

# Projet 3 - RODD

BATY LÉO, BRUNOD-INDRIGO LUCA

3 février 2021

## 1 Premier modèle

Notons  $\mathcal{F}$  (resp.  $\mathcal{M}$ ) l'ensemble des individus femelles (resp. mâles) et  $\mathcal{K}_{ij}^1$  (resp.  $\mathcal{K}_{ij}^2$ ) l'ensemble des individus ayant un (resp. deux) exemplaire(s) de l'allèle  $j$  du gène  $i$ . On considère également les variables entières  $(x_k)_{k=1\dots p}$  donnant le nombre de progénitures de l'individu  $k$ .

L'espérance du nombre d'allèles disparus s'écrit :

$$\begin{aligned}\mathbb{E}(N^{disparus}) &= \sum_{i \leq M} \sum_{j \leq t(i)} \mathbb{E}(\mathbf{1}_{ij \text{ disparaît}}) \\ &= \sum_{i \leq M} \sum_{j \leq t(i)} \mathbb{P}(ij \text{ disparaît})\end{aligned}$$

Or, on remarque que :

$$\mathbb{P}(ij \text{ disparaît}) = \begin{cases} 0 & \text{si } \exists k \in \mathcal{K}_{ij}^2 \text{ tel que } x_k > 0 \\ \left(\frac{1}{2}\right)^{\sum_{k \in \mathcal{K}_{ij}^1} x_k} & \text{sinon.} \end{cases}$$

Ce qui, les  $x_k$  étant entiers, se réécrit :

$$\mathbb{P}(ij \text{ disparaît}) = \max\left(0, \left(\frac{1}{2}\right)^{\sum_{k \in \mathcal{K}_{ij}^1} x_k} - \sum_{k \in \mathcal{K}_{ij}^2} x_k\right)$$

On en déduit le modèle suivant :

$$\begin{aligned}
& \min_{x,p,t} \sum_{i \leq M} \sum_{j \leq t(i)} p_{ij} \\
& s.t. \log(t_{ij}) \geq \sum_{k \in K_{ij}^1} x_k \log\left(\frac{1}{2}\right) \\
& p_{ij} \geq t_{ij} - \sum_{k \in K_{ij}^2} x_k \\
& \sum_{k \in \mathcal{M}} x_k = P \\
& \sum_{k \in \mathcal{F}} x_k = P \\
& x_k \in \mathbb{N} \\
& p_{ij} \geq 0
\end{aligned}$$

## 2 Approximation linéaire

En approchant la fonction logarithme par une fonction affine par morceaux comme indiqué dans le sujet, on obtient le modèle suivant :

$$\begin{aligned}
& \min_{x,p} \sum_{i \leq M} \sum_{j \leq t(i)} p_{ij} \\
& s.t. \log(\theta_r) + \frac{1}{\theta_r}(t_{ij} - \theta_r) \geq \sum_{k \in K_{ij}^1} x_k \log\left(\frac{1}{2}\right) \\
& p_{ij} \geq t_{ij} - \sum_{k \in K_{ij}^2} x_k \\
& \sum_{k \in \mathcal{M}} x_k = P \\
& \sum_{k \in \mathcal{F}} x_k = P \\
& x_k \in \mathbb{N} \\
& p_{ij} \geq 0
\end{aligned}$$

Ce modèle fournit bien une solution admissible au problème, étant donné que le vecteur  $x_k$  vérifie toujours les contraintes d'intégrité et de conservation de la taille de la population. De plus, puisqu'il s'agit d'une approximation par excès du logarithme, les contraintes  $\log(\theta_r) + \frac{1}{\theta_r}(t_{ij} - \theta_r) \geq \sum_{k \in K_{ij}^1} x_k \log\left(\frac{1}{2}\right)$  à  $x_k$  fixés conduisent à des valeurs de  $t_{ij}$  plus faibles que la contrainte  $\log(t_{ij}) \geq \sum_{k \in K_{ij}^1} x_k \log\left(\frac{1}{2}\right)$ , donc également des valeurs plus faibles des  $p_{ij}$  et de l'objectif. Une solution optimale de ce modèle est donc bien une borne inférieure.

