points (x_1, x_2) admissibles est beaucoup plus petit pour P_3' que pour P_3 : 25 au lieu de +inf. En réalité, on sait que la solution de P_3 se situe sur un des 4 sommets.

b) Résolution par séparation/évaluation (branch and bound)

- On pourrait résoudre P_3' en calculant z pour chacun des 25 points (x_1, x_2) admissibles. On peut faire beaucoup mieux.
- La méthode de séparation/évaluation est une méthode générale de problèmes de RO. Son principe consiste à séparer l'ensemble admissible en plusieurs parties, puis à évaluer la valeur optimale de l'objectif sur chaque partie.
- Pour P_3', on commence par résoudre P_3 :

$$\left(x_{1,3}^*; x_{2,3}^*\right) = (3{,}75; 2{,}25) \Longrightarrow z_{max,3} = 41{,}25$$

  Cette solution n'est pas entière donc pas admissible. On sépare l'EA de P_3, noté E_3, en deux parties en choisissant une VD qui n'est pas entière dans la solution de P_3, par exemple x_1.
  On définit les deux sous-ensembles :

$$E_4 = E_3 \cap \{(x_1, x_2), x_1 \leq 3\}$$
$$E_5 = E_3 \cap \{(x_1, x_2), x_1 \geq 4\}$$

- On constate que $E_4 \cup E_5 \neq E_3$, mais que $E_4 \cup E_5$ contient les 25 points admissibles de P_3'. Or E_4 est l'EA d'un PL à variables continues :

$$\begin{aligned}&max\ z = 8x_1 + 5x_2\\ &sc\ \ x_1 + x_2 \leq 6\\ &9x_1 + 5x_2 \leq 45\\ &x_1 \leq 3\\ &x_1, x_2 \geq 0\\ &\Leftrightarrow P_4 = P_3 + (x_1 \leq 3)\end{aligned}$$

Résolution graphique de P_4 :

$$\left(x_{1,4}^*; x_{2,4}^*\right) = (3; 3) \Longrightarrow z_{max,4} = 39$$

Cette solution est entière donc admissible. Il est donc inutile de séparer de E_4. Or E_4 contient 22 points admissibles. On n'a donc pas besoin de calculer z pour les 21 points admissibles de E_4 autres que (3;3).

D'autre part, on est sûr que $z_{max} \geq 39$ pour P_3'.

- E_5 est l'EA du PL à variables continues P_5 : $P_5 = P_3 + (x_1 \geq 4)$
  Résolution graphique de P_5 : $\left(x_{1,5}^* \,; x_{2,5}^*\right) = (4\,; 1{,}8) \;\Longrightarrow z_{max,5} = 41$
  Comme cette solution est non entière <u>et</u> $z_{max,5} > 39$, on doit séparer E_5 :
  $$E_6 = E_5 \cap \{(x_1, x_2), x_2 \leq 1\}$$
  $$E_7 = E_5 \cap \{(x_1, x_2), x_2 \geq 2\}$$
  Résolution graphique de P_6 : $\left(x_{1,6}^* \,; x_{2,6}^*\right) = \left(\frac{40}{9}\,; 1\right) \;\Longrightarrow z_{max,6} = 40{,}5$
  On doit à nouveau séparer l'EA, car solution non entière <u>et</u> $z_{max} > 39$:
  $$P_8 = P_6 + (x_1 \leq 4)$$
  $$P_9 = P_6 + (x_1 \geq 5)$$
  Résolution graphique de P_8 : $\left(x_{1,8}^* \,; x_{2,8}^*\right) = (4\,; 1) \;\Longrightarrow z_{max,8} = 37$

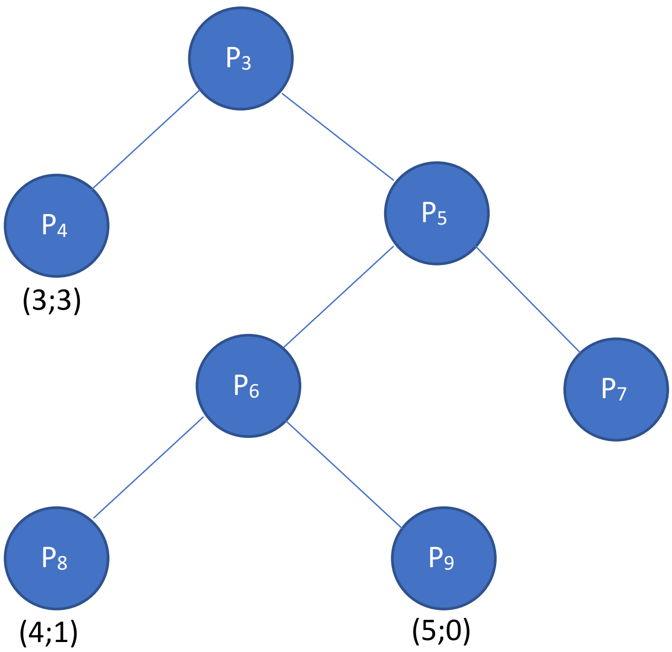
- Résolution graphique de P_9 : $\left(x_{1,9}^* \,; x_{2,9}^*\right) = (5\,; 0) \;\Longrightarrow z_{max,9} = 40$
  Solution admissible et on est sûr que $z_{max} \geq 40$.

- Résolution graphique de P_7 : E_7 = vide

- Bilan : $(x_1^* \,; x_2^*) = (5\,; 0) \;\Longrightarrow z_{max} = 40$

- Remarque : Si on avait arrondi (avec floor) la solution de P_3, on aurait trouvé :
  $$(x_1^* \,; x_2^*) = (3\,; 2) \;\Longrightarrow z_{max} = 34$$

c) Arbre de résolution



- La racine de cet arbre est P_3. Ses feuilles sont soit des PL à solution admissible (P_4, P_8 et P_9) soit des PL non réalisables (P_7).

- Cette méthode de résolution est convergente car l'arbre de résolution est de profondeur finie. Dans l'exemple de $P_3^{'}$, au lieu de calculer z pour les 25 points admissibles, on a résolu 7 PL.

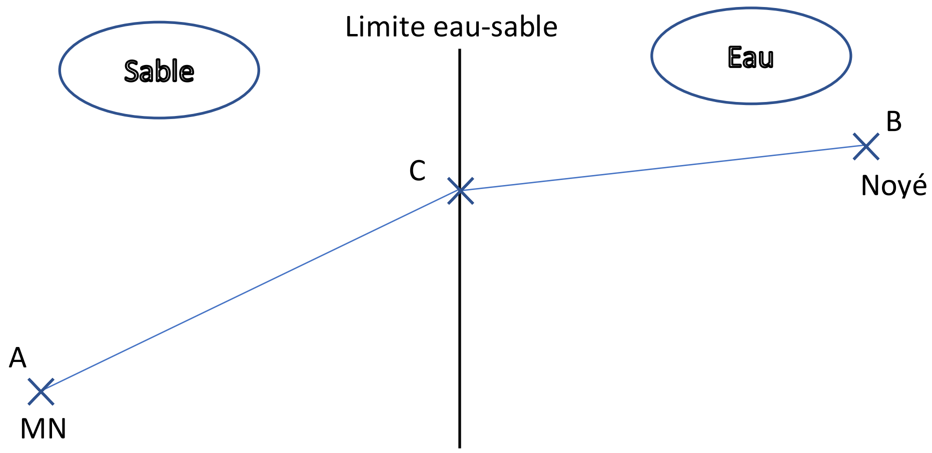
- Dans le TP2 :
  - Résolution de $P_3^{'}$ par récursivité d'une fonction de résolution d'un PL relaxé, qui utilise la fonction linprog de Matlab
  - Utilisation de la résolution par séparation/évaluation en Matlab.

- On a intérêt à parcourir l'arbre de résolution en profondeur d'abord : P_3, P_4

Programmation dynamique

1) Problème de maître-nageur

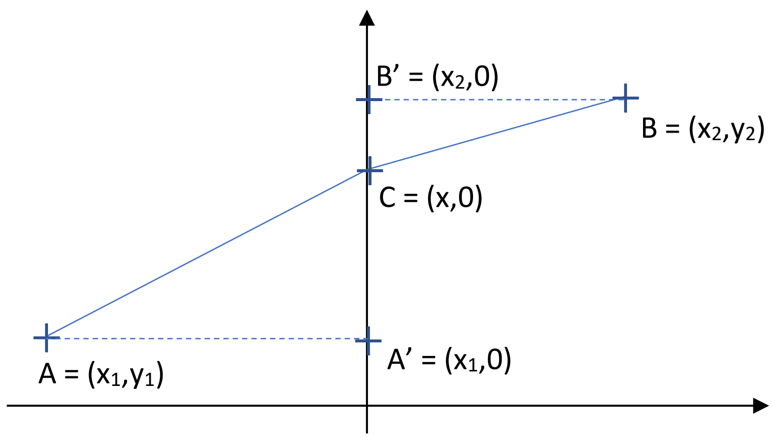
a) Énoncé du problème



La vitesse du MN vaut v_1 sur le sable et v_2 dans l'eau ($v_1 > v_2$).
Question : quel chemin le MN doit-il emprunter pour rejoindre le noyé au plus vite ?
Pour aller au plus vite entre 2 points dans l'eau ou entre 2 points dans le sable, il faut aller en ligne droite.
La question est donc : en quel point C le MN doit-il plonger ?