### 3 Calculs sur les instances de l'énoncé

Le tableau 1 montre les résultats obtenus sur les instances de l'énoncé en prenant  $\theta_1 = 0.0001$  et  $h = 50$ .

Instance	Temps	Nombre de noeuds	Espérance	Probabilités de disparition										Borne inférieure
				A	a	B	b	C	c	D	d	E	e	
$x_k \leq 2$	0.00	0	0.0625	0	0	0	0.0625	0	0	0	0	0	0	0.06243336502896749
$x_k \leq 3$	0.15	0	0.015625	0	0	0	0.015625	0	0	0	0	0	0	0.015620569018547845

TABLE 1 – Résultats sur les instances de l'énoncé

### 4 Essais sur des instances aléatoires

Le tableau 2 donne les résultats obtenus en appliquant le programme à des instances de tailles différentes en conservant les paramètres  $\theta_1 = 0.0001$  et  $h = 50$ .

Taille de population	Nombre de gènes	Temps	Nombre de noeuds	Espérance	Borne inférieure
10	6	0.03	0	3.814697265625e-6	0.0
12	8	0.02	0	0	0.0
16	10	0.03	0	6.103562191128731e-5	0.0
20	16	0.01	0	6.198883056640625e-6	0.0
30	19	0.02	0	5.122274165589302e-9	0.0
40	25	0.04	0	5.911715561532305e-12	0.0
50	31	0.06	0	7.549516567451072e-15	0.0
60	38	0.09	0	4.662069341687669e-18	0.0
80	50	0.19	0	9.155273437500011e-5	0.0
100	62	0.27	0	6.103515625e-5	0.0
160	100	0.80	0	0.00010681338608264923	0.0
200	125	0.97	0	0.00018310640007257462	0.0
300	188	2.84	0	7.629767065964188e-5	0.0
400	250	17.49	0	6.104307252030594e-5	0.0
500	312	14.50	0	0.00011062668636441231	0.0

TABLE 2 – Résultats sur des instances aléatoires de tailles différentes

On peut constater que, malgré l'augmentation de taille, CPLEX réussit toujours à résoudre le problème à la racine. Par ailleurs, et comme cela pouvait être attendu, l'écart entre la borne inférieure et l'espérance calculée à partir de la solution obtenue semble augmenter. Nous pensons que cela est dû au fait que le manque de précision dans l'approximation du logarithme a davantage d'impact quand la taille de l'instance augmente.

Le tableau 3 montre les résultats obtenus en appliquant le programme à une même instance aléatoire pour différentes valeurs de  $h$ . On remarque naturellement que l'augmentation de  $h$  augmente le temps de calcul puisqu'il y a davantage de contraintes. Cependant, alors qu'on aurait pu s'attendre à ce que l'espérance obtenue augmente du fait de la meilleure qualité de l'approximation du logarithme, on observe seulement deux valeurs différentes qui ne permettent pas de définir une tendance. Cela est pourrait être dû à une spécificité de l'instance aléatoire utilisée.

$h$	Temps	Nombre de noeuds	Espérance	Borne inférieure
50	1.1836340427398682	0	0.0001068115234375	0.0
60	1.3889548778533936	0	0.0001068115234375	0.0
70	1.713580846786499	0	0.0001068115234375	0.0
80	1.942518949508667	0	0.0001068115234375	0.0
90	2.1335439682006836	0	0.0001068115234375	0.0
100	2.448179006576538	0	0.0001068115234375	0.0
110	2.729564905166626	0	0.0001068115234375	0.0
120	3.1832730770111084	0	0.0001068115234375	0.0
130	3.364945888519287	0	0.0001068115234375	0.0
140	3.5438148975372314	0	0.0001068115234375	0.0
150	3.849707841873169	0	0.0001068115234375	0.0
160	4.177340030670166	0	0.0001068115234375	0.0
170	4.513378143310547	0	0.0001068115234375	0.0
180	5.585042953491211	0	6.866548210382462e-5	0.0
190	6.031890153884888	0	6.866548210382462e-5	0.0
200	5.396224021911621	0	0.0001068115234375	0.0

TABLE 3 – Résultats sur une instance aléatoire pour différentes valeurs de  $h$