b) Formalisation mathématique

- On munit le sol d'un repère orthonormé :



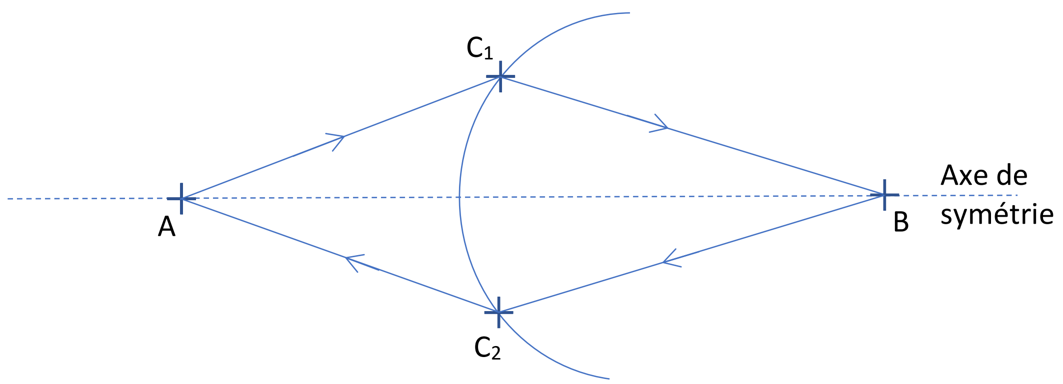
$$\overrightarrow{AC} = \begin{bmatrix} x - x_1 \\ -y_1 \end{bmatrix} \Rightarrow \|\overrightarrow{AC}\| = \sqrt{(x - x_1)^2 + y_1^2}$$
$$\overrightarrow{BC} = \begin{bmatrix} x - x_2 \\ -y_2 \end{bmatrix} \Rightarrow \|\overrightarrow{BC}\| = \sqrt{(x - x_2)^2 + y_2^2}$$

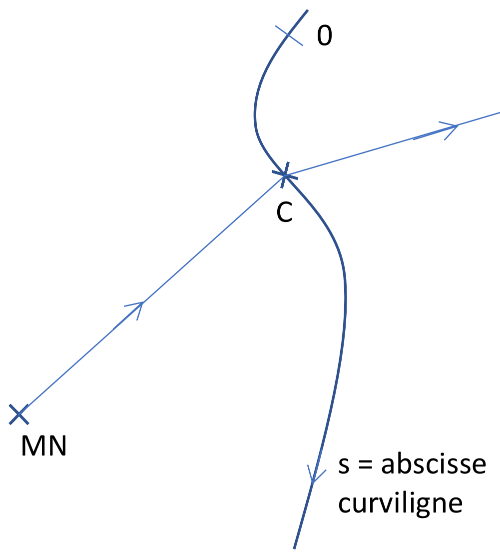
- Temps de parcours :
$$t(x) = \frac{\|\overrightarrow{AC}\|}{v_1} + \frac{\|\overrightarrow{BC}\|}{v_2} \Leftrightarrow t(x) = \frac{1}{v_1}\sqrt{(x - x_1)^2 + y_1^2} + \frac{1}{v_2}\sqrt{(x - x_2)^2 + y_2^2}$$

- Le problème s'écrit donc :$min\ t(x)\ x \in \mathbb{R}$
C'est un problème d'optimisation, comme les PL, mais :
    - Il n'y a pas de contrainte : plus simple que la PL
    - L'objectif t(x) n'est pas linéaire : plus compliqué que la PL

c) Résolution

- Condition nécessaire d'optimalité : $\frac{dt(x)}{dt} = 0\ \ (dérivée)$
Rappel :
$$\frac{d}{dx}[f(x)]^n = n[f(x)]^{n-1}\frac{df(x)}{dx}, \forall x \in \mathbb{R}$$
$$Donc\ pour\ n = \frac{1}{2} : \frac{d}{dx}\sqrt{f(x)} = \frac{1}{2}\frac{1}{\sqrt{f(x)}}\frac{df(x)}{dx}$$





Alignement de séquences

a) Séquences d'ADN

ADN = séquence de lettres parmi A, C, T, G.
La comparaison binaire n'est pas satisfaisante car les données réelles sont imprécises (ex : données manquantes). On va « forcer » l'alignement.

Exemple :
A A G T A G G C

A T G G T A C G T C

Les croisements sont interdits. On obtient :

- 6 correspondances correctes
- 2 substitutions
- 2 insertions

Le résultat semble plutôt bon.

Il existe un algorithme : DTW (Dynamic Time Warping). Il effectue l'alignement de façon optimale et donne un score au résultat. Il est très rapide car il utilise le principe de la programmation dynamique.

Si la séquence 1 comporte I lettres et la séquence 2 comporte J lettres, on forme une matrice de taille $(I+1)\times(J+1)$. Chaque ligne $i \in [2, I+1]$ correspond à une lettre de la séquence 1 et chaque colonne $j \in [2, J+1]$ correspond à une lettre de la séquence 2.

		A	T	G	G	T	A	C	G	T	C
	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
A	∞	0	1	2	3	4	4	5	6	7	8
A	∞	0	1	2	3	4	4	5	6	7	8
G	∞	1	1	1	1	2	3	4	4	5	6
T	∞	2	1	2	2	1	2	3	4	4	5
A	∞	2	2	3	3	2	1	2	3	4	5
G	∞	3	3	2	2	3	2	2	2	3	4
G	∞	4	4	2	2	3	3	3	2	3	4
C	∞	5	5	3	3	3	4	3	3	3	3

substitution

- Initialisations :
$$\begin{cases} g(2:end,1) = g(1,2:end) = \infty \\ g(1,1) = 0 \end{cases}$$
Pour les autres cases : *cases au-dessus* *cases de gauche* *case en haut à gauche*
$$g(i,j) = \min(g(i-1,j) + p_1 \times d, g(i,j-1) + p_2 \times d, g(i-1,j-1) + p_3 \times d)$$
$où\ d = \begin{cases} 0\ si\ lettres\ identiques \\ 1\ sinon \end{cases}$ dans le cas des séquences d'ADN.
et les poids p_1, p_2, p_3 sont des paramètres de l'algorithme DTW.
Pour les séquences d'ADN : $p_1 = p_2 = p_3 = 1$.

Par conséquent, pour ce problème particulier :
$g(i,j) = d + \min\big(g(i-1,j), g(i,j-1), g(i-1,j-1)\big)$

- Le score de cet alignement vaut : $S = \frac{g(I+1,J+1)}{I+J}$, ici $S = \frac{3}{8+10} \approx 0{,}16$
  Un « bon » alignement correspond à un score faible.

- Le parcours optimal s'obtient en recherchant, pour chaque case, la case qui a la valeur la plus faible. En cas de conflit, on privilégie la diagonale. Les flèches horizontales (ou verticales) correspondent à des <u>insertions</u>. Pour les flèches diagonales, deux cas se présentent :
  - Si les valeurs des deux cases sont égales, c'est une <u>bonne correspondance</u>,
  - Sinon c'est une <u>substitution</u>.

- S'il y a conflit entre les cases de gauche et du dessus, on choisit au hasard.

- Un avantage annexe de cet algorithme est qu'il donne l'alignement optimal entre deux séquences tronquées, par exemple AAGTA et la séquence 2 (en vert). Le score devient : $S = \frac{5}{5+10} = \frac{1}{3}$

b) Reconnaissance vocale

On sait maintenant aligner n'importe quelle paire de séquences. Si on dispose d'un dictionnaire de séquences, on peut comparer une « séquence de test » à toutes les séquences du dictionnaire en calculant les scores.

Le score optimal (ici : minimal) correspond à l'entrée du dictionnaire la plus proche de la séquence de test. C'est le principe de la reconnaissance (ex : Shazam).

Au lieu d'aligner des séquences d'ADN on aligne des enregistrements sonores (ES). Chaque ES est transformé en une séquence de vecteurs de $\mathbb{R}^{12}$ qui caractérise la voix. Un ES est donc une matrice de taille $12 \times n$, où n est proportionnel à la durée de l'ES. On peut calculer autant de scores qu'il y a d'entrées dans le dictionnaire. Si le score minimal est au-dessus d'un seuil, l'ES de test n'est pas reconnu. Sinon, cela permet de lancer une action : c'est le principe de la commande vocale (ex : serveur vocal interactif). Cf TP4